Arnold L. Rosenberg, Denis Trystram

Understand Mathematics, Understand Computing

Discrete Mathematics that All Computing Students Should Know

February 13, 2019

Springer

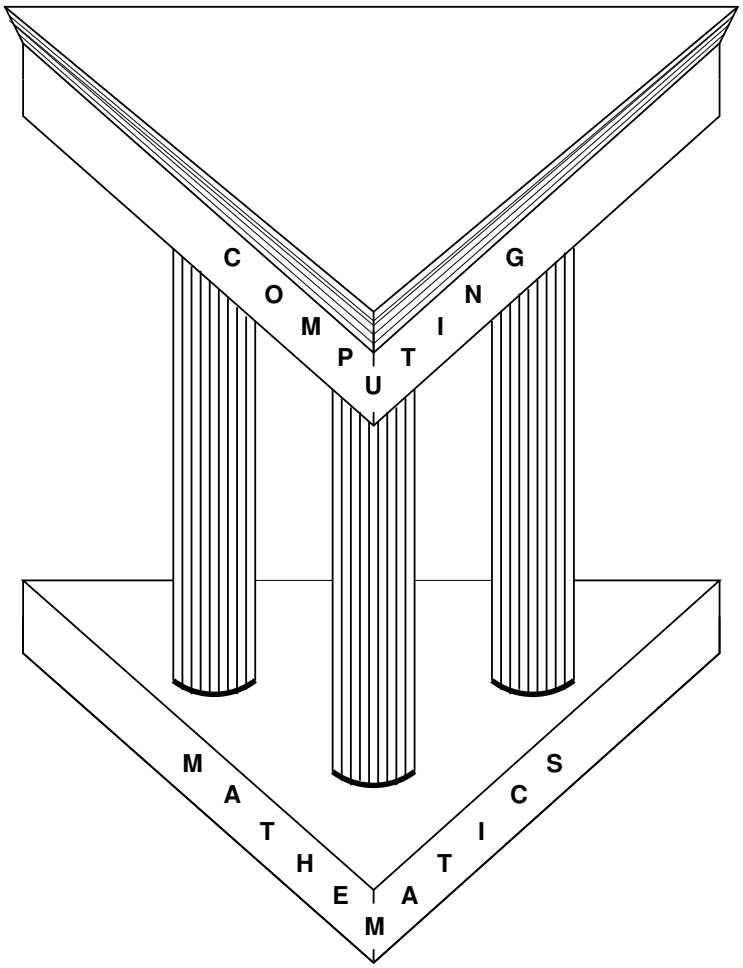
Understand Mathematics, Understand Computing

Discrete Mathematics that All Computing Students Should Know

Arnold L. Rosenberg
Distinguished University Professor Emeritus
University of Massachusetts
Amherst, MA 01003, USA
rsnbrg@cs.umass.edu

Denis Trystram
Distinguished Professor
Univ. Grenoble Alpes
Grenoble, FRANCE
denis.trystram@imag.fr

Mathematics is the foundation on which the edifice of Computing stands



Contents

Chapter 1
INTRODUCTION

Entia non sunt multiplicanda praeter necessitatem.

Occam's Razor

This famous admonition by William of Occam (14th cent.) to strive for simplicity is worth heeding when seeking mathematical models of computational phenomena.

1.1 The "Manifesto" Underlying This Text

As the technology that enabled both the hardware and software systems of modern computers has advanced, the ability to design and utilize such systems "by the seat of one's pants" has commensurately decreased. A vast array of formal aids for the activities of designing, analyzing, utilizing, and verifying computer systemshas developed, and mathematical tools have always been at or near the forefront of such aids.

The fundamental goal of this book—and, therefore, of each of its chapters and sections—is to endow the reader with an operational level of conceptual and methodological understanding of the discrete mathematics that is used to study and understand the activity of computing and the systems that enable that activity. We construe an "operational" level of understanding to be one that enables the reader to "do" the relevant mathematics.

Somewhat surprising to the non-mathematician, a large portion of "doing" mathematics, the often-touted "queen of the sciences", is pattern-matching—albeit of a monumentally sophisticated variety. Mathematicians are trained to understand pieces of reality to a depth that allows them to understand how apparently unrelated concepts A and B can be conceptualized via the same abstract representation, and to analyze (computational, in our bailiwick) advantages to exploiting such representations.

