

## IN DIESEM KAPITEL

werden wir über Zahlen, ihren Aufbau und ihre elementaren Gesetzmäßigkeiten sprechen.

betonen wir den konstruktiven Aspekt der Zahlen

und nutzen sie bei der Beschreibung elementarer logischer Gesetze.

# Kapitel 1

# Zahlen und ihre Logik

**A**lles ist Zahl« war der Glaubenssatz der alten Pythagoreer, einer altgriechischen philosophischen Schule. Der aus Schulzeiten bekannte Pythagoras, untrennbar mit dem »Satz es Pythagoras« verbunden, hat ihr seinen Namen gegeben. Für heutige Informatiker ist dieser Satz »Alles ist Zahl« wieder bemerkenswert aktuell geworden. Lässt sich für sie doch jeder Text, jedes Bild und jeder Ton als Folge von Nullen und Einsen auffassen und damit als binäre Zahlen. »Zahlen inside«, so könnte man dies ausdrücken. Sie sind die Wurzel der Mathematik, auch wenn sich Mathematiker in späteren Zeiten (scheinbar) von den Zahlen emanzipierten und sich in immer dünnerer Luft mit immer abstrakteren Objekten beschäftigten. Und so bleiben Zahlen in Schule und Hochschule der natürliche Einstieg in die Mathematik.

## Was es über die Vielfalt der Zahlen zu sagen gibt

Man kann Zahlen als etwas »vom Himmel gefallenes« ansehen: »Die natürlichen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk.« formulierte einst der exzellente Zahlentheoretiker Leopold Kronecker. Die natürlichen Zahlen sind uns danach als Ordnungszahlen, also »erster«, »zweiter«, »dritter« usw. auf natürliche Weise vorgegeben. Ein solcher Standpunkt hat den unschätzbareren Vorteil, einen philosophischen Abschluss bezüglich weiterer Grundlegungen anzuerkennen und damit einigen Diskussionen über die Zahlen und ihr »Wesen« aus dem Wege zu gehen – zunächst. Denn im dritten Kapitel werden wir – Kronecker wird es verzeihen – doch noch etwas tiefer »graben«.

In jedem Fall steht es einem Informatiker gut zu Gesicht, ausgehend von obiger Grundlegung den weiteren Zahlenraum konstruktiv zu erschließen, was wiederum ganz im Sinne

von Kronecker wäre. Dabei werden schon die natürlichen Zahlen selbst über ein konstruktives Verfahren charakterisiert, den des Zählens. Können wir zählen, dann ist damit ein erster wichtiger Einstieg in das Verständnis der natürlichen Zahlen gegeben. Wir können dann nämlich, wenn auch zunächst etwas mühsam, mit den Fingern einfache Additions- und Subtraktionsaufgaben durchführen. Es lohnt sich deshalb, dem Vorgang des Zählens im nächsten Abschnitt mehr Raum zu widmen.

Insbesondere die Subtraktion führt uns an Grenzen, an denen wir mit den natürlichen Zahlen nicht mehr weiterkommen. Diese Grenzen werden deshalb mittels der Einführung negativer Zahlen erweitert. Historisch geschah dies durch einen besonderen Druck im Wirtschaftsleben. Wenn man Ausgaben tätigen wollte, die das eigene Geldvermögen überstiegen, brauchte man so etwas wie negative Zahlen für die dann notwendige Beschreibung des resultierenden »Nettogeldvermögens«. Auf diesem Weg erhalten wir die *ganzen Zahlen*.

An weitere Grenzen stoßen wir bei der Durchführung der Division, also beim Teilen von Zahlen. Die natürlichen Zahlen erweisen sich ebenso wie die ganzen Zahlen als recht »widerspenstig« gegenüber Divisionen. Manche dieser Zahlen lassen sich (ohne Rest) durcheinander teilen. Die meisten jedoch verweigern sich dem, wie Sie nur allzu gut wissen. Dies führt zur Einführung von Brüchen und den sogenannten *rationalen Zahlen*. Weitere Grenzen wie z.B. beim Wurzelziehen werden wir in diesem Abschnitt noch kennenlernen und sehen, wie weit sich die Zahlenräume beim Überschreiten dieser Grenzen jeweils erweitern lassen.

*Manche Wissenschaften tun sich sehr viel leichter als die Mathematik, wenn es darum geht, die Objekte ihrer jeweiligen Wissenschaft zu beschreiben. So kann der Geologe einfach auf den Boden zeigen, auf dem er steht, auf Steine, Mineralien oder mit ausholender Geste auf ganze Erdformationen. Ein Chemiker kann ins Küchenregal und zum Salzstreuer greifen. Seine Objekte sind Stoffe, chemische Elemente und ihre Verbindungen. Geisteswissenschaftler, z.B. Linguisten, tun sich ein klein wenig schwerer. Ihre Objekte sind Wörter und Sätze. Sie sind obgleich etwas flüchtiger noch hörbar und in einer Schriftsprache jedoch konservierbar. Immerhin haben viele Wörter eine anschauliche Bedeutung. Man kann sich unter ihnen etwas vorstellen. Stellt ein Mathematiker hingegen die Frage: »Was stellen Sie sich unter einer Zahl vor?« kann man ins Grübeln geraten. Vergleichbar dem hl. Augustinus bei der Frage nach dem Wesen der Zeit. Man weiß es eigentlich – solange man nicht danach gefragt wird.*

## Zahlen zählen

Zurück auf »Los«. Es hat – und das darf man als Dozent nur mit äußerster Zurückhaltung bemerken – seine Vorteile, wenn man vergessen hat, wie man etwas gelernt hat. Es ist dann automatisiert, wie das Gleichgewichthalten beim Fahrradfahren, das Binden einer Krawatte und eben auch das Rechnen mit Zahlen. Fragt man also »Was ist eine Zahl?«, kann das Auswirkungen haben, wie die Frage an den Tausendfüßler, wie er es denn schaffe, seine Füße so koordiniert bewegen zu können. Also besser Sie fragen nicht.

Sie sagen »Zahlen? Das ist bei mir doch schon automatisiert.« Da haben Sie recht. Und Sie wollen vermeiden, dass Sie – wie der Tausendfüßler bei seiner Frage – sich hoffnungslos verheddern und ins Stolpern geraten? Dann wäre es als »Ingenieur des Geistes« durchaus

zulässig, so zu antworten, wie es der Ingenieur Edison bei einer vergleichbaren Gelegenheit tat:

»Was Elektrizität ist, gnädige Frau? Elektrizität ist, nutzen Sie sie.«

Und so könnte die spontane Antwort auf die Frage, was Zahlen seien, lauten: »Zahlen sind, nutzen Sie sie.« Wie? Indem wir zunächst das *Zählen* lernen.



So erinnert sich einer der Autoren (der ältere) daran, dass man im Kindergarten die Zahlen der Reihe nach laut aufsagend durch die Straßen zog. Man lernte, dass jede Zahl ihren eindeutig bestimmten Nachfolger hatte. Man kam dabei spielerisch im Zählraum bis in die Hunderte und erfasste die Systematik des dezimalen Zahlensystems.

## Zahlen aufs Papier – und später auf den Rechner

Zählen können, ist eine Sache. Die Zahlen in einfach codierter Form aufs Papier zu bringen, eine andere. Geht es nur darum, von einer Zahl zur nächsten zu kommen, genügt ein Strichcode. Den benutzen wir auch heute noch, z.B. wenn wir bei Wahlen die abgegebenen Ja- und Nein-Stimmen auszählen. Rechnen im eigentlichen Sinne müssen wir dabei nicht. Es genügt der Vergleich.

Geht es nur um die Addition, kommt man mit dem Strichcode noch halbwegs zurecht. Schwieriger wird es mit der Multiplikation. Das merkt man spätestens dann, wenn man römische Zahlen miteinander multiplizieren will. Römische Zahlen spiegeln eben noch den Geist eines Strichcodes wider.

Zum effizienten Multiplizieren benötigen wir ein gegenüber den römischen Zahlen fortgeschrittlicheres Zahlensystem, ein Stellenwertsystem.



Ein solches wurde in den deutschen Landen von Adam Ries (»Adam Riese«) propagiert und erfolgreich verbreitet. Er publizierte auf Deutsch für Handwerker und Kaufleute und bediente damit »seinen Markt«. Eines seiner Lehrbücher wurde 120-mal aufgelegt. Seitdem also rechnen wir »nach Adam Riese«.

War dieses System alternativlos und deshalb ohne Konkurrenz? Nicht ganz. Leibniz schlug (neben anderen) ein vergleichbares, nämlich ein auf der Zahl Zwei basierendes System vor, die Dualzahlen. Die ersten elf Zahlen von Null bis Zehn in binärer Darstellung, werden dann wie folgt notiert:

0,1,10,11,100,101,110,111,1000,1001,1010 ...

Das Bildungsgesetz ist analog dem Bildungsgesetz der Dezimalzahlen. Hat man bei Dezimalzahlen nach der Ziffer 9 einen Überlauf, so hat man bei Dualzahlen den Überlauf schon bei der 1. Dies führt dazu, dass die Dualzahlen im Schnitt mehr als dreimal so lang sind wie dieselben Zahlen als Dezimalzahlen. Das macht Dualzahlen für den täglichen Umgang unhandlich. Für elektronische Rechenmaschinen jedoch erwiesen sie sich als das Mittel der Wahl.

