Lasse Rempe und Rebecca Waldecker

Zahlentheorie ◇ Algorithmik ◇ Kryptographie

Primzahltests

für Einsteiger

Ein Buch für Schule und Studium

Inhalt

Vorwort

„Forschung in der Mathematik – wie geht das denn?“ Mit dieser Frage werden wir in Gesprächen häufig konfrontiert. Dass in Physik, Chemie, Biologie und weiteren Wissenschaften noch viele Fragen ungelöst sind, ist wohl weithin bekannt. Aber dass dies auch in der Mathematik der Fall ist, scheint selbst mathematikinteressierten Menschen kaum bewusst zu sein.

Das hängt einerseits mit dem Bild der Mathematik in der Gesellschaft zusammen, andererseits aber auch mit der Natur der modernen Mathematik selbst. Etwa besteht eine Schwierigkeit darin, dass aktuelle mathematische Fragestellungen und Ergebnisse für Nichtexperten meist nicht zugänglich sind. Selbst für uns als Mathematiker sind Forschungsergebnisse, welche nicht in unsere Spezialgebiete fallen, kaum zu erschließen. Natürlich gibt es Ausnahmen zu dieser Regel, wie zum Beispiel den in den 1990er Jahren von Andrew Wiles bewiesenen **großen Satz von Fermat**, dessen Formulierung an Einfachheit kaum zu übertreffen ist[1]. Doch gerade in diesem Fall ist der Beweis außerordentlich lang und schwierig, und nur wenige Experten weltweit sind wirklich in der Lage, ihn vollständig zu verstehen.

Im Sommer des Jahres 2002 gelang dem indischen Informatik-Professor Manindra Agrawal, gemeinsam mit seinen Studenten Neeraj Kayal und Nitin Saxena, ein Durchbruch im Gebiet der **algorithmischen Zahlentheorie**: Die drei Wissenschaftler beschrieben ein **effizientes** und **deterministisches** Verfahren, um festzustellen, ob eine gegebene natürliche Zahl eine Primzahl ist. (Die Bedeutung dieser Begriffe werden wir im Laufe dieses Buches erklären.) Besonders bemerkenswert an dieser Arbeit ist, dass sie trotz ihrer Bedeutung nur elementare mathematische Grundkenntnisse erfordert, welche Studenten der Mathematik oder Informatik üblicherweise im Grundstudium erwerben. Zusätzlich betrifft dieses Resultat einen Bereich der Mathematik, dessen Relevanz heute aufgrund der Anwendung von Verschlüsselungsverfahren im Internet (von „eBay“ bis zum Online-Banking) unbestritten ist.

Diese Konstellation empfanden wir als einzigartigen Glücksfall, weshalb wir

[1] „Ist $n > 2$, so gibt es keine ganzen Zahlen $a, b, c \neq 0$ mit $a^n + b^n = c^n$.“

im Sommer 2005 im Rahmen der „Deutschen SchülerAkademie“ einen Kurs zu diesem Thema angeboten haben. Dort haben wir über zweieinhalb Wochen hinweg 16 hochmotivierte Oberstufenschüler auf dem Weg von den Grundlagen bis zum Verständnis dieses aktuellen mathematischen Ergebnisses begleitet. Der Spaß, den die Teilnehmenden dabei hatten, und der Enthusiasmus, den sie an den Tag legten, motivierten uns, das vorliegende Buch zum selben Thema zu schreiben. Ein großer Teil des Manuskripts enstand daher zeitnah zur Schülerakademie, von November 2005 bis April 2007, und orientiert sich am Aufbau des Kurses.
Wir danken Frau Schmickler-Hirzebruch vom Verlag Vieweg+Teubner für die reibungslose Zusammenarbeit und Helena Mihaljević-Brandt, Katharina Radermacher, Stefanie Söllner und Yasin Zähringer für ihre Hilfe beim Korrekturlesen. Ganz besonderer Dank gilt „unseren“ DSA-Kursteilnehmenden: Andreas, Christin, Coline, Ina, Fabian, Feliks, Haakon, Johannes, Katharina, Kerstin, Hinnerk, Martin und Martin, Tabea, Yasin und Yvonne – ihr wart ein wunderbarer Kurs! Ohne euch wäre dieses Büchlein nie enstanden!

Lasse Rempe und Rebecca Waldecker
Sommer 2009

Einleitung

Die meisten von uns lernen in der Schule, was eine Primzahl ist: eine Zahl, welche genau zwei Teiler besitzt, nämlich 1 und sich selbst. Des Weiteren hören wir, dass jede natürliche Zahl in ihre Primzahlfaktoren zerlegt werden kann – zum Beispiel ist $2009 = 7 \cdot 7 \cdot 41$, und sowohl 7 als auch 41 sind Primzahlen. Was aber in der Schule eher nicht deutlich wird, ist, dass dies nur der Anfang einer mehrere tausend Jahre alten Geschichte ist, in der Mathematikerinnen[2] im Gebiet der „Zahlentheorie“ den Geheimnissen der Primzahlen auf den Leib zu rücken versuchten. Und nach wie vor gibt es in diesem Zusammenhang Probleme, die weit von einer Lösung entfernt sind! (Einige offene Fragen finden sich im Anhang.)

In der Regel ist uns auch nicht bewusst, dass wir im täglichen Leben heute nahezu ständig mit Primzahlen umgehen. Noch im Jahr 1940 schrieb der englische Mathematiker Hardy in seinem Buch „A mathematician's apology“ [Har], dass die Zahlentheorie keine vorstellbaren praktischen Anwendungen hätte, sondern es verdiene, allein wegen ihrer Schönheit studiert zu werden. In der zweiten Hälfte des zwanzigsten Jahrhunderts aber gewann die Frage nach sicheren elektronischen Kommunikationsmethoden aufgrund von Fortschritten in der Computertechnik stark an Bedeutung. Während dieser Entwicklung zeigte sich, dass Hardy der Zahlentheorie nicht in ihrer Schönheit, wohl aber in ihrer Anwendbarkeit Unrecht getan hatte. Die Informatiker Rivest, Shamir und Adleman entwickelten 1977 das heute nach ihnen benannte **RSA-Verfahren** zur sicheren Übertragung von Nachrichten, zu welchen außer Senderin und Empfängerin niemand Zugang haben sollte. Es ist heute Grundlage aller gängigen Verschlüsselungsmethoden, z.B. bei Kreditkartentransaktionen im Internet oder beim Online-Banking.

Das RSA-Verfahren werden wir in Abschnitt 4.2 genau untersuchen – die Grundidee liegt im folgenden, zunächst sehr überraschenden Prinzip:

> Es ist (vergleichsweise) **einfach**, einer Zahl anzusehen, ob sie eine Primzahl ist. Ist sie aber keine, so ist es **schwierig**, ihre Primfaktoren tatsächlich zu bestimmen.

[2] Wir verwenden standardmäßig die weibliche Form, schreiben also z.B. „Leserinnen“, wenn wir „Leserinnen und Leser“ meinen.

```
RSA-2048 = 25195908475657893494027183240048398571429282126204
           03202777713783604366202070759555626401852588078440
           69182906412495150821892985591491761845028084891200
           72844992687392807287776735971418347270261896375014
           97182469116507761337985909570009733045974880842840
           17974291006424586918171951187461215151726546322822
           16869987549182422433637259085141865462043576798423
           38718477444792073993423658482382428119816381501067
           48104516603773060562016196762561338441436038339044
           14952634432190114657544454178424020924616515723350
           77870774981712577246796292638635637328991215483143
           81678998850404453640235273819513786365643912120103
           97122822120720357
```

Abbildung 1. Für das Auffinden der Primfaktoren der Zahl „RSA-2048“ war bis 2007 ein Preis von 200 000 US-Dollar ausgeschrieben. Bis heute ist keine Faktorisierung bekannt.

Angesichts des in der Schule vermittelten Wissens über Primzahlen ist das eine erstaunliche Behauptung. Zum Beispiel können wir doch mit Hilfe des jahrtausendealten „Siebs des Eratosthenes“ (siehe Abschnitt 1.5) feststellen, ob eine gegebene Zahl prim ist; falls das nicht der Fall ist, erhalten wir damit auch gleich eine Liste ihrer Primfaktoren.

Wer allerdings das Sieb des Eratosthenes einmal zum Auffinden etwa aller Primzahlen unter 400 verwendet hat, dem leuchtet ein, dass dieses Verfahren für Zahlen mit mehreren hundert oder gar tausend Stellen – und solche werden in der Kryptographie tatsächlich verwendet – selbst für moderne Computer nicht praktikabel ist. Um die Forschung in diesem Gebiet voranzutreiben, gab es zwischen 1991 und 2007 eine wahre Herausforderung für Experten und Knobler auf der ganzen Welt: die Factoring Challenge der RSA Laboratories. Veröffentlicht wurde eine Liste sogenannter RSA-Zahlen (Produkte zweier verschiedener, extrem großer Primzahlen), mit dem Aufruf, sie zu faktorisieren. Für manche Zahlen wurden bei erfolgreicher Faktorisierung sogar hohe Preisgelder gezahlt. Dass die offizielle Herausforderung beendet ist, heißt übrigens nicht, dass alle Faktorisierungen gefunden wurden!