Toward the end of guiding the reader through this forest of abstractions, we categorize our targets in three ways

1. *Fundamental concepts*
   Examples:
   - sets—and their embellishments: tuples, arrays, tables, etc.—as the embodiment of *object*
   - numbers—and their operational manifestations, numerals—as the embodiment of *quantity*
   - graphs—in their many, varied, forms—as the embodiment of *connectivity* and *relationship*
   - algebras—adding operations to sets, numbers, and graphs—as the embodiment of *structured dynamism* and *computing*
2. *Fundamental representations*
   Examples:
   - viewing numbers via the metaphors of: slices of pie, tokens arranged in stylized ways, rectangles of varying dimensions
   - using grouping and/or replication to represent relationships among objects
   - viewing interrelated objects via varying structures: tables, tuples, graphs, geometric drawing
3. *Fundamental tools/techniques*
   Examples:
   - using induction to expand from simple examples to complex ones
   - using integration to approximate summation, and vice versa, to (mentally) "hop" between the discrete and continuous worlds
   - using the conceptual tools of asymptotics to argue qualitatively about quantitative phenomena.

1.2 Overview

In the early days of computing, all aspects of the field were considered the domain of the "techies"—the engineers and scientists and mathematicians who designed the early computers and figured out how to use them to solve a range of problems that is expanding even to this day. Back then, one expected every computer-oriented professional to have a mastery of many mathematical topics. Times—and the field of computing—have changed: classical computing curricula, which have traditional led to a BS degree within a school of science or engineering have been joined by curricula that lead to a BA degree or a degree in IT or in business or .... Within this new world, many aspiring computer professionals arguably need little knowledge of mathematics. However, many argue that the computing field suffers for this lack of mathematics, which has weakened the ability of many practitioners to design and perform experiments, to analyze their results, and to formulate well-reasoned conclusions based on these results. This situation has denied computer science the popu-

lar confidence enjoyed by other empirical disciplines; indeed, many would question the math-deprived practitioners' ability to reason rigorously about basic computational phenomena. Yet others, however, argue that such "scientific" acumen is not needed by practitioners in many of the newer segments of the computing field.

The preceding discussion is important to computing educators because we serve a large population of students, with quite diverse needs and aspirations. We strive to keep this diversity in mind within this essay. As we discuss a range of mathematical topics for possible inclusion as prerequisites for the undergraduate study of computing, we try to indicate why the selected topics are needed. Readers can then evaluate which topics are needed for students enrolled in their computing program.

1.2.1 How to Use This Text

1.2.2 Sample Curricula Based on This Text

1.3 The Elements of Rigorous Reasoning

1.3.1 Basic Reasoning

Distinguishing name from object. A fundamental stumbling block in the road to cogent reasoning arises from the inability to distinguish names from the objects they denote. A prime example within the world of computing resides in the inability to distinguish a function (which can be viewed as an infinite set of argument-value pairs) from a program, which can be viewed as a name for the function. **Note** that the often-used view of a function as a *rule* for assigning values to arguments should be avoided, because it suggests – *erroneously* – that an implementable such rule always exists!

Quantitative reasoning. Students should understand the foundational distinction between "growing without bound" and being innite. Within this theme, they should appreciate situations such as the following. Every integer, and every polynomial with integer coefficients, is finite, but there are infinitely many integers and infinitely many polynomials. Students should be able to verify (cogently but not necessarily via any particular formalism) assertions such as the following.

- Let us be given polynomials $p(x)$ of degree a and $q(x)$ of degree $b > a$, where a, b need not be integers. There must exist a constant $X_{p,q}$ (i.e., a constant that depends on the properties of polynomials p and q) such that for all $x > X_{p,q}$, $p(x) < q(x)$.
  Thus, polynomials having bigger degrees eventually *majorize*—i.e., have larger values than—polynomials having smaller degrees.

- Continuing with polynomial q of degree b: For any real number $c > 1$, there exists a constant $Y_{c;q}$ (i.e., a constant that depends on the properties of polynomial q and constant c) such that for all $x > Y_{c;q}$, $c^x > q(x)$.
  Thus, exponential functions eventually *majorize* polynomials.

1.3.1.1 The Elements of Formal Reasoning

induction
  proof by contradiction

1.3.1.2 The Elements of Empirical Reasoning

Empirical reasoning does not convey the certitude that formal reasoning does.

1.4 Our Approach to Mathematical Preliminaries

"If your only tool is a hammer . . . "

We now review a broad range of mathematical concepts that are central to the study and practice of CS/CE. As we develop these concepts, we shall repeatedly observe instances of the following "self-evident truth" (which is what "axiom" means).