Es kann zu Mehrdeutigkeit kommen, ob es sich bei einer hingeschriebenen Ziffernfolge um eine Dualzahl oder eine Dezimalzahl handelt. So kann 10 ohne Hintergrundinformation, wenn es sich also nicht aus dem Kontext ergibt, als Dualzahl »Zwei« oder als Dezimalzahl »Zehn« gelesen werden. In solchen Fällen notiert man die benutzte Basis, also Zwei oder Zehn, durch eine tief gestellte »2« oder eine tief gestellte »10«. Damit ist  $10_2 = 2_{10}$ ,  $11_2 = 3_{10}$ ,  $100_2 = 4_{10}$  usw.



Stellenwertsysteme sind allgemein das Mittel der Wahl, wenn es um das Multiplizieren beliebig großer Zahlen geht. Um dies effizient durchführen zu können, genügt die Beherrschung des kleinen  $1 \times 1$ . Das einfachste Stellenwertsystem ist das Dualzahlensystem. Hier besteht das kleine  $1 \times 1$  nur aus vier Einträgen. Dass man es im täglichen Gebrauch nicht durchgängig benutzt, hat mehrere Gründe: Tradition und die relativ schnell wachsenden Längen der Dualzahlen im Verhältnis zu den Dezimalzahlen.

### Es darf auch etwas mehr sein – über die natürlichen Zahlen hinaus

Betriebswirte oder auch Finanzdienstleister (Finanzminister bilden da keine Ausnahme) kennen rote und schwarze Zahlen. Den Bedarf für Zahlen mit zweierlei »Farben« kannten schon die alten Babylonier. »Schwarze« Zahlen standen für Guthaben und »rote« Zahlen für Schulden. Hat man nur einen einfarbigen Stift zur Hand, schreibt man die »schwarzen« Zahlen wie gewohnt, die »roten« Zahlen aber mit einem vorangestellten Minuszeichen. Das geht schneller, als nach einem zweiten Stift zu suchen. Die Zahlen mit einem Minuszeichen davor heißen *negative* Zahlen, die anderen sind die *positiven* Zahlen. Die natürlichen Zahlen sind also positive Zahlen. Die Null liegt dazwischen und könnte je nach Geschmack schwarz oder rot dargestellt werden. Finanzminister bevorzugen die »schwarze Null«.

Die natürlichen Zahlen und die daraus (mit einem Minuszeichen) davor gewonnenen negativen Zahlen bilden zusammen die *ganzen* Zahlen. Die durch den Zählprozess gewonnenen natürlichen Zahlen um die negativen Zahlen zu erweitern, hat einen Sinn. Denn die Rechenregeln für natürliche Zahlen können unverändert und nahtlos auf den größeren Bereich der ganzen Zahlen erweitert werden. Der größte Vorteil besteht darin, dass die Subtraktion ohne Hemmungen sozusagen für beliebige ganze Zahlen durchgeführt werden kann. Im Bereich der ganzen Zahlen kann also von einer positiven Zahl eine größere positive Zahl subtrahiert werden. Das Ergebnis ist dann eine ganze Zahl, die negativ ist. Kontoinhaber mit einem hohen Dispositionsräumen schätzen diesen Umstand. Demgegenüber lässt ein Guthabekonto nur das Rechnen mit Ergebnissen im Bereich der natürlichen Zahlen zu.



Im Rechnen mit ganzen Zahlen merken wir uns, dass das Negative einer negativen ganzen Zahl wieder eine positive natürliche Zahl ist. In der Schule lernten wir dazu: Minus mal Minus ergibt Plus. Und so ist auch das Produkt zweier negativer ganzer Zahlen eine positive natürliche Zahl.

## Ganzzahlige Brüche – ein zweiter Nachschlag

Beim Teilen von Zahlen, auch das haben Sie in der Grundschule gelernt, bleibt häufig ein Rest. Da Sie lieber ohne Rest teilen wollten, unterstellen wir das einmal, empfanden Sie das als misslich. Mit dem, was als Zahlobjekte in Gestalt der ganzen Zahlen zur Verfügung stand, ging es aber nicht anders. Das Teilen ging in den meisten Fällen nicht auf. Damit auch das Teilen generell problemlos wird, müssen wir ein zweites Mal unseren Zahlenbereich erweitern. Die dabei zusätzlich ins Spiel kommenden Zahlen sind die ganzzahligen Brüche.



Ganzzahlige Brüche können in verschiedenen Formaten, also durch verschiedene »Datenstrukturen« dargestellt werden. Zum einen kann jeder ganzzahlige Bruch in der Form  $\frac{a}{b}$  geschrieben werden. Alternativ kann man jede solche Zahl als periodischen Dezimalbruch darstellen. So notiert man ein Drittel als ganzzahligen Bruch in der Form  $\frac{1}{3}$  und als Dezimalbruch in der bekannten Form  $0.\overline{3}$ . In beiden Fällen ist eine jeweils exakte Speicherung der Brüche auf dem Rechner mit endlichen Speicherressourcen prinzipiell möglich. Im Falle des ganzzahligen Bruches genügen ganzzahliger Zähler und Nenner zum Abspeichern. Im Falle eines periodischen Dezimalbruches kann etwas mehr Aufwand erforderlich sein. Hier wäre zu speichern: die ganze Zahl vor dem Komma, die Ziffernfolge bis zum Beginn der Periode und die Ziffernfolge der Periode selbst. In jedem Fall sind ganzzahlige Brüche auf einem Digitalrechner grundsätzlich exakt darstellbar, denn der Platzbedarf ist immer nur endlich. Dass man dennoch nicht (immer) so verfährt und Rundungsfehler zulässt, hat praktische Gründe. Hier wie auch anderswo geht Geschwindigkeit vor perfekter Genauigkeit.

Ganzzahlige Brüche heißen *rationale Zahlen*. Rationale Zahlen werden aufgefasst als natürliche Erweiterung der ganzen Zahlen, so wie die ganzen Zahlen als natürliche Erweiterung der natürlichen Zahlen dienen. Damit können ganze Zahlen automatisch auf unterschiedliche Weisen notiert werden. Die Zahl »Eins« kann geschrieben werden

- ✓ in der ursprünglichen Form einer ganzen Zahl als »1«,
- ✓ als unmittelbar ganzzahliger Bruch in der Form » $\frac{1}{1}$ «,
- ✓ als periodischer Dezimalbruch »1,000 ...« (kurz: » $1,\overline{0}$ «).

Besteht die Periode nur aus Nullen, spart man sich gerne deren Darstellung. Man schreibt also auch 1,0 anstelle von  $1,\overline{0}$ .

Rationale Zahlen in der Form periodischer Dezimalbrüche haben den verführerischen Vorteil, dass eine eigene »Bruchrechnung« nicht wirklich gelernt werden müsste. Setzt man die Komma an die richtigen Stellen, kommt man mit den Rechenregeln der Addition, Subtraktion, Multiplikation und Division für ganze Zahlen prinzipiell durch.



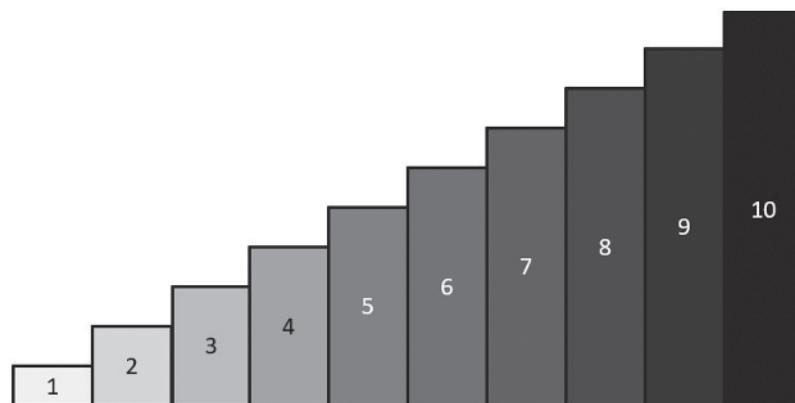
Mit dem Versuch, der eigentlichen Bruchrechnung aus dem Wege zu gehen, indem man alle Brüche in Dezimalbrüche umformt, tut man sich langfristig keinen Gefallen. Der Preis hierfür kann in einer schnell wachsenden Rechenkomplexität bestehen, auch wenn die Rechenregeln selbst einfacher sind. Um dies nachzusempfinden, berechnen Sie bitte den Bruch  $\left(\frac{1}{3} + \frac{1}{4}\right) \cdot \frac{1}{2}$  einmal mit den Regeln der Bruchrechnung und einmal in der Form  $(0,3 + 0,25) \cdot 0,5$ .

Die Regeln der Bruchrechnung sind entgegen weit verbreiteter Ansicht aber nicht wirklich kompliziert. Sollten Sie sie nicht mehr beherrschen, können Sie sich später noch einmal damit auseinandersetzen.