Warum sollte aber – wie oben behauptet – das Erkennen von Primzahlen einfacher sein als das Auffinden von Primfaktoren? Der Schlüssel ist, Eigenschaften von Primzahlen zu finden, die eben nicht auf das Auffinden von Faktoren oder Ähnliches hinauslaufen, sondern die mit weniger Aufwand überprüfbar sind.

Bereits 1640 wurde von Fermat eine Eigenschaft von Primzahlen formuliert, die leicht testbar ist. In den 1970er Jahren wurde dann eine Verfeinerung dieser Eigenschaft von den Informatikern Miller und Rabin in einen praxistauglichen Primzahltest umgewandelt, welcher heute die Basis der Verschlüsselung mit Hilfe des RSA-Verfahrens darstellt. Nebenbei bemerkt ist das ein Beispiel dafür, dass auch aus mathematischen Theorien, welche nicht im Hinblick auf Anwendbarkeit entwickelt wurden, im Nachhinein praktischer Nutzen gezogen werden kann.

Auch wenn die Relevanz der im zwanzigsten Jahrhundert entdeckten Verfahren zur Primzahlerkennung für die Praxis außer Frage steht, haben sie eine aus theoretischer Sicht etwas unbefriedigende Eigenschaft – sie sind **randomisiert**, d.h. ihre Ausführung basiert auf der zufälligen Auswahl gewisser Parameter. Daher besteht auch eine (geringe) Chance, dass die Ausführung nicht in befriedigender Zeit ein korrektes Ergebnis liefert. Da erreicht werden kann, dass diese „Fehlerwahrscheinlichkeit“ verschwindend gering ist, hat dies für die Praxis keine ernsthaften Auswirkungen. Es wirft aber die Frage auf, ob Randomisierung wirklich notwendig ist oder ob es auch ein effizientes Verfahren der Primzahlerkennung geben könnte, welches **deterministisch** ist, also ohne die Verwendung von Zufallszahlen auskommt.

Dieses Problem blieb über Jahrzehnte ungelöst, bis im Jahr 2002 die indischen Informatiker Agrawal, Kayal und Saxena eine elegante Lösung vorlegten. Wegen der grundlegenden Bedeutung des Resultats und der elementaren Natur der Lösung stieß dieses Ergebnis quer durch die Mathematik auf große Beachtung. In den *Mitteilungen der Deutschen Mathematiker-Vereinigung* wurde die Arbeit dementsprechend noch im selben Jahr in einem Artikel von Folkmar Bornemann als „Durchbruch für Jedermann“ gefeiert [Bo]. Im Jahr 2004 wurde sie in den „Annals of Mathematics“ veröffentlicht [AKS], der renommiertesten Fachzeitschrift für Mathematik.

Das Ziel des vorliegenden Buches ist es, den Beweis des Resultats von Agrawal, Kayal und Saxena vollständig darzustellen, ohne von der Leserin Vorwissen zu erwarten, welches über allgemeine Rechenkenntnisse und die Fähigkeit und Bereitschaft zum logischen Denken hinausgeht. Dabei werden wir naturgemäß die mathematischen und informatischen Hintergründe entwickeln, die für das Verständnis des Beweises und seiner mathematischen Bedeutung vonnöten sind. Wir hoffen, dass die Lektüre zugleich einen Eindruck von der Schönheit der behandelten Methoden vermitteln kann und davon, wie viele Fragen trotz aller Fortschritte noch offen sind.

Über dieses Buch

Dieses Buch richtet sich an interessierte Schülerinnen und Lehrerinnen, aber auch an Mathematik- und Informatik-Studierende (für die es schon im Grundstudium

zugänglich ist). Es eignet sich etwa zur Begleitung eines intensiven Sommerkurses für Schülerinnen oder eines (Pro-)Seminars während des Studiums.

Es ist dabei nicht unsere Absicht, vor allem eine Einführung in die Zahlentheorie oder Algorithmik zu geben. Derartiger Bücher gibt es viele – in den Literaturangaben zu den einzelnen Kapiteln wird die Leserin einige dieser Texte wiederfinden – und wir könnten ihnen nicht viel hinzufügen. Andererseits ist unser Buch auch keine mathematische Forschungsarbeit; es ist weder von noch für Experten geschrieben. Mathematikerinnen oder Informatikerinnen mit einem soliden Grundwissen, die sich für die Arbeit von Agrawal, Kayal und Saxena interessieren, werden im Original-Artikel oder in anderen Quellen (wie dem für eine fortgeschrittenere Zielgruppe geschriebenen Buch „Primality Testing in Polynomial Time“ von Dietzfelbinger [Dtz]) ein angemesseneres Schritttempo vorfinden.

Unsere Absicht ist, über das gesamte Buch das eigentliche Ziel – die Behandlung des Algorithmus von Agrawal, Kayal und Saxena („AKS-Algorithmus“) – im Auge zu behalten und genau jene Konzepte zu behandeln, welche als Hintergrund erforderlich sind. Gleichzeitig führen wir die Leserin behutsam in die Welt der mathematischen Beweisführung ein. Unseres Wissens nach unterscheidet sich unser Text in dieser vollständigen Behandlung eines aktuellen mathematischen Ergebnisses grundlegend von anderen Büchern mit derselben Zielgruppe.

Der erste Teil des Buches dient hauptsächlich der Einführung in die Zahlen- und Algorithmentheorie, soweit das für den AKS-Algorithmus erforderlich ist. Wir geben außerdem einen kurzen historischen und mathematischen Einblick in das Gebiet der Kryptographie. Insgesamt haben wir uns in Inhalt und Reihenfolge stark an den entsprechenden Vorbereitungen in unserem Kurs bei der Deutschen SchülerAkademie orientiert, auch was die Ausführlichkeit betrifft.

Im zweiten Teil stellen wir dann im Wesentlichen den Inhalt der AKS-Arbeit dar – dabei können wir mathematisch auf den ersten Teil zurückgreifen und uns weitere „Zutaten“ zu gegebener Zeit erarbeiten. Hier ist es uns wichtig, die zugrundeliegenden Ideen zu erläutern und gleichzeitig den Beweis korrekt und ausführlich darzustellen. Leserinnen mit soliden Grundkenntnissen können den ersten Teil überspringen, sich gleich am AKS-Algorithmus versuchen und gegebenenfalls zurückblättern.

Zahlreiche Aufgaben und Bemerkungen sollen die Lektüre vertiefen. Dabei sind die Aufgaben nicht nur dazu gedacht, zu überprüfen, ob man die neuen Ideen verstanden hat, sondern sollen eine generelle Einladung zum „Lernen durch Selbermachen“ sein. Unserer Meinung nach begreift man Mathematik so am besten. Und man weiß eine ansonsten ganz natürlich erscheinende Idee viel mehr zu schätzen, wenn man sie nach eventuell tagelangem Überlegen selbst gefunden hat! Wir haben die Aufgaben absichtlich nicht nach Schwierigkeit geordnet und diejenigen, die den Einsatz eines Computers erfordern, mit (P) gekennzeichnet. Falls

eine Aufgabe später im Text verwendet wird, versehen wir sie mit einem (!). Am Ende eines Abschnittes gibt es meist weiterführende (evtl. schwierigere) Übungen und Anmerkungen. Wir möchten damit interessierte Leserinnen einladen, sich das jeweilige Thema weiter zu erschließen – sie können aber beim ersten Lesen getrost übersprungen werden. Im Anhang befinden sich ein Abschnitt über offene Probleme im Zusammenhang mit Primzahlen sowie Lösungen und Hinweise zu den mit (!) gekennzeichneten Übungsaufgaben. Vollständige Aufgabenlösungen und unsere Kontaktdaten sind auf der Internetseite

http://www.viewegteubner.de/index.php;do=show/site=v/book_id=17128

über „Online Plus“ zu finden. Wir freuen uns über Hinweise auf Fehler (selbst, wenn es nur Tippfehler sind) und über Fragen und Verbesserungsvorschläge.

Beweise

Der „Beweis“ ist ein zentrales Konzept der Mathematik. Er dient dazu, die Wahrheit einer mathematischen Behauptung nachzuweisen und sie damit über jeden Zweifel zu erheben. In einem Beweis machen wir eine Reihe logischer Schlüsse, eventuell unter Verwendung schon bekannter Ergebnisse, um aus den gegebenen Voraussetzungen die gewünschte Aussage zu folgern. In der Schule werden Beweise manchmal etwas stiefmütterlich behandelt und erscheinen dann oft mysteriös und unverständlich.