Schon die alten Griechen rechneten mit Brüchen, sprachen von ihnen aber nicht als Zahlen, sondern nannten sie »Proportionen«. Die Griechen waren einfach etwas restriktiver, wem sie das volle »Bürgerrecht« im Reich der Zahlen gaben. Proportionen wurden auf natürliche Weise durch Längen dargestellt. Eine Informatiker würde hierzu bemerken, dass sie eine *analoge* im Gegensatz zu unseren digitalen Darstellungen benutzten.

Auch unsere heutigen Grundschulkinder arbeiten noch mit analogen Zahldarstellungen. Zahlen werden dabei durch Stäbchen verschiedener Längen dargestellt (siehe 1.1). Ein Stäbchen mit genormter Länge 1 steht dann für die Zahl »Eins«. Ein Stäbchen mit genormter Länge 5 steht für die Zahl »Fünf« usw. Legt man also 5 Stäbchen der Länge 1 aneinander, dann sind sie insgesamt so lang wie ein Stäbchen der Länge 5. Die hierbei genutzte Haptik unterstützt das Zahlenverständnis auf subtile und dennoch effektive Weise. Zwei Stäbchen, zu denen es ein kürzeres Stäbchen gibt, so dass man die Längen der ersten Stäbchen jeweils lückenlos mit Kopien des kürzeren Stäbchens auffüllen kann, nennt man *kommensurabel*. Ein Stäbchen der Länge 2 ist also kommensurabel zu einem Stäbchen der Länge 5. Denn die Länge des Stäbchens mit der Länge 2 kann ebenso wie die Länge des Stäbchens mit der Länge 5 gleichermaßen durch Stäbchen der Länge 1 perfekt aufgefüllt werden. Das Verhältnis der beiden Längen 2 und 5 ist also  $2 : 5$ , nämlich 2 Stäbchen der Länge 1 zu 5 Stäbchen der Länge 1. Das Verhältnis der beiden ist eine rationale Zahl, nämlich  $\frac{2}{5}$  oder 0,4.



**Abbildung 1.1:** Analoge Darstellung ganzer Zahlen durch Stäbchen mit kalibrierten Längen



Für die Griechen galt es als »gesetzt«, dass es nur kommensurable Proportionen geben könne. Alles andere wäre für die griechischen Denker »irrational« gewesen. Beim gleichzeitigen Betrachten der Seiten und der Diagonalen eines regulären Fünfecks erlebten Sie allerdings eine böse Überraschung. Deren Verhältnis war *irrational*! Dies war sozusagen die Geburtsstunde der Irrationalzahlen und das Todesurteil für denjenigen, der diese Erkenntnis ausplauderte. Es war schon immer ein Problem, sich mit etablierten Glaubenssätzen anzulegen. Wir haben es heute etwas einfacher. (Nicht mit dem Anlegen gegenüber Autoritäten, sondern mit dem Finden irrationaler Zahlen.) Wir nehmen nicht mehr das regelmäßige Fünfeck, sondern das regelmäßige Viereck, sprich das Quadrat. Auch dort ist die Seitenlänge in einem irrationalen Verhältnis zur Diagonale. Gibt man der Seitenlänge den Wert 1, dann hat die Diagonale den Wert  $\sqrt{2}$ .

## Die Welt der rationalen Zahlen ist für Informatiker genug – Mathematiker sind weniger bescheiden

Den Informatikern geht es ein wenig wie den alten Griechen. Sie haben mit den irrationalen Zahlen ein Problem. Es liegt darin, dass in der digitalen Darstellung von Zahlen technisch nur rationale Zahlen erfasst werden können. Warum? Digitalisierung bedeutet, wie man es z.B. bei der digitalen Darstellung von Bildern sehen kann, stets eine Rasterung. Digital dargestellte Objekte können deshalb nicht wirklich stufenlos verkleinert oder vergrößert werden. Das dafür notwendige Kontinuum ist digital nicht zu erfassen. Dem Kontinuum nähern können wir uns mit Objekten, die das Kontinuum mehr oder weniger direkt widerspiegeln. So können wir mit einem Faden kontinuierlich sich ändernde Längen abgreifen. Theoretisch, denn auf atomarer Ebene war es das dann doch mit der kontinuierlichen Messbarkeit.



Das Konzept solcher analogen Repräsentationen, also Zahlen als Längen, oder elektrischen Spannungen oder Stromstärken hat dennoch in Spezialanwendungen seine Marktnischen. Es gibt dort »Integrierer«, »Differenzierer« und anderes mehr, die als Hauptkomponente das im obigen Sinne analog operierende Bauteil des elektronischen Operationsverstärkers benutzen. Die Älteren haben mechanische Analogrechner noch in Form des Rechenschiebers kennengelernt. Zahlen sind dort auf eine besondere Weise, nämlich durch logarithmisch wachsende Längen codiert, die den Rechenschieber bei Multiplikationsaufgaben zum Mittel der Wahl werden lassen – sollte der Strom für die Digitalrechner einmal ausfallen. Rechenschieber gehörten zum Ingenieur wie heute das Smartphone zum Studenten. Und es ist wenig bekannt, dass die Mondfahrer der Apollo-Missionen selbstverständlich mit Rechenschiebern als absolut ausfallsicherem Werkzeug ausgerüstet waren.

Und so können Schulkinder analog rechnen, indem sie mehrere Stäbchen mit normierten Längen zusammengesetzt auf Längengleichheit hin vergleichen. Sie erhalten auf diese Weise wie im Beispiel der Abbildung 1.2 dargestellt, dass

$$10 = 1 + 9 = 2 + 8 = 1 + 1 + 8 = 1 + 2 + 7 = 2 + 2 + 6 = 1 + 2 + 3 + 4 = 2 + 1 + 3 + 2 + 2.$$

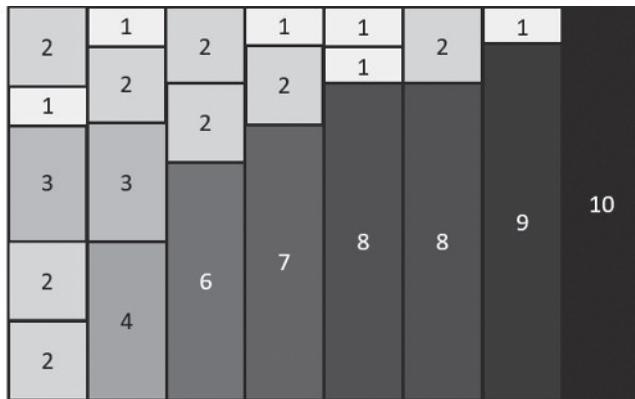


Abbildung 1.2: Addition mit analogen Größen. Zahlen dargestellt als Längen

Die Unschärifen unserer Welt, letztlich die Unschärifen auf der Ebene der Mikrophysik, verhindern aber eine perfekte Darstellungen irrationaler Zahlen wie  $\sqrt{5}$  oder  $\sqrt{2}$ . Die Genauigkeit bei analoger Darstellung von Zahlen ist auf wenige Dezimale hinter dem Komma beschränkt. Man brauchte eine neue, eine platonische Welt ohne jede Unschärfe, um irrationale Zahlen exakt auf einen Analogrechner zu bringen. Eine solche platonische Welt ist aber »Nicht von dieser Welt«. Ob Informatiker also eine digitale Zahldarstellung benutzen, was der Normalfall ist, oder eine analoge, für Irrationalzahlen haben wir nur mehr oder weniger gute Näherungen. Ingenieurtechnisch hat man hier keine andere Chance.

Natürlich schrecken Mathematiker vor den Irrationalzahlen dennoch nicht zurück. Und es gibt, wie wir gleich sehen werden, auch für Informatiker gute Gründe, das Rechnen mit Irrationalzahlen nicht ganz außen vor zu lassen.



Die rationalen Zahlen und die irrationalen Zahlen werden zum Zahlkörper der *reellen Zahlen* zusammengefasst. Auch wenn das Rechnen mit Zahlen in der Regel digital erfolgt, benutzen wir zur Visualisierung mathematischer Sachverhalte automatisch das geometrische Bild des Zahlenstrahls. Das ist eine Gerade, auf der die 0 und die 1 markiert sind, und bei der alle weiteren Punkte auf dieser Geraden genau eine reelle Zahl bedeuten. Jeder Punkt der Geraden steht für eine reelle Zahl. Umgekehrt gehört zu jeder reellen Zahl genau ein Punkt auf der Geraden. Die Gerade steht für ein eindimensionales Kontinuum. Streng genommen sind wir damit wieder bei einer analogen, einer nicht-digitalen Darstellung der Zahlen gelandet, aber bei einer sehr praktischen. Dieses geometrische Bild des Zahlenstrahls suggeriert, dass wir bei unserem Zahlenaufbau hier einen Abschluss erreicht haben sollten. Auf der Zahengeraden ist einfach kein Platz mehr für neue zusätzliche Zahlen. Wir werden im nächsten Abschnitt sehen, dass es dennoch zusätzliche Erweiterungsmöglichkeit gibt.