Im Wort „Beweis“ steckt aber auch „weisen“; daher kann man einen Beweis als den Versuch verstehen, der Leserin einen Weg zu weisen, wie sie das Resultat einsehen kann. Das ist es, was wir mit den Beweisen in unserem Buch erreichen möchten. Ausgehend nur von elementaren, aus der Schule bekannten Rechenregeln leiten wir im Laufe des Buches die notwendigen Grundlagen her, um schließlich das Resultat von Agrawal, Kayal und Saxena beweisen zu können. Dabei bemühen wir uns stets, die Ideen klar herauszuarbeiten und die einzelnen logischen Schritte sehr deutlich zu machen. Aus diesem Grund verzichten wir manchmal auf die mathematisch elegantesten und kürzesten Argumente, um stattdessen eine tiefere Einsicht in das Ergebnis zu ermöglichen.

Wir hoffen, dass die Leserin sich im Laufe der Lektüre nicht nur an das Prinzip der mathematischen Beweisführung gewöhnt, sondern sie dann auch in der Lage ist, einfache Ergebnisse selbst herzuleiten. In späteren Kapiteln verlagern wir daher einzelne Beweisschritte gern in die Aufgaben – mit großzügigen Hinweisen.

Sätze, Hilfssätze und Definitionen

Der Begriff **Satz** bezeichnet in der Mathematik eine bewiesene Aussage. Um dabei schwierigere und tieferliegende Resultate von Hilfs- oder Zwischenergebnissen

zu unterscheiden, werden letztere als **Hilfssätze** (oder auch „Lemmata“) bezeichnet. In welche der beiden Kategorien wir eine gegebene Aussage einordnen, kann allerdings vom persönlichen Geschmack abhängen.

Ergebnisse, die auf einfache Art und Weise aus einem zuvor bewiesenen Satz folgen, werden wir naturgemäß als **Folgerungen** bezeichnen. Die Einführung einer mathematischen Schreibweise oder eines neuen Konzeptes wird **Definition** genannt.

Der Einfachheit halber sind Sätze, Hilfssätze, Definitionen, Aufgaben etc. in jedem Abschnitt durchgehend numeriert.

Mathematische Schreibweisen

Aus der Schule kennen wir die folgenden Zahlbereiche:

- die natürlichen Zahlen $\mathbb{N} = \{1, 2, 3, 4, \dots\}$;
- die ganzen Zahlen $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$;
- die rationalen Zahlen (Brüche) $\mathbb{Q} = \left\{ \frac{p}{q} : p, q \in \mathbb{Z}, q \neq 0 \right\}$;
- die reellen Zahlen $\mathbb{R}$.

Sind a und b Zahlen aus einem dieser Zahlbereiche, so schreiben wir $a \leq b$ bzw. $a < b$ für „a ist kleiner als b oder $a = b$“ bzw. „a ist echt kleiner als b“. Analog definieren wir $a \geq b$ und $a > b$.

Wir weisen ausdrücklich darauf hin, dass $\mathbb{N}$ nach unserer Konvention die Null nicht enthält. (Darüber besteht in der Mathematik keinerlei Konsens.) Wir definieren daher zusätzlich die Menge

$$\mathbb{N}_0 := \{a \in \mathbb{Z} : a \geq 0\}. \qquad (*)$$

Das Symbol $:=$ bedeutet dabei „definitionsgemäß gleich“. Es wird benutzt, um eine Abkürzung bzw. Bezeichnung einzuführen, und *nicht*, um eine Gleichheit zu behaupten. (Wir lesen $(*)$ also als „$\mathbb{N}_0$ bezeichne die Menge aller nicht-negativen ganzen Zahlen“.) Allgemeiner schreiben wir

$$\{x \in M : x \text{ hat die Eigenschaft } \dots\}$$

für die Menge aller Elemente von M, welche die angegebene Eigenschaft besitzen.

Grundsätzlich verwenden wir Bezeichnungen der Mengenlehre, die aus der Schule bekannt sind und betrachten dabei fast ausschließlich Mengen von Zahlen. Ist M eine Menge, so schreiben wir $x \in M$ für „x ist ein Element der Menge M“. Umgekehrt bedeutet $y \notin M$, dass y *nicht* in M liegt. Eine Menge N ist eine

Teilmenge der Menge M, falls jedes Element von N auch ein Element von M ist; in diesem Fall schreiben wir $N \subseteq M$. Zum Beispiel gilt

$$\mathbb{N} \subseteq \mathbb{N}_0 \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}.$$

Nach Definition ist M selbst eine Teilmenge von M. Ist $N \subseteq M$ und $N \neq M$, so heißt N eine **echte Teilmenge** von M, und wir schreiben $N \subsetneq M$. Das Symbol $\emptyset$ steht für die leere Menge. Mit $\#M$ bezeichnen wir die Anzahl der Elemente in M, zum Beispiel ist $\#\{2,4,6,8,10\} = 5$ und $\#\mathbb{N} = \infty$. (Das Zeichen ∞ heißt wie üblich „unendlich“).

Wir halten uns an die Standardnotation aus der Schule für Addition, Subtraktion, Multiplikation und Division sowie für die Darstellung von Potenzen. Gelegentlich lassen wir den Punkt bei der Multiplikation weg, z.B. $3x$ anstelle von $3 \cdot x$. Elementare Rechenregeln, wie z.B. das Distributivgesetz, werden ohne weitere Verweise benutzt.

Sind $x_1, \ldots, x_n$ Zahlen, so verwenden wir die übliche Summen- und Produktschreibweise:

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + \cdots + x_n; \quad \prod_{i=1}^{n} x_i = x_1 \cdot x_2 \cdots x_n.$$

Zum Beispiel ist $\sum_{i=1}^{n} i = 1 + 2 + \cdots + n$ und $\sum_{i=1}^{n} \frac{1}{i} = \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n}$.

Für jede natürliche Zahl n ist $n!$ („n Fakultät“) definiert durch

$$n! := \prod_{i=1}^{n} i \quad \Big(= 1 \cdot 2 \cdots (n-1) \cdot n \Big).$$

Es ist also etwa $1! = 1$, $3! = 6$ und $5! = 120$. Außerdem setzen wir $0! := 1$.

Wir erinnern an die Potenzregeln: Sind $a, b > 0$ und $x, y \in \mathbb{R}$, so gilt

$$a^x \cdot a^y = a^{x+y}, \quad (ab)^x = a^x \cdot b^x \quad \text{und} \quad (a^x)^y = a^{x \cdot y}.$$

Anstelle von $a^{(x^y)}$ schreiben wir a^{x^y}. Im Allgemeinen ist das *nicht* dasselbe wie $(a^x)^y$, zum Beispiel gilt $2^{3^2} = 512$ und $(2^3)^2 = 64$. Definitionsgemäß ist $a^0 = 1$. Wir sagen, dass eine natürliche Zahl n eine **echte Potenz** von $a \in \mathbb{N}$ ist, falls es ein $b \in \mathbb{N}$ gibt mit $b \geq 2$ und $n = a^b$.

Den Logarithmus der Zahl x zur Basis 2 bezeichnen wir mit $\log x$, d.h. $\log x$ ist diejenige Zahl ℓ mit der Eigenschaft $2^\ell = x$, etwa $\log 2 = 1$ und $\log 8 = 3$. Ab und zu verwenden wir auch den **natürlichen Logarithmus** $\ln x$, also den Logarithmus zur Basis e, wobei e die Eulersche Konstante ist. D.h. $\ln x$ ist diejenige Zahl ℓ mit der Eigenschaft $e^\ell = x$.

Mit „$f : \mathbb{N} \to \mathbb{R}$“ meinen wir „$f$ ist eine Funktion von $\mathbb{N}$ nach $\mathbb{R}$“. Das bedeutet, dass f jedem $n \in \mathbb{N}$ eine reelle Zahl, $f(n)$ genannt, zuordnet. Zum Beispiel definiert $f(n) := \log n$ eine Funktion von $\mathbb{N}$ nach $\mathbb{R}$.

Ist $x \in \mathbb{R}$, so schreiben wir $|x|$ für den **Betrag** von x. (Also $|x| = x$ für $x \geq 0$ und andernfalls $|x| = -x$.) Wir bezeichnen außerdem mit $\lfloor x \rfloor$ die größte ganze Zahl n mit $n \leq x$. Zum Beispiel ist $\lfloor \frac{3}{2} \rfloor = 1$. Analog steht $\lceil x \rceil$ für die kleinste ganze Zahl n mit $n \geq x$. (Siehe Aufgabe 1.1.7.)