Warum also sollten sich auch Informatiker mit den Rechenregeln für Irrationalzahlen beschäftigen? Die Gründe hierfür liegen zum Teil im Kerngebiet der Informatik, der Entwicklung effizienter Algorithmen. Zum Teil liegen sie auch in Bereichen, wie etwa

dem Quantencomputing. Dort liegen irrationale Brüche der Form  $\frac{1}{\sqrt{2}}$  einfach in der Natur der Sache. Man hat sich der sinnvollen Handhabung solcher Brüche zu stellen. Selbstverständlich kann (und muss) man beim praktischen Rechnen hiermit diese Brüche durch rationale Brüche näherungsweise erfassen. Es spielt rechnerisch aber eine große Rolle, wie man dies tut.



### 1. Ansatz

Wollen wir  $\frac{1}{\sqrt{2}}$  näherungsweise ausrechnen, können wir das tun, indem wir zunächst für  $\sqrt{2}$  selbst einen Näherungswert berechnen. Als eine erste Näherung könnte man 1,41 benutzen, um dann 1 durch 1,41 mit den bekannten Methoden der Dezimalrechnung zu dividieren. Man erhält hierbei einen Wert näherungsweise gleich 0,704. Dieses Ergebnis ist ab der dritten Dezimale hinter dem Komma nicht mehr exakt. Benötigt man eine bessere Näherung, könnte man die Rechnung mit einem genaueren Wert für  $\sqrt{2}$  wiederholen, z.B. mit 1,4142. Man dividiert wiederum 1 durch diesen genaueren Wert und erhält jetzt als besseren Näherungswert für  $\frac{1}{\sqrt{2}}$  die Zahl 0,70711. Diese Zahl ist erst ab der fünften Stelle hinter dem Komma falsch. Offensichtlich ist die Division von 1 durch 1,4142 mit erheblich mehr Aufwand verbunden als die Division durch 1,41. Schlimmer noch, das Ergebnis der ersten Division ist von keinerlei Nutzen für die zweite Division. Man fängt wieder ganz vorne an.

### 2. Ansatz

Beherrscht man den Umgang mit Irrationalzahlen (und auch die Bruchrechnung), kann man sich das Leben etwas leichter machen. Im ersten Schritt liefern uns die Regeln der Bruchrechnung

$$\frac{1}{\sqrt{2}} = \frac{1 \cdot \sqrt{2}}{\sqrt{2} \cdot \sqrt{2}} = \frac{\sqrt{2}}{2}.$$

Beginnt man die Rechnung mit dem ersten Näherungswert für  $\sqrt{2}$ , nämlich 1,41, so genügt einfaches Kopfrechnen, um als Ergebnis 0,705 zu erhalten. Benötigten wir eine höhere Genauigkeit, könnten wir jetzt 1,4142 durch 2 dividieren. Dabei können wir nahtlos auf die erste Näherung aufsetzen. Das erste Ergebnis dient also als Zwischenergebnis, das übernommen werden kann, um bei der weiteren Division die hinzugekommenen Dezimalen 4 und 2 von 1,4142 mit einzukalkulieren. Wir erhielten somit als Zweite Näherung 0,70710. Der Aufwand für die erste Näherung ist nicht verloren, und das Ergebnis hat sogar eine höhere Genauigkeit als die erste Rechnung via  $1 : 1,4142$ .

## Komplexe Zahlen erweitern den Zahlenraum ein weiteres Mal

Die mathematische Erweiterung des Zahlenbereichs bis hin zu den Irrationalzahlen kennzeichnet aber noch nicht die Grenzen des Zahlenbereiches. Der Zahlenstrahl suggerierte

nur einen Abschluss. Dreimal jedoch geht es noch weiter. Dabei verlieren wir bei jeder Erweiterung bedauerlicherweise jeweils eine der im Reich der reellen Zahlen ausnahmslos gültigen Eigenschaften und Rechenregeln. So können wir die komplexen Zahlen nicht mehr im Sinne eines »kleiner« und »größer« anordnen. Das hängt damit zusammen, dass die komplexen Zahlen gleichsam in eine neue Dimension »ausbüchsen«.

Der entscheidende Punkt im Übergang zu den komplexen Zahlen ist die Einführung einer einzigen neuen Zahl, die man »imaginär« nennt und mit  $i$  gekennzeichnet wird. Diese Zahl spannt eine ganz neue Dimension auf. Das heißt, sie liegt nicht mehr auf dem (eindimensionalen) Zahlenstrahl, sondern in der Ebene oberhalb des Zahlenstrahls. Man nennt diese Zahl  $i$  und gibt ihr die Eigenschaft, dass

$$i^2 = -1 \quad \text{oder} \quad i = \sqrt{-1}.$$

Zahlen mit einer solchen Eigenschaft können nicht mehr auf unserem Zahlenstrahl liegen, da alle Quadrate der Zahlen auf dem Zahlenstrahl ausnahmslos nicht negativ sind. Mit dieser Erweiterung muss der neue Zahlenbereich als Ebene dargestellt werden. Jede komplexe Zahl  $z$  kann nämlich in der Standardform  $z = x + iy$  geschrieben werden und findet ihren natürlichen Ort in der Ebene als Punkt mit den Koordinaten  $(x, y)$ . Die Punkte mit  $y$ -Koordinate 0 stehen für die reellen Zahlen. Sie liegen auf der  $x$ -Achse, die dem ursprünglichen Zahlenstrahl entspricht.

Man kann hier mit gutem Grund die Sinnfrage stellen, warum man »aus dem Nichts«, so hat es den Anschein, eine neue Zahl einführt. Der Grund liegt in den Lösungsformeln für kubische Gleichungen, die man in der beginnenden Neuzeit fand. Die Zahl  $i$  erwies sich dabei als unverzichtbares Sprungbrett für die Berechnung *reeller* Lösungen.



Beim Versuch, kubische Gleichungen, also Gleichungen allgemein der Form  $x^3 + ax^2 + bx = c$  zu lösen, fand man für die spezielle Gleichungsform

$$x^3 + bx = c$$

die Lösungsformel für ein reelles  $x$  mit

$$x = \sqrt[3]{c/2 + \sqrt{(c/2)^2 - (b/3)^3}} + \sqrt[3]{c/2 - \sqrt{(c/2)^2 - (b/3)^3}}.$$

Für gewisse Werte von  $b$  und  $c$  erhalten wir als Teilausdrücke der Lösungsformel Terme mit einer negativen Wurzel. Wählen wir  $b = -15$  und  $c = 4$ , betrachten wir also die Gleichung  $x^3 - 15x = 4$ , dann liefert die Lösungsformel

$$\begin{aligned} x &= \sqrt[3]{2 + \sqrt{4 + (-125)}} + \sqrt[3]{2 - \sqrt{4 + (-125)}} \\ &= \sqrt[3]{2 + \sqrt{-121}} + \sqrt[3]{2 - \sqrt{-121}}. \end{aligned}$$

Lassen wir alle bekannten Regeln unverändert, gelten insbesondere

$$\sqrt{a \cdot b} = \sqrt{a} \cdot \sqrt{b},$$

dann lässt sich  $\sqrt{-121}$  schreiben als  $\sqrt{(-1) \cdot 121} = \sqrt{-1} \cdot \sqrt{121} = i \cdot 11$ . Um damit die beiden dritten Wurzeln  $\sqrt[3]{2+i \cdot 11}$  und  $\sqrt[3]{2-i \cdot 11}$  zu errechnen, zeigen wir umgekehrt, dass  $(2+i)^3 = 2+11i$ :

$$\begin{aligned}(2+i)^3 &= 8 + 12i + 6i^2 + i^3 \quad (\text{Ausmultiplizieren}) \\ &= 8 + 12i - 6 - i \quad (i^2 = -1, i^3 = -i) \\ &= 2 + 11i.\end{aligned}$$

Auf dem gleichen Weg zeigen wir, dass  $(2-i)^3 = 2-11i$ . Damit haben wir gezeigt, dass

$$\begin{aligned}\sqrt[3]{2 + \sqrt{-121}} &= (2+i) \\ \sqrt[3]{2 - \sqrt{-121}} &= (2-i).\end{aligned}$$

Setzen wir diese Werte in unsere obige Lösungsformel ein, ergibt sich für  $x$ :

$$x = (2+i) + (2-i) = 4.$$

Dieses  $x = 4$  eingesetzt in unsere Gleichung  $x^3 - 15x = 4$  zeigt die Gültigkeit des Ergebnisses.

Die komplexen Zahlen kann man also aufbauend auf der imaginären Einheit  $i$  als Sprungbrett zum Auffinden reeller Lösungen von reellen Gleichungen benutzen. Hat man die reelle Lösung – in unserem Fall war es die 4 – gefunden, konnte man sagen: »*imaginäre Einheit hat ihre Schuldigkeit getan – die imaginäre Einheit kann gehen.*« Aber die komplexen Zahlen hatten damit »einen Fuß in der Tür« und sie funktionierten weiterhin so gut, dass man ihnen über kurz oder lang das volle »Bürgerrecht« im Reich der Zahlen verlieh. Tatsächlich ist das Rechnen mit den komplexen Zahlen so einfach wie das Rechnen mit den bekannten reellen Zahlen. Nichts, rein gar nichts ändert sich an den Rechenregeln, dem Verteilungsgesetz, dem Vertauschungsgesetz und anderem mehr. Und immer dann, wenn ein  $i \cdot i$  auf den Plan tritt, können wir es durch  $i^2 = -1$  ersetzen, ein  $i^3 = (i \cdot i) \cdot i$  durch  $-i$ , ein  $i^4$  durch  $i^2 \cdot i^2 = (-1) \cdot (-1) = 1$  usw.