Alle weiteren Begriffe und Schreibweisen werden zu gegebener Zeit eingeführt und mit Beispielen versehen – wir verweisen auch auf den Index und das Notationsverzeichnis am Ende des Buches.

Teil I

Grundlagen

Kapitel 1

Natürliche Zahlen und Primzahlen

Im ganzen Buch beschäftigen wir uns mit natürlichen Zahlen und der Frage, welche natürlichen Zahlen Primzahlen sind und welche nicht. Daher beginnen wir damit, uns an einige ihrer grundlegenden Eigenschaften zu erinnern und diese sorgfältig herzuleiten.

Zunächst behandeln wir das Prinzip der vollständigen Induktion und definieren Teilbarkeit. Auf dieser Basis können wir den Euklidischen Algorithmus erarbeiten und anwenden und schließlich beweisen, dass jede natürliche Zahl sich eindeutig als Produkt von Primzahlpotenzen schreiben lässt („Fundamentalsatz der Arithmetik“). Gegen Ende dieses ersten Kapitels betrachten wir den ältesten bekannten Primzahltest – das sogenannte Sieb des Eratosthenes – und zeigen, dass es unendlich viele Primzahlen gibt.

1.1 Die natürlichen Zahlen

Wir haben ein intuitives Verständnis dafür, was natürliche Zahlen sind – vielleicht abgesehen von der Frage, ob die Zahl 0 dazugehört oder nicht. Von diesem Standpunkt aus möchten wir nun einige wichtige Eigenschaften besprechen, welche die natürlichen Zahlen von den anderen Zahlbereichen unterscheiden.

Die Zahlen in $\mathbb{N}$ stellen wir uns dabei vor als diejenigen, die wir ganz naiv zum Zählen benötigen – das heißt, wann immer wir Objekte zählen, soll ihre Anzahl ein Element von $\mathbb{N}$ sein. Da Zählen erst dann sinnvoll ist, wenn etwas zum Zählen da ist, gehört für uns vor diesem Hintergrund 0 *nicht* zu $\mathbb{N}$. Die erste bzw. kleinste natürliche Zahl ist demnach 1.

Wenn wir $\mathbb{N}$ mit anderen Zahlbereichen vergleichen, fallen uns zunächst viele Dinge auf, die in den natürlichen Zahlen *nicht* möglich sind. Wir können sie

nicht beliebig voneinander subtrahieren, sie nicht beliebig durcheinander teilen und erst recht keine beliebigen Wurzeln ziehen. Diesen Defekten steht aber eine Eigenschaft gegenüber, die $\mathbb{N}$ den anderen Zahlbereichen voraus hat und die viele nützliche Konsequenzen hat: das „Wohlordnungsprinzip“.

1.1.1. Wohlordnungsprinzip.
Jede nicht-leere Teilmenge von $\mathbb{N}$ enthält ein kleinstes Element.

Wir sehen sofort, dass das Wohlordnungsprinzip für $\mathbb{Z}$, $\mathbb{Q}$ und $\mathbb{R}$ nicht gilt. Ist nämlich a eine beliebige ganze Zahl, so ist $a-1$ eine ganze Zahl mit $a-1<a$; also besitzt $\mathbb{Z}$ kein kleinstes Element. Allerdings gilt das Wohlordnungsprinzip für *nach unten beschränkte* Teilmengen von $\mathbb{Z}$; siehe Aufgabe 1.1.7. Von dieser Tatsache werden wir häufig Gebrauch machen.

Umgekehrt ist das Prinzip für die natürlichen Zahlen sofort intuitiv einsichtig. Ist nämlich A eine nicht-leere Teilmenge von $\mathbb{N}$, so enthält A ja irgendein Element $n_0 \in A$. Ist n_0 nicht kleinstes Element von A, so gibt es ein weiteres Element $n_1 \in A$ mit $n_1 < n_0$. Ist n_1 nicht kleinstes Element von A, so gibt es ein noch kleineres Element n_2, und so weiter. Nun gibt es aber nur $n_0 - 1$ natürliche Zahlen, die kleiner als n_0 sind, also muss dieser Prozess zwangsläufig irgendwann ein Ende finden und wir ein kleinstes Element von A erhalten.

Das ist streng genommen kein Beweis – einen solchen können wir nicht führen, da wir keine formale Definition der natürlichen Zahlen aufgestellt haben. Stattdessen setzen wir das Wohlordnungsprinzip als „Axiom“ voraus, also als einleuchtenden Grundsatz, welchen wir ohne Beweis als wahr annehmen. (Siehe aber auch Aufgabe 1.1.18.)

Das Prinzip des kleinsten Verbrechers

Das Wohlordnungsprinzip ist unter anderem deshalb sehr nützlich, weil es uns ein Werkzeug in die Hand gibt, um Aussagen für alle Zahlen aus $\mathbb{N}$ zu beweisen. Im nächsten Satz illustrieren wir diese Idee. (Zum Namen siehe Anmerkung 1.1.15.)

1.1.2. Satz (Irrationalität von $\sqrt{2}$)**.**
Es sei n eine natürliche Zahl. Dann gibt es keine natürliche Zahl m mit $2m^2 = n^2$.

Beweis. Wir nehmen an, die Aussage sei falsch, d.h. es gebe natürliche Zahlen n und m mit $2m^2 = n^2$, und führen das zum Widerspruch. Nach unserer Annahme ist die Menge

$$A := \{n \in \mathbb{N} : \text{ es gibt ein } m \in \mathbb{N} \text{ mit } 2m^2 = n^2\}$$

nicht leer, sie hat also aufgrund des Wohlordnungsprinzips ein kleinstes Element n_0. Nach Definition von A gibt es daher ein $m_0 \in \mathbb{N}$ mit $2m_0^2 = n_0^2$; insbesondere ist n_0^2 eine gerade Zahl. Wegen $1 < 2$ ist außerdem $m_0 < n_0$.

Das Quadrat einer ungeraden Zahl ist nie gerade (siehe Aufgabe 1.1.6), also ist auch n_0 selbst gerade. Deshalb ist $n_0 = 2n'$ für ein geeignetes $n' \in \mathbb{N}$. Nun gilt

$$2m_0^2 = n_0^2 = (2n')^2 = 4n'^2,$$

und damit ist $m_0^2 = 2n'^2$. Also ist $m_0 \in A$. Das ist der gewünschte Widerspruch, da $m_0 < n_0$ ist und n_0 als kleinstes Element von A gewählt war. ■

Die Grundidee des vorangehenden Beweises wird als das Beweisprinzip des **unendlichen Abstiegs** oder auch das **Prinzip des kleinsten Verbrechers** bezeichnet. Wir nehmen dabei an, dass es eine natürliche Zahl gibt, für die die betrachtete Aussage falsch ist. Nach dem Wohlordnungsprinzip gibt es dann einen „kleinsten Verbrecher“, also eine kleinste Zahl, die unsere Aussage verletzt. Können wir folgern, dass es einen noch kleineren „Verbrecher“ geben muss, so erhalten wir einen Widerspruch.

Vollständige Induktion

Das Prinzip des kleinsten Verbrechers ist eng verwandt mit dem Beweisprinzip der **vollständigen Induktion**. Der folgende Satz formuliert dieses Prinzip allgemein; weiter unten führen wir es dann an einem Beispiel vor.

1.1.3. Satz (Vollständige Induktion)**.**
Es sei $M \subseteq \mathbb{N}$ eine Menge natürlicher Zahlen. Ferner gelte:

(a) Die Zahl 1 ist ein Element von M, und

(b) ist n eine natürliche Zahl mit $n \in M$, so ist auch der Nachfolger $n+1$ ein Element von M.

Dann ist $M = \mathbb{N}$, d.h. jede natürliche Zahl liegt in M.

Beweis. Wir nehmen $M \neq \mathbb{N}$ an und leiten mit Hilfe des Prinzips des kleinsten Verbrechers einen Widerspruch her. Das Komplement $A := \{n \in \mathbb{N} : n \notin M\}$ von M ist dann nämlich eine nicht-leere Menge natürlicher Zahlen und besitzt nach dem Wohlordnungsprinzip ein kleinstes Element n_0. Mit Voraussetzung (a) ist $1 \notin A$, also insbesondere $n_0 \neq 1$. Daher ist auch $m := n_0 - 1$ eine natürliche Zahl. Da n_0 das kleinste Element von A ist, ist m kein Element von A, also $m \in M$. Laut Voraussetzung (b) ist dann aber auch $m + 1 = n_0$ ein Element von M, und das ist ein Widerspruch. ■

Anschaulich können wir uns zur Erklärung des Induktionsprinzips eine (unendliche) Reihe von Dominosteinen vorstellen. Stoßen wir den ersten Dominostein an, und sind die Steine so aufgestellt, dass jeder beim Umfallen den nächsten anstößt, so besagt das Prinzip der vollständigen Induktion, dass dann jeder der Dominosteine irgendwann umfällt. Das entspricht sicherlich unserer Intuition!

Um durch vollständige Induktion eine Aussage über natürliche Zahlen zu beweisen, müssen wir Folgendes zeigen:

- Die Aussage ist für die Zahl 1 erfüllt (**Induktionsanfang**), und
- gilt sie für eine natürliche Zahl n, so auch für $n+1$ (**Induktionsschritt**).

Dann folgt aus Satz 1.1.3, dass die Aussage für jede natürliche Zahl n erfüllt ist.

1.1.4. Beispiel. Wir zeigen: Für alle natürlichen Zahlen n ist n^3-n ein Vielfaches von 3. (Das heißt, es gibt ein $m \in \mathbb{Z}$ mit $n^3 - n = 3m$.)

Beweis. Es gilt

$$1^3 - 1 = 1 - 1 = 0 = 3 \cdot 0,$$

die Behauptung ist also im Fall $n = 1$ wahr. Dies liefert den Induktionsanfang.

Jetzt nehmen wir uns eine beliebige natürliche Zahl n her, für die die Behauptung stimmt; es sei also $n^3 - n = 3m$ für ein geeignetes $m \in \mathbb{Z}$. Das wird als **Induktionsvoraussetzung** bezeichnet.

Wir müssen nun zeigen, dass die Behauptung auch für $n+1$ erfüllt ist, also dass $(n+1)^3 - (n+1)$ ein Vielfaches von 3 ist. Dazu multiplizieren wir $(n+1)^3$ aus (siehe auch Satz 1.1.5) und erhalten

$$\begin{aligned}(n+1)^3 - (n+1) &= n^3 + 3n^2 + 3n + 1 - n - 1 \\ &= n^3 - n + 3(n^2 + n) = 3m + 3(n^2 + n) = 3(m + n^2 + n).\end{aligned}$$

Dabei haben wir in der vorletzten Gleichung die Induktionsvoraussetzung verwendet. Also ist $(n+1)^3 - (n+1)$ ein Vielfaches von 3, wie behauptet.

Damit ist die Induktion abgeschlossen, und die Behauptung gilt in der Tat für alle natürlichen Zahlen n. ■

Wir möchten noch auf einige Varianten des Induktionsprinzips hinweisen, die wir gelegentlich verwenden.

(a) Manchmal ist es zweckmäßig, den Induktionsschritt nicht von n nach $n+1$, sondern von $n-1$ nach n zu vollziehen.

(b) Auch Aussagen, die für alle natürlichen Zahlen ab einer bestimmten Größe gelten, können mit vollständiger Induktion bewiesen werden. Man beginnt dann anstatt mit 1 mit dem kleinsten Element, auf das sich die Aussage bezieht, alles andere ist genau wie oben.

Aufgaben

1.1.6. Aufgabe (!). Zur Erinnerung: Eine natürliche Zahl n heißt **gerade**, falls es eine natürliche Zahl m gibt mit $n = 2m$. Die Zahl n heißt **ungerade**, falls es eine natürliche Zahl m gibt mit $n = 2m - 1$.

(a) Zeige, dass jede natürliche Zahl entweder gerade oder ungerade ist, aber nicht beides. (*Hinweis:* Nach dem Wohlordnungsprinzip gibt es eine kleinste Zahl m mit $2m \geq n$.)

(b) Zeige: Das Produkt zweier gerader Zahlen ist gerade, und das Produkt zweier ungerader Zahlen ist ungerade.

1.1.7. Aufgabe (!). Es sei M eine nicht-leere Teilmenge von $\mathbb{Z}$. Dann heißt M **nach oben beschränkt** bzw. **nach unten beschränkt**, falls es eine ganze Zahl $K \in \mathbb{Z}$ gibt mit $x \leq K$ bzw. $x \geq K$ für alle $x \in M$.

Zeige: Ist M nach unten beschränkt, so hat M ein kleinstes Element; ist M nach oben beschränkt, so hat M ein größtes Element.

(*Hinweis:* Für den ersten Teil betrachte die Menge $\{1 + x - K : x \in M\}$ und wende das Wohlordnungsprinzip an. Für den zweiten Teil betrachte die Menge $\{-x : x \in M\}$ und wende den ersten Teil an.)

Als Zusatz überlege, ob diese Aussagen auch für Teilmengen von $\mathbb{Q}$ oder $\mathbb{R}$ gelten.

1.1.8. Aufgabe (!). Beweise durch vollständige Induktion, dass für alle $n \in \mathbb{N}$ gilt:

(a) $2^n \geq 2n$.

(b) $\displaystyle\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$ („Gaußsche Summenformel“).

(c) $\displaystyle\sum_{k=0}^{n-1} x^k = \frac{1-x^n}{1-x}$ für alle reellen Zahlen $x \neq 1$.

(d) $\displaystyle\sum_{k=0}^{n-1} (k+1) \cdot x^k = \frac{nx^{n+1} - (n+1)x^n + 1}{(1-x)^2}$ für alle reellen Zahlen $x \neq 1$.

(e) $\displaystyle\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$.

(f) $\displaystyle\sum_{k=0}^{n} (k \cdot k!) = (n+1)! - 1$.

(g) $\displaystyle\sum_{k=1}^{n} \frac{1}{k(k+1)} = \frac{n}{n+1}$.

1.1.9. Aufgabe. (a) Zeige: Für alle $n \in \mathbb{N}$ ist $n^5 - n$ ein Vielfaches von 5.

(b) Ist für alle $n \in \mathbb{N}$ die Zahl $n^4 - n$ ein Vielfaches von 4? Falls nicht, gib ein Gegenbeispiel und erläutere, woran der Beweis aus (a) hier scheitert.

1.1.10. Aufgabe. Gegeben seien n paarweise nicht parallele Geraden $g_1, ..., g_n$ in der Ebene. (D.h. sind i und j zwei verschiedene Zahlen zwischen 1 und n, so setzen wir g_i und g_j als nicht parallel voraus.) Weiter gebe es keinen Punkt der Ebene, in dem sich mehr als zwei dieser Geraden schneiden. In wie viele Teile wird die Ebene von diesen n Geraden zerteilt? Entwickle eine Idee, stelle eine Behauptung auf und beweise sie mit vollständiger Induktion. Was ändert sich, wenn doch zwei oder mehr Geraden parallel sein dürfen?

1.1.11. Aufgabe (!). (a) Begründe anhand der intuitiven Definition von Binomialkoeffizienten, wieso die rekursive Formel (1.1) gilt.

(b) Begründe außerdem, weshalb

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{1.2}$$

gilt für alle $k, n \in \mathbb{N}_0$ mit $k \leq n$.

(c) Beweise die Formel (1.2) durch Induktion mit Hilfe der rekursiven Formel (1.1).

1.1.12. Aufgabe (!). Seien $n, k, \ell \in \mathbb{N}_0$. Zeige:

(a) Es gilt $\binom{n+\ell}{k} \geq \binom{n}{k}$.

(b) Es gilt $\binom{n+\ell}{k+\ell} \geq \binom{n}{k}$.

(c) Die „mittleren“ Binomialkoeffizienten $\binom{2n}{n}$ wachsen mindestens exponentiell:

$$\binom{2n}{n} \geq 2^n.$$

(*Hinweis:* Für die ersten beiden Teile ist es nützlich, sich die Behauptungen erst einmal am Pascalschen Dreieck klarzumachen. Für den dritten Teil verwende Induktion, die rekursive Formel für Binomialkoeffizienten und die ersten beiden Teilaufgaben.)

1.1.13. Aufgabe (!). Es seien $n, k \in \mathbb{N}_0$. Die Anzahl der Möglichkeiten, ohne Berücksichtigung der Reihenfolge bis zu k (nicht notwendigerweise verschiedene) Zahlen zwischen 1 und n auszuwählen, werde mit $a(n, k)$ bezeichnet. (Dabei zählt es als eine Möglichkeit, *gar keine* Zahlen auszuwählen; z.B. ist $a(n, 0) = 1$ und $a(n, 1) = n + 1$ für alle n.)

(a) Begründe, dass $a(n, m)$ für $n, m \geq 1$ die Rekursionsformel

$$a(n, m) = a(n-1, m) + a(n, m-1)$$

erfüllt.

(b) Beweise durch Induktion, dass

$$a(n, m) = \binom{n+m}{m}$$

gilt.

1.1.14. Aufgabe. Zeige, dass die Folge f_n der Fibonacci-Zahlen durch folgende Formel gegeben ist:

$$f_n = \frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{2^n \cdot \sqrt{5}}.$$

(*Hinweis:* Verwende Variante (c) des Induktionsprinzips.)