Informatiker können komplexe Zahlen auf natürliche Weise auffassen als abstrakte Datentypen. Sie sind nur am Ergebnis der Operationen interessiert und nicht an der speziellen Implementierung der komplexen Zahlen. Völlig entmystifiziert wurden die komplexen Zahlen im Modell der Gaußschen Zahlenebene. Hierzu stellt man jede komplexe Zahl der Form  $a + ib$  mit reellen Koeffizienten  $a$  und  $b$  als Punkt in der reellen 2-dimensionalen Ebene mit den Koordinaten  $(a, b)$  dar. Das »Ausbüchsen« der Zahl  $i$  kann man hierbei grafisch sehr schön visualisieren: Die Zahl  $i$  liegt oberhalb des reellen Zahlenstrahls im Abstand 1 über der 0.

Bei den komplexen Zahlen endet normalerweise der Aufbau des Zahlensystems. In Abbildung 1.3 ist er bildhaft dargestellt. Es beginnt im Innersten mit den natürlichen

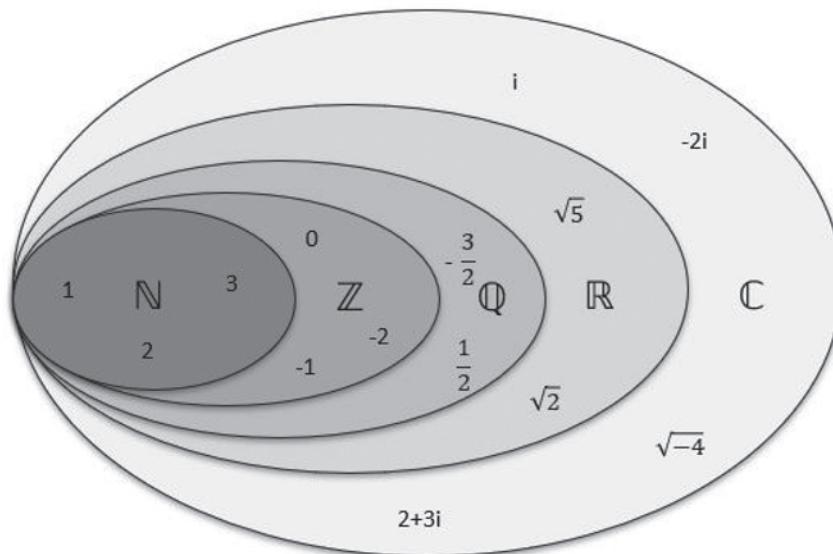


Abbildung 1.3: Aufbau des Zahlensystems von den natürlichen bis zu den komplexen Zahlen

Zahlen, setzt sich nach außen hin fort mit den ganzen Zahlen, den rationalen Zahlen und den reellen Zahlen und wird abgeschlossen durch die komplexen Zahlen. Es gelten die üblichen Bezeichnungen:

- ✓  $\mathbb{N}$  für die natürlichen Zahlen
- ✓  $\mathbb{Z}$  für die ganzen Zahlen
- ✓  $\mathbb{Q}$  für die rationalen Zahlen
- ✓  $\mathbb{R}$  für die reellen Zahlen
- ✓  $\mathbb{C}$  für die komplexen Zahlen

## Blick auf die Gipfel: Hyperkomplexe Zahlen und Optionen

Die zweitletzte Erweiterung unseres Zahlenbereichs (für Experten) betrifft die hyperkomplexen Zahlen, die auch Quaternionen genannt werden. In ihnen steckt, ein wenig verkappt, das Kreuzprodukt von Vektoren, das einem vielleicht einmal im Physikunterricht über den Weg gelaufen ist. Damit kann man Drehbewegungen modellieren, womit die Quaternionen für Informatiker im Bereich der Computergrafik relevant werden. Für Quaternionen gilt nicht mehr das Vertauschungsgesetz. Beim Produkt zweier Quaternionen spielt es nämlich eine Rolle, welche der beiden Faktoren vorne und welche hinten steht. Quaternionen sind, wie es der Name schon andeutet, darstellbar als vierdimensionale Zahlen. Das Wort »darstellbar« ist jetzt nur noch im übertragenen Sinne zu verstehen, denn unmittelbar visualisieren lässt sich der vierdimensionale Raum nicht mehr.

Mit der letzten Erweiterung hin zu den Oktionen oder Cayley-Zahlen ist dann (beweisbar) endgültig Schluss, soweit man verlangt, dass es sich um Divisionsalgebren handelt. Relationsalgebra bedeutet, dass man noch nullteilerfrei dividieren kann. Cayley-Zahlen kann man sich als achtdimensionale Zahlen »vorstellen«. Ihnen fehlt neben dem Vertauschungsgesetz zusätzlich das Gesetz der Assoziativität, das bei den Quaternionen noch gilt. Beim Produkt dreier Oktionen dürfen wir nämlich nicht mehr beliebig klammern. Je nach Klammerung kann das Ergebnis ein Unterschiedliches sein. Wo werden Oktionen angewendet? Man hofft, in der physikalischen Stringtheorie mit ihnen weiterzukommen. Für die Informatik, wie wir sie hier vor Augen haben, sind sie noch irrelevant.

## **Wir wissen nun, über was wir reden, wir wollen jetzt wissen, wie wir darüber reden**

Ein gutes Sprachgefühl ist nicht nur für die Abfassung erfolgreicher Projektanträge wichtig. Es hilft auch, mathematische Aussagen besser zu verstehen, da sie im Kern denselben Gesetzmäßigkeiten gehorchen. Mathematische Aussagen sind Sätze. Sätze sind aufgebaut, wie Sie es im Deutsch- oder Englischunterricht gelernt haben: im einfachsten Fall aus Subjekt, Prädikat und Objekt. Anstelle von »Prädikat« sagte man in der Grundschule oft auch »Satzaussage«. Damit wurde zum Ausdruck gebracht, dass der Kern eines Satzes in der Aussage des Satzes, nämlich im Prädikat liegt. Subjekte und Objekte sind in umgangssprachlichen Sätzen typischerweise Personen oder Gegenstände, die im täglichen Leben eine Rolle spielen. Betrachten wir also den Satz »John liebt Mary.«, dann bildet das Verb »liebt« die »Satzaussage« (das Prädikat) und »John« und »Mary« bilden die Objekte, von denen die Satzaussage handelt.

Im Deutschunterricht lernt man auch den Unterschied zwischen transitiven und intransitiven Verben. Das Verb »arbeiten« z.B. ist intransitiv. Das Verb »bearbeiten« hingegen transitiv. Ein transitivs Verb besitzt nur eine obligatorisch zu besetzende »Bindungsstelle«, nämlich ein Subjekt, um einen vollständigen Satz zu bilden. Ein transitivs Verb besitzt zwei. Die jeweiligen Bindungsstellen werden durch Subjekte und (direkte) Objekte besetzt. Korrekt gebildete Sätze haben damit die Form »John arbeitet« mit dem Wort »John« als Subjekt oder »John bearbeitet eine Aufgabe« mit dem Subjekt »John« und dem direkten Objekt »Aufgabe«. Die Anzahl der notwendigen Bindungsstellen steht für die »Stelligkeit« des mit dem jeweiligen Verb gebildeten Prädikats. So ist das Prädikat »arbeitet« ein einstelliges Prädikat, und das Prädikat »bearbeitet« ist zweistellig.



Die Einbeziehung zusätzlicher indirekter Objekte kann zu drei- und mehrstelligigen Prädikaten führt. Die in der Mathematik gebräuchlichen Prädikate sind, wie man im nächsten Abschnitt sieht, in der Regel ein- und zweistellig.

## **Prädikat – besonders wertvoll**

Mathematische Aussagen sind analog aufgebaut. Subjekt und Objekt können in unserem Fall aber nur aus Zahlen bestehen, denn andere Objekte stehen uns erst einmal nicht zur

Verfügung. Mit welchen Prädikaten arbeiten wir in der Mathematik? Wir kennen sie schon aus der Umgangssprache. Eine wichtige Auswahl:

- ✓ »ist gleich« (zweistelliges Prädikat, symbolisch »=«)
- ✓ »ist kleiner als« (zweistelliges Prädikat, symbolisch »<«)
- ✓ »Teiler von« (zweistelliges Prädikat, symbolisch »|«)
- ✓ »ist Primzahl« (einstelliges Prädikat)
- ✓ »positiv« (einstelliges Prädikat, symbolisch »> 0«)
- ✓ »ist rational« (einstelliges Prädikat)

Die in umgangssprachlichen Worten formulierten mathematischen Prädikate, also »ist gleich«, »ist kleiner« usw. werden in der Mathematik durch besondere Symbole gekennzeichnet. Hiermit können dann auf gewohnte Weise Aussagen über Zahlen gebildet werden. In der folgenden Liste sehen Sie eine kleine Auswahl:

- ✓  $4 + 5 = 9$
- ✓  $4 > 5$
- ✓  $5 > 4$
- ✓  $4|12$
- ✓  $3|10$

Zweierlei wird Ihnen nicht entgangen sein: Manche der Aussagen sind wahr und manche sind falsch. Und bei einer der Aussagen, der ersten, haben wir ein Operatorsymbol, das Pluszeichen, benutzt. Wie kommen wir zu unseren wahren mathematischen Aussagen? Und welche Rolle spielen dabei die Operatoren?