Weiterführende Übungen und Anmerkungen

1.1.15. Satz 1.1.2 ist äquivalent zu der bekannten Tatsache, dass $\sqrt{2}$ irrational ist. In der Tat ist ja $2m^2 = n^2$ nur eine andere Art, die Gleichung

$$\left(\frac{n}{m}\right)^2 = 2$$

zu schreiben. Wir können diese Aussage auch geometrisch interpretieren: Es gibt kein Quadrat, für das sowohl die Seitenlänge a als auch die Länge d der Diagonalen natürliche Zahlen sind. (Sonst wäre nach Pythagoras $2a^2 = d^2$.)

1.1.16. Ist n eine natürliche Zahl, so sagen wir, dass $n+1$ der **Nachfolger** von n ist. Es gilt:

(I) 1 ist eine natürliche Zahl.

(II) Jede natürliche Zahl besitzt genau einen Nachfolger.

(III) Es gibt keine natürliche Zahl, deren Nachfolger 1 ist, aber jede natürliche Zahl $n \neq 1$ ist selbst Nachfolger einer natürlichen Zahl.

(IV) Verschiedene natürliche Zahlen haben verschiedene Nachfolger.

(V) Ist M eine Teilmenge von $\mathbb{N}$, die 1 enthält und mit jedem Element auch dessen Nachfolger, so ist $M = \mathbb{N}$.

Hierbei ist (V) genau das in Satz 1.1.3 bewiesene Induktionsprinzip. Die Eigenschaften (I) bis (V) werden als **Peano-Axiome** bezeichnet, nach dem italienischen Mathematiker Guiseppe Peano.

Es stellt sich heraus, dass die Peano-Axiome die natürlichen Zahlen eindeutig beschreiben, d.h. es gibt (bis auf Umbenennung der Elemente) keine andere Menge mit diesen Eigenschaften. Aus diesem Grund können sie verwendet werden, um die natürlichen Zahlen zu *definieren*; dies ist der heute in der Mathematik übliche Weg. Wir haben uns entschieden, statt dem Induktionsprinzip mit dem Wohlordnungsprinzip zu beginnen, da das für Einsteiger vielleicht intuitiv einsichtiger ist.

1.1.17. Aufgabe. Wir weisen darauf hin, dass die Peano-Axiome keine Aussagen über die Grundrechenarten enthalten: Sie erfordern nur, dass für jede natürliche Zahl n der Nachfolger $n+1$ definiert ist.

Zeige, dass mit Hilfe dieser Nachfolgerfunktion die Summen $n+m$, $n \cdot m$ und n^m rekursiv definiert werden können.

1.1.18. Aufgabe. Zeige, dass das Wohlordnungsprinzip aus dem Induktionsprinzip (und damit aus den Peano-Axiomen) folgt.

Kapitel 2

Algorithmen und Komplexität

In diesem Kapitel beschäftigen wir uns mit **Algorithmen**: automatischen Verfahren zur Lösung von Problemen. Wir beginnen damit, den Begriff des Algorithmus selbst zu erklären – zumindest so genau wie für unsere Zwecke nötig – und machen uns anhand zahlreicher Beispiele mit seinen Eigenschaften vertraut. Daraufhin erläutern wir, was es bedeutet, dass ein mathematisches Problem **algorithmisch lösbar** ist. Ein besonders wichtiger Gesichtspunkt ist dabei *Effizienz*, ein Maß dafür, wie „praktikabel" ein Algorithmus ist. Wir unterscheiden zwischen effizient lösbaren und effizient verifizierbaren Problemen und besprechen, welche Methoden es zur Verkürzung der Laufzeit von Algorithmen gibt.

2.1 Algorithmen

Was ist ein Algorithmus?

Seit ihrem Anbeginn sucht die Mathematik nach Methoden, mit deren Hilfe die Lösung eines Problems möglichst schnell und sozusagen „automatisch" gefunden werden kann. Solche Verfahren werden als **Algorithmen** bezeichnet. Die schriftliche Addition, Multiplikation und Division, die wir schon in der Grundschule lernen, sind Beispiele von Algorithmen, ebenso wie das Sieb des Eratosthenes und der Euklidische Algorithmus, denen wir im letzten Kapitel begegnet sind. Mit der Entwicklung des Computers haben die Suche nach Algorithmen und deren systematische Betrachtung seit der zweiten Hälfte des zwanzigsten Jahrhunderts eine noch größere Bedeutung gewonnen.

Was aber ist ein Algorithmus? Wir stellen uns darunter eine Anleitung vor, die wir – wie ein Kochrezept – nur Schritt für Schritt befolgen müssen, um das vorgegebene Problem zu lösen. Zur Erläuterung betrachten wir zwei sehr unterschiedliche „Algorithmen". Der erste ist in der Tat ein Kochrezept (welches wir im Eigenversuch getestet haben).

ALGORITHMUS PFANNKUCHEN

Eingabe: Ein Ei, eine Tasse Mehl, eine Tasse Milch, eine Prise Salz, ein Teelöffel Sonnenblumenöl.

1. Verrühre das Ei, das Mehl und die Milch in einer Schüssel zu Pfannkuchenteig. Füge das Salz hinzu.
2. Erhitze das Öl in einer Bratpfanne auf mittelhoher Flamme, bis es brutzelt.
3. Fülle den Pfannkuchenteig in die Pfanne und schwenke diese, bis der Teig verteilt ist.
4. Wende den Pfannkuchen nach 2-3 Minuten.
5. Nach weiteren 2 Minuten ist der Pfannkuchen zum Verzehr bereit.

Der zweite „Algorithmus“ ist eine Anleitung zum Verfassen eines Bestseller-Krimis.

ALGORITHMUS BESTSELLER-KRIMI

1. Erfinde eine kantige, aber sympathische Hauptfigur, vorzugsweise eine Privatdetektivin oder Polizeikommissarin.
2. Erfinde außerdem eine (fast) perfekte Straftat.
3. Entwirf eine Handlung, in welcher diese Straftat von der Hauptfigur durch Ermittlungen und evtl. eine Reihe glücklicher Zufälle aufgeklärt wird.
4. Beschreibe diese Handlung auf unterhaltsame und spannende Weise in einem Roman.

Es fällt sofort auf, dass diese beiden Anleitungen von sehr verschiedener Natur sind. Während die erste die einzelnen Schritte und ihre Abfolge klar beschreibt, lässt die zweite viele Details offen. Auch wenn diese Beispiele überspitzt erscheinen mögen, veranschaulichen sie genau die wesentliche Eigenschaft von Algorithmen: Ein solcher darf von der ausführenden Person nicht erwarten, eigene Denkarbeit zu leisten und kreativ zu sein. Die korrekte Befolgung der Anweisungen

sollte stets – unabhängig von persönlicher Fähigkeit oder Begabung – dasselbe (korrekte) Ergebnis liefern.

Insofern stimmt die Leserin hoffentlich mit uns überein, dass das Pfannkuchen-Rezept sich (sofern das für nicht-mathematische Beispiele überhaupt möglich ist) als Algorithmus qualifiziert, während die Anleitung zum Besteller-Schreiben hinter diesen Ansprüchen weit zurückbleibt.

Wir formulieren nun etwas genauer, welche Forderungen ein Algorithmus erfüllen soll:

(a) Er hat eine endliche Beschreibung;

(b) er besteht nur aus „elementaren“ Schritten, seine Durchführung erfordert insbesondere keine Erfindungskraft;

(c) er darf zwar beliebig große, zu jedem Zeitpunkt aber nur **endliche** Ressourcen (Papier, Tinte, Speicherplatz, ...) benötigen;

(d) zu jedem Zeitpunkt ist der nächste auszuführende Schritt eindeutig bestimmt (**Determinismus**).

Wir können stattdessen ebenfalls sagen: *Ein Algorithmus ist ein Verfahren, das sich in einer gängigen Programmiersprache auf einem Computer implementieren lässt.* Selbstverständlich ist das keine *Definition* im formalen Sinne der Mathematik. Mit etwas Aufwand ist es zwar möglich, den Algorithmusbegriff mathematisch zu erfassen (siehe unten), aber für dieses Buch begnügen wir uns mit obiger informeller Beschreibung und füllen sie im Folgenden etwas mit Leben.

Beispiele und Erläuterungen zum Algorithmusbegriff

Wir beginnen mit einem Verfahren, das schon in der Grundschule gelehrt wird – der schriftlichen Addition. Der Einfachheit halber formulieren wir es für Zahlen im **Binärsystem**. (Siehe Anmerkung 2.1.4.)

b \ a	0	1
0	0	1
1	1	10

(a) Übertrag 0

b \ a	0	1
0	1	10
1	10	11

(b) Übertrag 1

Abbildung 2.1. Binäre Additionstafeln, zur Verwendung im Algorithmus ADDITION.

Algorithmus ADDITION

Eingabe: Zwei natürliche Zahlen m und n, dargestellt in Binärschreibweise.

1. Schreibe beide Zahlen so untereinander, dass die letzten beiden Ziffern von m und n übereinanderstehen, ebenso die vorletzten beiden Ziffern und so weiter. Ziehe einen Strich unter die beiden Zahlen.

2. Haben die Zahlen ungleiche Stellenanzahl, so ergänze die kürzere durch führende Nullen, bis beide Zahlen die gleiche Anzahl s von Stellen haben.

3. Notiere in einem separaten „Übertragskästchen“ eine Null und setze $j := 1$.

4. Es sei a die j-te Stelle von m und b die j-te Stelle von n, jeweils von *rechts* gezählt. Lies aus der dem Wert im Übertragskästchen entsprechenden Tabelle aus Abbildung 2.1 den Wert in der a-ten Spalte und b-ten Zeile ab. Wir nennen diese Zahl k.

5. Trage die letzte Ziffer der Zahl k an der j-ten Stelle unterhalb der Zahlen m und n ein.

6. Ist k einstellig, so ersetzen wir die Zahl im Übertragskästchen durch eine Null, andernfalls durch eine Eins.

7. Ist $j \neq s$ (d.h. wir sind noch nicht ganz links angekommen), dann ersetzen wir j durch $j + 1$ und kehren zu **4.** zurück.

8. Andernfalls notieren wir die Ziffer aus dem Übertragskästchen als erste Stelle vor dem Rest unseres Ergebnisses und sind fertig.

Die Leserin sei dazu aufgefordert, das Verfahren an einem Beispiel selbst auszuführen. Für die Addition der Zahlen 3 und 6 sind die einzelnen Zwischenschritte in Abbildung 2.2 dargestellt. Im Binärsystem entspricht der Zahl 3 die Ziffern-

	$\boxed{0}$	$\boxed{0}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$
11	011	011	011	011	011
110	110	110	110	110	110
		1	01	001	1001

Abbildung 2.2. Berechnung von $3 + 6$ mit Hilfe von ADDITION.

folge 11 und der Zahl 6 die Ziffernfolge 110; das Ergebnis 1001 ist in der Tat die Binärdarstellung der Zahl 9.

Auch wenn es sich etwas umständlich liest, haben wir nichts anderes als das übliche Verfahren der schriftlichen Addition beschrieben. (Wir haben übrigens das Binär- nur deshalb anstelle des Dezimalsystems gewählt, weil dann die Additionstafeln überschaubarer bleiben.) Diese Anleitung genügt unseren Kriterien für einen Algorithmus. Die Beschreibung – bestehend aus dem Algorithmus zusammen mit den Additionstafeln aus Abbildung 2.1 – ist sicherlich endlich. Jeder Schritt enthält eine Anweisung, die wir guten Gewissens als „elementar“ bezeichnen können, und verwendet außerdem nur endlich viele Ressourcen. Zu guter Letzt ist die Abfolge der Schritte eindeutig bestimmt; unser Verfahren ist also deterministisch.

Vielleicht möchten wir aber nicht nur zwei Zahlen addieren, sondern gleich eine ganze Liste. Wir könnten das übliche Verfahren für die schriftliche Addition mehrerer Zahlen ausformulieren, aber es geht noch einfacher:

Algorithmus ADDITION-VIELE

Eingabe: Eine Liste von mindestens einer und höchstens endlich vielen natürlichen Zahlen.

1. Enthält die gegebene Liste nur eine Zahl, so sind wir fertig.

2. Andernfalls wende den Algorithmus ADDITION auf die letzten beiden Zahlen der Liste an.

3. Ersetze die letzten beiden Zahlen der Liste durch die in Schritt **2.** errechnete Zahl.

4. Kehre zu Schritt **1.** zurück.

Hier sehen wir eine wichtige Eigenschaft von Algorithmen – sie lassen sich kombinieren. Das heißt, wir können in einer Algorithmenbeschreibung andere Algorithmen verwenden, die wir bereits formuliert haben. Das macht das Leben

einfacher, denn wie das Beispiel der schriftlichen Addition zeigt, kann es ziemlich aufwendig sein, selbst einfache Verfahren detailliert in ihre Einzelschritte zu zerlegen.

Für den Rest des Buches lassen wir daher die Grundrechenarten in den natürlichen Zahlen als elementare Anweisungen zu, da für diese wohlbekannte Algorithmen existieren (siehe Aufgabe 2.1.1). Ebenso werden wir bereits formulierte Algorithmen in späteren Kapiteln wiederverwenden.

Ein weiteres Beispiel möchten wir noch erwähnen; sei dazu $n \in \mathbb{N}$. Es ist eine bekannte Tatsache (siehe etwa [Lo]), dass $n \cdot \pi$ und $n \cdot e$ selbst keine natürlichen Zahlen sind – unabhängig von der Wahl von n. Aber wie steht es mit der Zahl $n \cdot (\pi + e)$? Um zu versuchen, eine natürliche Zahl n zu finden, für die auch $n \cdot (\pi + e) \in \mathbb{N}$ ist, könnten wir folgenden „Algorithmus“ verwenden:

ALGORITHMUS N*(PI+E)

1. Setze $n := 1$.

2. Berechne die Zahl $a_n := n \cdot (\pi + e)$.

3. Falls die errechnete Zahl a_n ganzzahlig ist, sind wir fertig.

4. Andernfalls kehre zu Schritt **2.** zurück, wobei n durch die Zahl $n + 1$ ersetzt wird.

Auf den ersten Blick sieht das wie ein Algorithmus aus. Bei genauerer Betrachtung fällt aber auf, dass wir zum Beispiel nichts dazu gesagt haben, *wie* die Rechnung in Schritt **2.** durchgeführt werden soll. Natürlich können wir mit Hilfe eines Computers oder Taschenrechners (und mit viel Aufwand auch per Hand) die Zahl a_n mit beliebiger Genauigkeit, d.h. gerundet bis auf eine vorgegebene Zahl von Dezimalstellen, ausrechnen. Aber wenn alle so berechneten Nachkommastellen gleich Null sind, so heißt das noch lange nicht, dass $a_n \in \mathbb{N}$ gilt; es könnte sein, dass die gewählte Genauigkeit nicht ausreicht! (Man berechne als Beispiel die Zahl $a_{56602103}$ mit einem Taschenrechner.) Also ist eine solche Rechnung nicht ausreichend, um Schritt **3.** korrekt auszuführen. In Anbetracht dieser Überlegungen ist hier die Forderung nach „elementaren“ Schritten verletzt: Das Verfahren stellt keinen Algorithmus dar.

Dieses Beispiel führt vor Augen, dass wir bei der Formulierung von Algorithmen etwas vorsichtig sein müssen. Nichtsdestotrotz hoffen wir, die Leserin davon überzeugt zu haben, dass wir Algorithmen *stets als solche erkennen können.* Ist bei jedem der angegebenen Schritte eindeutig klar, dass und wie dieser automatisch ausgeführt werden kann, so haben wir einen Algorithmus vor uns – sonst nicht. Wir ermutigen explizit dazu, sich bei jeder neuen Algorithmenbeschreibung

zu vergewissern, dass die angegebenen Schritte in der Tat entweder elementar sind oder nur die Ausführung eines bereits bekannten Verfahrens erfordern.

Noch eine Bemerkung zum Determinismus, also Teil (d) der zu Anfang des Abschnittes formulierten Bedingungen: Sicherlich erscheint diese Forderung auf den ersten Blick einleuchtend. Allerdings kann es sinnvoll sein, sie etwas abzuschwächen, indem wir **Zufallsentscheidungen** erlauben. Die Vorteile solcher **randomisierter Algorithmen** lernen wir in Abschnitt 2.5 kennen.

Formale Definitionen des Algorithmusbegriffs

Die Suche nach Algorithmen erhielt Ende des neunzehnten und Beginn des zwanzigsten Jahrhunderts eine neue Bedeutung. Die ersten „echten“ Computer wurden zwar erst Mitte des zwanzigsten Jahrhunderts, aber die im Rahmen der industriellen Revolution erfolgte Mechanisierung verschiedenster Prozesse hatte bereits begonnen, den Blick von Mathematikerinnen für die algorithmische Lösung von Problemen zu schärfen.