## (Mathematische) Wahrheit

Was ist Wahrheit? Ein Mathematiker könnte hier forschen antworten: Wahrheit ist die Menge aller wahren Sätze. Dies führt natürlich sofort zur Frage: Was ist ein wahrer Satz? Oder wie kommen wir zu unseren wahren Sätzen? Hier greift der Mathematiker zu den Methoden der formalen Logik. Der Informatiker kann sich die Sache wieder etwas leichter machen. Wie? Er kann mathematische Aussagen als Programme auffassen, die als Ergebnistyp einen der zwei Wahrheitswerte »wahr« oder »falsch« besitzen. Als Eingaben besitzen diese Programme wenige (in der Regel ein oder zwei) Eingabeparameter. Die Programme prüfen dann, ob die betreffende Aussage für die eingegebenen Parameter wahr oder falsch sind.

## Mathematische Aussagen als Methodenaufrufe

Die genannten mathematischen Aussagen übersetzt der Informatiker also in Methodenaufrufe. Als Methodennamen wählt der Informatiker einfach den Namen des jeweiligen Prädikats. Aus der Aussage » $4 > 5$ « macht er den Aufruf `istGroesserAls(4,5)` der Methode

```
boolean istGroesser(Zahl x, Zahl y).
```

Die Eingaben für  $x$  und  $y$  dürfen nur Zahlen sein und sind deshalb vom Typ `Zahl`.

Andere mathematische Aussagen werden analog überführt in Methodenaufrufe der Form:

- ✓ `istGroesserAls(4,5)`
- ✓ `istGleich(4,4)`
- ✓ `istTeilerVon(3,7)`

Sie liefern `false` für `istGroesserAls(4,5)`, `true` für `istGleich(4,4)` und wieder `false` für `istTeilerVon(3,7)`.

Nun hat man in den Programmiersprachen für die betrachteten Prädikate ähnliche Symbole wie in der Mathematik. Das spart Schreibaufwand und macht die Programme leichter lesbar. So wird man in Programmiersprachen also keine Aufrufe der Art `istGroesserAls(4,5)` finden. Stattdessen kann man wie von der Mathematik her gewohnt schreiben  $4 > 5$  (was in diesem Fall natürlich `false` zum Ergebnis haben wird).

## Programmierung von zusammengesetzten Prädikaten

Für manche mathematischen Prädikate gibt es in Programmiersprachen keine direkte Entsprechung. Sie müssen aus anderen schon vorhandenen dann zusammengesetzt werden. So kann das Prädikat `ist TeilerVon` im Aufruf `istTeilerVon(3,7)` z.B. programmiert werden in der Form  $7\%3 == 0$ . Allgemein also:

`istTeilerVon(a,b)` wird realisiert durch  $a\%b == 0$ .

Benutzt wird hier der sogenannte Modulo-Operator `%`.  $a\%b$  liefert nämlich den verbleibenden ganzzahligen Rest bei einer ganzzahligen Division von  $a$  durch  $b$ . Ist dieser Rest gleich 0, dies wird mit dem Gleichheitsprädikat `==` überprüft, dann war offensichtlich  $b$  ein Teiler von  $a$ . Im Falle des Aufrufs von  $7\%3 == 0$  dürfen wir also davon ausgehen, dass wir `false` als Ergebnis erhalten. Denn  $7\%3 == 1$  und damit ungleich 0. Der mathematische Satz »3 ist ein Teiler von 7« ist demnach ein falscher Satz. Der entsprechende Programmaufruf teilt uns dies als Ergebnis mit.

## Operatoren – Aus Zahlen werden Zahlen

Mathematische Aussagen benutzen Operatoren. Die ersten Operatoren, die Sie in Ihrer Schulzeit gelernt haben, waren Addition und Multiplikation. Auch Operatoren werden in Programmiersprachen durch Methodenaufrufe realisiert. Im Unterschied zu Methoden eines Prädikats liefern Operator-Methoden keine Wahrheitswerte als Ergebnis, kein `true` oder `false`. Sie liefern als Ergebnis ein Objekt eines passenden Typs, meistens eine Zahl. So ist das Ergebnis der Eingabe von 1 und 2 bei der Addition bekanntermaßen die 3. Die Schreibweise » $1 + 2$ « für die Operation der Addition haben Sie wie auch die üblichen Schreibweisen der anderen Operationen in der Grundschule gelernt. Sie wird in Programmiersprachen eins zu eins unverändert übernommen.

# Logische Operatoren – Schnittstellen zur Logik

In der Umgangssprache benutzen wir logische Partikel wie »und«, »oder«, »nicht« mit denen wir verschiedene (Teil-)Sätze zu einem einzigen Satz zusammenfügen können. Triviale Beispiele solchermaßen zusammengesetzter Sätze sind wohlbekannt. Betrachten wir hierzu den Satz:

»Die Sonne scheint, und wir sind im Schwimmbad.«

Er ist zusammengesetzt aus den zwei (Teil-)Sätzen »Die Sonne scheint« und »Wir sind im Schwimmbad«. Jeder dieser Teilsätze hat unabhängig vom anderen Teilsatz einen eigenen Sinn und kann damit unabhängig vom jeweils anderen wahr oder falsch sein. So können wir uns vorstellen, dass die Sonne scheint, wir aber nicht im Schwimmbad sind, oder dass wir im Schwimmbad sind, obwohl die Sonne nicht scheint, oder dass die Sonne nicht scheint und wir auch nicht im Schwimmbad sind. In der Umgangssprache hat »und« also den Charakter eines Operators, der aus zwei (Teil-)Sätzen einen neuen Satz bildet.

Für die formale Logik ist ein mit »und« zusammengesetzte Satz insofern interessant, als dass seine Wahrheit sich aus den Wahrheitswerten der Teilsätze auf ganz spezifische Weise ergibt.

## Verrechnung von Wahrheitswerten

Eine mit »und« zusammengesetzte Aussage ist falsch, sobald nur eine einzige der in ihr enthaltenden Teilaussagen falsch ist. Dies ist die formale Bedeutung des »und«. Es ist also die Verrechnung von Wahrheitswerten, die dem »und« in der formalen Logik seine spezifische Bedeutung gibt. Damit wird »und« zu einem Operator, der aus zwei eingegebenen Wahrheitswerten wieder einen Wahrheitswert als Ergebnis liefert. Er tut dies synchron zu unserem Sprachempfinden des umgangssprachlichen »und«: Nur aus »wahr« und »wahr« macht unser formales »und« ein »wahr« ansonsten stets ein »falsch«. Vergleichbare Festlegungen gelten das formale »oder« und das formale »nicht«.

## Junktoren

In der formalen Logik erhalten das formale »und«, »oder« und »nicht« besondere Bezeichnungsweisen, die bedauerlicher Weise nicht normiert sind. Es finden sich in unterschiedlichen Lehrbüchern unterschiedliche Notationen hierfür. Hier sollen folgende Zeichen benutzt werden:

- ✓  $\wedge$  für das formale »und«, in Programmiersprachen wird `&` benutzt
- ✓  $\vee$  für das formale »oder«, in Programmiersprachen wird `|` benutzt
- ✓  $\neg$  für das formale »nicht«, in Programmiersprachen wird `!` benutzt.

Sie werden *Junktoren* genannt.

## Wahrheitstabellen

Die Wirkung dieser Junktoren als Operatoren auf Wahrheitswerten beschreibt man typischerweise in sogenannten Wahrheitstabellen. Die Wahrheitstabelle für  $\wedge$  kann nach unseren Vorbetrachtungen auf folgende Weisen dargestellt werden:

$\wedge$	true	false
true	true	false
false	false	false

alternativ:

$x$	$y$	$x \wedge y$
false	false	false
false	true	false
true	false	false
true	true	true

Der Operator  $\wedge$  liefert also dann und nur dann true, wenn beide Eingaben ausnahmslos true sind.

Auf analoge Weise und entsprechend unserem Sprachgefühl wird die Wahrheitstabelle für  $\vee$  wie folgt festgelegt:

$\vee$	true	false
true	true	true
false	true	false

alternativ:

$x$	$y$	$x \vee y$
false	false	false
false	true	true
true	false	true
true	true	true

Der Operator  $\vee$  liefert dann und nur dann false, wenn beide Eingaben ausnahmslos false sind.