David Hilbert, einer der führenden mathematischen Denker seiner Zeit, stellte daher in den frühen Jahren des zwanzigsten Jahrhunderts ein ambitioniertes Programm auf. Er wollte die Mathematik ein für alle Mal auf eine formale Grundlage stellen, die keine Widersprüche enthielte und in der jede wahre mathematische Aussage auch beweisbar sei. Insbesondere suchte Hilbert nach einer Methode (also einem Algorithmus), mit welcher die Wahrheit einer beliebigen mathematischen Aussage entschieden werden kann. (Dies ist bekannt als **Hilberts Entscheidungsproblem**.)

Hilberts Fragen motivierten in den dreißiger Jahren gleich mehrere Mathematiker (darunter Alan Turing und Alonso Church), den Begriff des Algorithmus mathematisch zu formalisieren. Obwohl die dabei entstandenen Konzepte auf den ersten Blick wenig gemein haben, stellte sich schnell heraus, dass sie äquivalent sind. Dasselbe gilt für alle Algorithmusdefinitionen, die seitdem vorgestellt wurden. Insbesondere sind die Verfahren, die sich in einer der heute üblichen Programmiersprachen implementieren lassen, genau dieselben, welche in Turings Maschinenmodell beschrieben werden können. Aus diesem Grund geht man heute davon aus, dass diese vielfältigen Definitionen tatsächlich genau das widerspiegeln, was wir intuitiv unter einem „Algorithmus“ verstehen. Das wird auch als die „Church-Turing-These“ bezeichnet und rechtfertigt, dass wir uns in diesem Buch mit einem informellen Algorithmus-Begriff begnügen.

Aufgaben

2.1.1. Aufgabe. Formuliere Algorithmen für:

(a) Die Multiplikation zweier natürlicher Zahlen.

Notationsverzeichnis

Stichwortverzeichnis

Literaturverzeichnis

[AB] Agrawal, M. und Biswas, S.: Primality and Identity Testing via Chinese Remaindering. *Journal of the ACM* **50** (2003), No. 4, 429 – 443.

[AKS] Agrawal, M., Kayal, N. und Saxena, K.: PRIMES is in P. *Annals of Math.* **160** (2004), No. 2, 781 – 793.

[ANF] Alten, H.-W., Naini, A.D., Folkerts, M., Schlosser, H., Schlote, K.-H- und Wußing, H.: *4000 Jahre Algebra.* Springer, 2003.

[AÖ] Ağargün, A., Göksel und Özkan, E. Mehmet: A historical survey of the fundamental theorem of arithmetic. *Historia Math.* **28** (2001), No. 3, 207 – 214.

[Ba] Barth, A.P.: *Algorithmik für Einsteiger.* Vieweg, 2003.

[Bo] Bornemann, F.: Ein Durchbruch für „Jedermann". *DMV-Mitteilungen* 4/2002, 14–21.

[Br] Brands, G.: *Verschlüsselungsalgorithmen.* Vieweg+Teubner, 2002.

[Chen] Chen, J.R.: On the representation of a larger even integer as the sum of a prime and the product of at most two primes. *Sci. Sinica* **16** (1973), 157176.

[Con] Conrey, J.B.: The Riemann Hypothesis. *Notices of the AMS*, March 2003, 341–353.

[CM] Coron, J.-S. und May, A.: Deterministic Polynomial-Time Equivalence of Computing the RSA Secret Key and Factoring. *J. Cryptology* **20** (2007), 39–50.

[CP] Crandall, R. und Pomerance, C.: *Prime Numbers: A computational perspective.* Springer, 2005.

[De] Derbyshire, J.: *Prime Obsession: Bernhard Riemann and the Greatest Unsolved Problem in Mathematics.* Penguin, 2004.

[Dst] Diestel, R.: *Graphentheorie.* Springer, 1996.

[Dtz] Dietzfelbinger, M.: *Primality Testing in Polynomial Time: from randomized algorithms to „PRIMES is in P".* Springer, 2004.

[EFT] Ebbinghaus, H.-D., Flum, J. und Thomas, W.: *Einführung in die Mathematische Logik.* Spektrum, 2007.

[E] Ebbinghaus et al.: *Zahlen.* Springer, 1992.

[Fo1] Forster, O.: *Algorithmische Zahlentheorie.* Vieweg+Teubner, 1996.

[Fo2] Forster, O.: *Analysis 1.* Vieweg+Teubner, 2008.

[Fr] Franzén, T.: *Gödel's Theorem: An Incomplete Guide to its Use and Abuse.* Peters, 2005.

[Fü] Fürer, M.: *Faster integer multiplication.* Proceedings of the 39th Annual ACM Symposium on Theory of Computing (2007), 57–66.

[G] Granville, A.: It is easy to determine whether a given integer is prime. *Bull. Amer. Math. Soc.* **42** (2005), No. 1, 3–38.

[GT] Green, B. and Tao, T.: The primes contain arbitrarily long arithmetic progressions. *Annals of Math.* (2) **167** (2008), No. 2, 481–547.

[Hal] Halmost, P.R.: *Naive Mengenlehre.* Vandenhoeck & Ruprecht, 5. Auflage, 1994.

[Har] Hardy, G.H.: *A mathematician's apology.* Cambridge University Press, 1992.

[HW] Hardy, G.H. und Wright, E. M.: *An Introduction to the Theory of Numbers.* Oxford University Press, 2008.

[Ho] Hofstadter, D.: *Gödel, Escher, Bach: ein Endloses Geflochtenes Band.* DTV, 1992.

[HMU] Hopcroft, J.E., Motwani, R. und Ullman, J.D.: *Introduction to Automata Theory, Languages, and Computation.* Pearson/Addison-Wesley, 2007.

[J] Jameson, G.J.O.: *The Prime Number Theorem.* Cambridge University Press, 2008.

[Ke] Kelly, T.: The myth of the Skytale. *CRYPTOLOGICA*, Vol. XXII No. 3 (1998).

[Kra] Kramer, J.: *Zahlen für Einsteiger: Elemente der Algebra und Aufbau der Zahlbereiche.* Vieweg+Teubner, 2007.

[Kre] Krengel, U.: *Einführung in die Wahrscheinlichkeitstheorie und Statistik.* Vieweg+Teubner, 2005.

[KS] Kurzweil, H. und Stellmacher, B.: *Theorie der endlichen Gruppen.* Springer, 1998.

[LaPe] Laubenbacher, R. und Pengelley, D.: "Voici ce que j'ai trouvé": Sophie Germain's grand plan to solve Fermat's Last Theorem. Preprint, 2009, wird erscheinen in Historia Mathematica.

`http://www.math.nmsu.edu/~davidp/germain02.pdf`

[LP] Lenstra, H.W. Jr. und Pomerance, C.: Primality testing with Gaussian periods. Preprint, 2009.

`http://math.dartmouth.edu/~carlp/PDF/complexity12.pdf`

[LPa] Lewis, H.R. und Papadimitriou, C.H.: *Elements of the theory of computation.* Prentice Hall International, 1998.

[LiN] Lidl, R. und Niederreiter, H.: *Finite Fields.* Addison-Wesley, 1983.

[Lo] Lorenz, F.: *Einführung in die Algebra I.* Spektrum, 1999.

[ME] Murty, M.R. und Esmonde, J.: *Problems in Algebraic Number Theory.* Springer, 2004.

[N] Nair, M.: On Chebyshev-type inequalities for primes. *Amer. Math. Monthly* **89**, No. 2, (1982), 126–129.

[NZM] Niven, I., Zuckerman, H.S. und Montgomery, H.L.: *An Introduction to the Theory of Numbers.* John Wiley & Sons, 1991.

[P] Papadimitriou, C.H.: *Computational complexity.* Addison-Wesley, 1995.

[RU] Remmert, R. und Ullrich, P.: *Elementare Zahlentheorie.* Birkhäuser, 1995.

[RSA] Rivest, R., Shamir, A. und Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Comm. of the ACM* **21** (1978), No. 2, 120–126.

[Rob] Robinson, S.: Still Guarding Secrets after Years of Attacks, RSA Earns Accolades for its Founders. *SIAM News* **36**, No. 5 (2003).

[Ross] Ross, P.M.: On Chen's theorem that each large even number has the form p_1+p_2 or $p_1+p_2p_3$. *J. London Math. Soc.* **10** (1975), 500–506.

[S] Singh, S.: *Geheime Botschaften.* Hanser Verlag, 1999.

[St] Stroth, G.: *Algebra. Einführung in die Galoistheorie.* De Gruyter, 1998.

[TZ] Tao, T. und Ziegler, T.: The primes contain arbitrarily long polynomial progressions. *Acta Math.* **201**, No. 2 (2008), 213–305.

[vK] von Koch, H.: Ueber die Riemann'sche Primzahlfunction. *Math. Annalen* **55** Nr. 3 (1901), 441–464.

[Z] Zagier, D.: Newman's short proof of the prime number theorem. *American Math. Monthly* **104** (1997), 705-708.