Letztendlich erhält  $\neg$  die Wahrheitstabelle:

$x$	$\neg x$
true	false
false	true

**Für den einen ist es duplo, für den anderen  
die längste Praline der Welt – zur Doppelrolle  
der Zahlen in der formalen Logik**

Die Zahlen 0 und 1 können eine subtile Doppelrolle spielen. Die bisher betrachteten Objekte, also Zahlen und Wahrheitswerte, sind in vielen Programmiersprachen von

unterschiedlichem »Typ«. Dies bedeutet, dass man einer Variablen, die vom Typ »Zahl« (z.B. `integer`) ist, keinen Wahrheitswert (vom Typ `boolean`) zuweisen kann. Umgekehrt kann einer Variablen vom Typ »Wahrheitswert« (etwa `boolean`) keine Zahl zugewiesen werden. Sie sind unterschiedlich »getypt«. Zahlen und Wahrheitswerte können damit auch nicht gemeinsam verrechnet werden. Eine Zahl und einen Wahrheitswert kann man damit nicht addieren. Die Operatoren für Zahlen und die für Wahrheitswerte (also die logischen Junktoren) sind nach der reinen Lehre strikt getrennt.

## Vermischen von Zahlen und Wahrheitswerten

Es gibt Programmiersprachen, die zwischen Zahlen und Wahrheitswerten keinen Unterschied machen. Wahrheitswerte *sind* dort Zahlen, `true` wird zu 1 und `false` zu 0. Man kann diesen Weg weitergehen, indem man die logischen Operatoren arithmetisiert. Dies bedeutet, dass man die Wahrheitswerte als Zahlen ernst nimmt und mit ihnen deshalb wie mit Zahlen *rechnet*.

Dies kann geschehen, wenn man die logischen Operationen  $\wedge$ ,  $\vee$ ,  $\neg$  mit den Gleichsetzungen  $1 = \text{true}$  und  $0 = \text{false}$  wie folgt redefiniert ( $x, y$  gleich 0 oder 1):

- ✓  $x \wedge y := x \cdot y$
- ✓  $x \vee y := x + y - x \cdot y$
- ✓  $\neg x := 1 - x$ .

Sie rechnen hiermit leicht nach, dass diese Formeln äquivalente Wahrheitstabellen liefern wie die, die wir oben mit `true` und `false` aufgeführt haben:

$x$	$y$	$x \wedge y$	$x \cdot y$
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

$x$	$y$	$x \vee y$	$x + y - x \cdot y$
0	0	0	0
1	0	1	1
0	1	1	1
1	1	1	1

$x$	$\neg x$	$1 - x$
1	0	0
0	1	1



Am Beispiel der Wahrheitsfunktionen sehen wir, dass wir das Ergebnis eines Funktionsaufrufes einerseits über den Zugriff auf einen Tabelleneintrag erhalten können, andererseits über eine Rechnung. Suchen wir z.B. das Ergebnis von `true`  $\vee$  `false`, können wir dies einerseits aus der Wahrheitstabelle einfach ablesen. Wir können aber auch mit  $1 \vee 0 = 1 + 0 - 1 \cdot 0$ . Offensichtlich sind beide Vorgehensweisen äquivalent, sie liefern dasselbe Ergebnis.

## Funktionen sind Tabellen

Könnte man bei anderen Funktionen analog vorgehen? Theoretisch ja und ansatzweise wurde das auch gemacht. Denken Sie an die Tabellenwerke mit Werten für Sinus, Cosinus, Logarithmen und anderen mehr. (Im Zeitalter der elektronischen Taschenrechner braucht man sie nicht mehr und sie verstauben im Regal – zumindest bis zum nächsten Blackout.) Letztendlich können Tabellen das Rechnen nicht ersetzen, da sie, wenn man es auf die Spitze triebe, quasi unendlich umfangreich sein müssten. Ihnen bleiben gewisse Nischen, wenn man z.B. an die Tabellen des kleinen Einmaleins denkt. Man benutzt sie beim Multiplizieren zur Erhöhung der Schnelligkeit. Man hat einfach auswendig gelernt, dass etwa 6 mal 7 gleich 42 ist. Das geht schneller als das 7-malige Addieren der 6.

Dennoch kann sich ein Mathematiker, besonders wenn er ein Platoniker sein sollte, zumindest vorstellen, ihm stünde eine komplette Tabelle zur Festlegung einer Funktion zur Verfügung. Diese Tabelle wäre völlig äquivalent zum rechnerischen Bildungsgesetz der Funktion. Man könnte mit Fug und Recht sagen: Die Tabelle *ist* die Funktion.

Ein Beispiel einer solchen Funktionstabelle liefert die Nachfolgeroperation der natürlichen Zahlen. Man kann ihr folgende Form geben:

$x$	0	1	2	3	...	100	101	...	$n$	...
<i>Nachfolger(x)</i>	1	2	3	4	...	101	102	...	$n + 1$	...

## Tabellen für Funktionen und für nichtberechenbare Funktionen

Ein weiteres Beispiel findet sich in Abbildung 1.4 zur »Berechnung« der Wurzel reeller Zahlen. Trotz ihres prinzipiellen Nachteils als theoretisch unendlich zu denkende Tabellen sind sie im Prinzip sogar ausdrucksstärker als rechnerische Bildungsgesetze. Mit ihnen können nämlich Funktionen *gedacht* werden, für die es gar kein endliches Rechengesetz gibt. In der theoretischen Informatik findet und betrachtet man solche Funktionen. Für sie gibt es kein endliches Rechengesetz. Sie sind ingenieurtechnisch, also mit finiten Mitteln, nicht *berechenbar*. Auch wenn sich solche Tabellen in unserem Universum also nicht vollständig realisieren lassen, so bieten sie doch ein didaktisches Hilfsmittel: Funktionen können prinzipiell als unendliche Tabellen gedacht werden. Überträgt man die Tabelleninhalte in ein Koordinatensystem, so erhält die Funktion ein bildliches Aussehen, was »mehr ausdrückt als tausend Formeln«. Auf diese Weise erhielten die ersten einfachen Funktionen, mit denen Sie in der Schule konfrontiert wurde, erst ein Gesicht (Geraden, Parabeln, Hyperbeln). Man nennt dies den Graphen einer Funktion. Graphen und Tabellen sind äquivalent. Man kann zwischen beiden ohne Informationsverlust hin- und herwechseln (»toggeln»).

x	0	1	2	3	4	5	6	7	8	9
1	1,0000	0050	0100	0149	0198	0247	0296	0344	0392	0440
1,1	0488	0536	0583	0630	0677	0724	0770	0817	0863	0909
1,2	0954	0100	1045	1091	1136	1180	1225	1269	1314	1358
1,3	1402	1446	1489	1533	1576	1619	1662	1705	1747	1790
1,4	1832	1874	1916	1958	2000	2042	2083	2124	2166	2207
1,5	2247	2288	2329	2369	2410	2450	2490	2530	2570	2610
1,6	2649	2689	2728	2767	2806	2845	2884	2923	2961	3000
1,7	3038	3077	3115	3153	3191	3229	3266	3304	3342	3379
1,8	3416	3454	3491	3528	3565	3601	3638	3675	3711	3748
1,9	3784	3820	3856	3892	3928	3964	4000	4036	4071	4107
2	4142	4177	4213	4248	4283	4318	4353	4387	4422	4457
2,1	4491	4526	4560	4595	4629	4663	4697	4731	4765	4799
2,2	4832	4866	4900	4933	4967	5000	5033	5067	5100	5133
2,3	5166	5199	5232	5264	5297	5330	5362	5395	5427	5460
2,4	5492	5524	5556	5588	5620	5652	5684	5716	5748	5780
2,5	5811	5843	5875	5906	5937	5969	6000	6031	6062	6093
2,6	6125	6155	6186	6217	6248	6279	6310	6340	6371	6401
2,7	6432	6462	6492	6523	6553	6583	6613	6943	6673	6703
2,8	6733	6763	6793	6823	6852	6882	6912	6941	6971	7000
2,9	7029	7059	7088	7117	7146	7176	7205	7234	7263	7292
3	7321	7349	7378	7407	7436	7464	7493	7521	7550	7578
3,1	7607	7635	7664	7692	7720	7748	7776	7804	7833	7861
3,2	7889	7916	7944	7972	8000	8028	8055	8083	8111	8138
3,3	8166	8193	8221	8248	8276	8303	8330	8358	8385	8412
3,4	8439	8466	8493	8520	8547	8574	8601	8628	8655	8682
3,5	8708	8735	8762	8788	8815	8841	8868	8894	8921	8947
3,6	8974	9000	9026	9053	9079	9105	9131	9157	9183	9209
3,7	9235	9261	9287	9313	9339	9365	9391	9416	9442	9468
3,8	9494	9519	9545	9570	9596	9621	9647	9672	9698	9723
3,9	9748	9774	9799	9824	9849	9875	9900	9925	9950	9975
4	2,00000	0025	0050	0075	0100	0125	0149	0174	0199	0224
4,1	0248	0273	0298	0322	0347	0372	0396	0421	0445	0469
4,2	0494	0518	0543	0567	0591	0616	0640	0664	0688	0712
4,3	0736	0761	0785	0809	0833	0857	0881	0905	0928	0952
4,4	0976	1000	1024	1048	1071	1095	1119	1124	1166	1190
4,5	1213	1237	1260	1284	1307	1331	1354	1378	1401	1424
4,6	1448	1471	1494	1517	1541	1564	1587	1610	1633	1656
4,7	1679	1703	1726	1749	1772	1794	1817	1840	1863	1886
4,8	1909	1932	1954	1977	2000	2023	2045	2068	2091	2113
4,9	2136	2159	2181	2204	2226	2249	2271	2293	2316	2338
5	2361	2383	2405	2428	2450	2472	2494	2517	2539	2561
5,1	2583	2605	2627	2650	2672	2694	2716	2738	2760	2782
5,2	2804	2825	2847	2869	2891	2913	2935	2956	2978	3000
5,3	3022	3043	3065	3087	3108	3130	3152	3173	3195	3216
5,4	3238	3259	3281	3302	3324	3345	3367	3388	3409	3431
5,5	3452	3473	3495	3516	3537	3558	3580	3601	3622	3643

Abbildung 1.4: Tabelle für die Wurzelfunktion

## Quantoren in der Logik – Prädikate erhalten durch sie ihre Power

Quantoren in der Prädikatenlogik spielen eine den Schleifen in Programmiersprachen vergleichbare Rollen. Zu Quantoren gehören Variablen so wie zu Schleifen Schleifenvariablen gehören. In der Prädikatenlogik haben wir es mit zwei Quantoren zu tun, dem *Existenzquantor*  $\exists$  und dem *Allquantor*  $\forall$ .

## Der Existenzquantor $\exists$

Die in der Mathematik gebräuchlichen Prädikate zur Bildung mathematischer Aussagen werden wie in der Umgangssprache zur Bildung allgemeiner Aussagen gebraucht. Wenn man das Prädikat »ist Teiler von« betrachtet, bildet man nicht nur die (wahre) Aussage »3 ist Teiler von 12«. Man bildet auch Aussagen der Form »Es gibt eine Zahl, die Teiler von 12 ist«. Der sprachliche Partikel »Es gibt« wird in der formalen Logik wieder durch ein besonderes Zeichen, nämlich  $\exists$ , notiert. Es wird *Quantor* genannt, genauer *Existenzquantor*. Er wird stets kombiniert mit einem VariablenSymbol z.B.  $x$ , das als eine Art »Laufvariable« dient. Die Variable  $x$  durchläuft infrage kommende Objekte, die versuchsweise in das Prädikat als Eingabe eingesetzt werden, um die Wahrheit der dabei konkret entstehenden Aussagen zu testen. Formelhaft schreibt man demnach für »Es gibt eine Zahl, die Teiler von 12 ist.«:

$\exists x x|12.$

## Umsetzung des Existenzquantors in eine Schleife für Programmierer

Getestet wird die Aussage  $\exists x x|12$  im Rahmen einer passenden Methode, die den Namen `esGibtTeilerVon12` erhält. Das Prädikat | (»ist Teiler von«) wird programmiert mittels des Modulooperators %. Der Modulooperator liefert den verbleibenden ganzzahligen Rest bei einer ganzzahligen Division. So ist  $3\%2$  gleich 1. Teilbarkeit wird also dadurch getestet, ob der Rest gleich 0, die Modulooperation also 0 liefert. So wird  $x|12$  programmiert via  $12 \% x == 0$  und die Aussage  $\exists x x|12$  mit:

```
boolean esGibtTeilerVon12(){
    for(int x = 2; x < 12; x++){
        if(12% x == 0) return true;
    }
    return false;
}
```

Wir sehen hier deutlich die Rolle von  $x$  als Laufvariable. Es werden alle Zahlen von 2 bis 11 daraufhin getestet, ob sie Teiler von 12 sind. Das Programm kann mit `true` verlassen werden, sobald ein (einiger) Teiler gefunden wird. Denn darin liegt die Bedeutung von  $\exists$ . Sollte sich herausstellen, dass sich bei jedem Durchlauf kein Teiler findet, würde nach Beendigung der Schleife als Ergebnis `false` herausgegeben. Das wäre z.B. bei der folgenden Aussage der Fall

$\exists x x|13.$

Diese mathematische Aussage würde mit folgender Methode getestet:

```
boolean esGibtTeilerVon13(){
    for(int x = 2; x < 13; x++){
        if(13% x == 0) return true;
    }
    return false;
}
```

Da 13 eine Primzahl ist, lieferte kein  $x$  zwischen 2 und 12 einen Teiler. Die Schleife wird durchlaufen, ohne dass `true` ausgegeben werden kann. Am Ende heißt das Ergebnis folgerichtig `false`.



Machen Sie sich klar, dass der Existenzquantor als effiziente Abkürzung iterierter Junktoren  $\vee$  betrachtet werden kann:

$$\exists x \ x|12 \Leftrightarrow 2|12 \vee 3|12 \vee \dots \vee 10|12 \vee 11|12.$$

Einige Teilaussagen in der iterierten  $\vee$ -Verknüpfung sind falsch wie z.B.  $10|12$ . Für die Wahrheit der Gesamtaussage aber reicht es aus, dass z.B.  $2|12$  wahr ist. Die iterierten Junktoren spiegeln die Bedeutung des Existenzquantors exakt wider.

## Allquantor $\forall$

Das Gegenstück zum Existenzquantor  $\exists$  ist der Allquantor  $\forall$ . Auch er ist gekoppelt mit einer »Laufvariablen«. Eine einfache mit Hilfe des Allquantors formulierbare mathematische Aussage kann lauten:

»Für jede (ganze) Zahl gilt, dass sie gerade oder ungerade ist.«

Mit iteriertem  $\wedge$  würde sie lauten:

»(0 ist gerade oder ungerade)  $\wedge$  (1 ist gerade oder ungerade)  $\wedge$  (2 ist gerade oder ungerade)  $\wedge \dots$ «

Diese Aussage wird formal in zwei Zügen gebildet. Hierbei werden, was bei mathematischen Aussagen sehr häufig geschieht, Allquantoren mit Existenzquantoren gekoppelt. Im ersten Schritt werden die Aussagen » $x$  ist gerade« bzw. » $x$  ist ungerade« formalisiert. Mit Hilfe des Existenzquantors formuliert man dies so: Es gibt eine Zahl  $n$ , so dass  $x = 2 \cdot n$  ( $x$  ist gerade Zahl), oder es gibt eine Zahl  $n$ , so dass  $x = 2 \cdot n + 1$  ( $x$  ist ungerade Zahl). In Formeln

$$\exists n \ x = 2 \cdot n \vee \exists n \ x = 2 \cdot n + 1.$$

Die Aussage »Für jede (ganze) Zahl gilt, dass sie gerade oder ungerade ist.« erhält mit dem Allquantor  $\forall$  damit die Form

$$\forall x \ \exists n \ (x = 2 \cdot n) \vee \exists n \ (x = 2 \cdot n + 1).$$

Die mit dem Existenzquantor gebildeten Teilaussagen werden wie Bausteine in den vom Allquantor gebildeten Rahmen eingesetzt. Kenner der Prädikatenlogik erkennen unschwer, dass in der gerade formulierten Aussage, der zweite Existenzquantor nach vorne gezogen werden kann. Es ist also zulässig, die Aussage etwas kompakter zu schreiben:

$$\forall x \ \exists n \ (x = 2 \cdot n) \vee (x = 2 \cdot n + 1).$$



Kann der Existenzquantor  $\exists$  als effiziente Abkürzungen für viele  $\vee$  aufgefasst werden, so fungiert der Allquantor  $\forall$  in völlig analoger Weise als effiziente Abkürzung für viele  $\wedge$ .



Mathematik oder Logik – wer war zuerst da? Die Frage erinnert an: »Henne oder Ei?«. Hier wie dort gibt es zunächst gute Gründe für jeden dieser beiden Standpunkte. Es hat in der Geschichte der Mathematik und der Logik mehrere ernst zu nehmende Versuche gegeben, einerseits die Mathematik aus der Logik heraus zu begründen und umgekehrt die Logik aus der Mathematik. Bei der Frage »Henne oder Ei?« hat die Evolutionsbiologie das Problem gelöst, indem sie es gewissermaßen zum Verschwinden brachte. In analoger Weise scheint es denkbar, in den Ursprüngen von Logik und Mathematik diese nicht als etwas zu trennendes anzusehen. In dem einen oder anderen Lehrbuch finden sich hierzu sehr pragmatische Standpunkte:

- a. *Wir stellen uns auf den Standpunkt, dass wir bereits etwas Mathematik kennen.*
- b. *Wir beziehen einen Teil unserer Motivation aus der Logik, einen anderen aus dem Bestreben, die Mathematik zu begründen und mathematische Schlussweisen zu analysieren.*

Das klassische Bild der beiden sich gegenseitig malenden Hände von Escher mag diese Sichtweise symbolisieren. Mathematik, repräsentiert durch eine der beiden Hände, und Logik, vertreten durch die andere Hand, bedingen sich gegenseitig.