The conceptual axiom. *One's ability to think deeply about a complicated concept is always enhanced by having more than one way to think about the concept.*

Chapter 2
TECHNIQUES FOR "DOING" MATHEMATICS

This is the chapter that "amuses the palate"[1] in preparation for the introductory mathematical repast that we offer the reader. Before we present the hors d'oeuvres that we hope will entice the reader, we "set the table" for our feast.

2.1 Manifesto

This book has a simple, yet fundamental, goal. We want to endow each reader with an *operational* conceptual and methodological understanding of the discrete mathematics that is/can be used to study, and understand, and perform computing. We want each reader to *understand* the elements of computing, rather than just *knowing* them. Thereby, we hope that the interested reader will be able to *develop new concepts* and *invent new techniques* when encountering computational situations that are not explicitly covered in this text. We stress the word *operational:* we want the readers' level of understanding to allow them to "do" mathematics.

> *Lest the reader feel unworthy for the daunting task of "doing" mathematics, we invoke no less an authority than the great German mathematician Leopold Kronecker, to stiffen the spine and embolden the spirit. In [10] (page 477), Kronecker assures us that "God made the integers; all else is the work of man." We may, therefore, be standing on the shoulders of giants*[2] *when we "do" mathematics, but we are not attempting to wrest fire from Olympus!*

Somewhat surprising to the non-mathematician, a large portion of "doing" mathematics, the widely touted "queen of the sciences"[3], is *pattern-matching*—albeit of a monumentally sophisticated variety. Mathematicians are trained to understand pieces of reality to a depth that allows them to understand how apparently unrelated concepts A and B can be conceptualized via the same abstract representation, and to

---

[1] . . . in the sense of the French *amuse-bouche*.

[2] The phrase "standing on the shoulders of giants" has many authors, ranging from Sir Isaac Newton to Bernard of Chartres, and beyond.

[3] See, e.g., Wolfgang Sartorius von Waltershausen, *Gauss zum Gedächtniss* (1856).

analyze (computational, in our bailiwick) advantages to exploiting such representations. It may be useful to the reader to keep the "mathematics-as-pattern-matching" metaphor in mind while reading (from) this book, all the better to enjoy the many instances of pattern-matching that populate its pages.

This chapter is devoted to the practice of mathematics within the world of computing. By means of plentiful examples, we hope to convince the reader of the importance of mathematics in one's quest for mastery over the technology and methodology of computing. By means of extensive explanations—we often prove the same fact many times, from multiple, orthogonal vantage points, we hope to provide the reader with tools for seeing the mathematical aspects of computational settings and phenomena and technology and with guidelines for using those tools effectively.

2.2 Reasoning via Rigorous Proof

Mathematics helps one thrive within the world of computing in two ways: by enabling rigorous argumentation about properties of computational structures and processes and by enabling cogent analyses of such properties. This section is devoted to the first of these topics. We survey, explain, and exemplify a range of proof techniques tha are among most commonly useful within computational settings.

2.2.1 Classical vs. modern proofs and methodologies

Contrary to the all-too-common view of mathematics as arcane strings of symbols that must be manipulated in rigid way, mathematics is a vibrant, evolving system of thinking whose evolution is influenced by the ever-changing objects that are being thought about and by the ever-changing population that are doing the thinking.

Gone forever from the world of the practicing mathematician is the rigidity that characterized the proofs and analyses of the 19th and early-to-mid-20th century.

> *We do not go back earlier than the 19th century because formal notions of rigorous proof are, historically, a relatively recent phenomenon, stemming largely from seminal philosophical developments in the 19th century.*

What has replaced the rigid logical systems and rigid prescribed forms of "antiquity" is a vibrant system of thought that, when convenient,

- represents the number n, as convenient, by, e.g.,
  - a numeral in some positional number system, such as we use in daily discourse,
  - a set (usually imagined) of n balls or . . . or widgets,
  - a unit-width rectangle that is n units high.

- freely uses different modes of argumentation (e.g., numerical and structural induction, contradiction, counting, ...), even mixed and matched throughout a single analysis;
- freely invokes a highly tested computer program to check mind-numbing proliferations of clerically verifiable details.

Regrettably, mathematics education has lagged behind practice, despite the emergence of technical/technological fields such as computer science that cannot advance very far without at least informal extensions and adaptations of the now-archaic formal proof systems.

The current volume is dedicated to trying to overcome people's resistance to mathematical analysis and argumentation via a modernized and humanized—but no less rigorous—methodology for "thinking mathematically", especially within computational frameworks. We attempt to develop proof systems and methods that the reader can comfortably develop facility with.

Our avenue for promoting mathematical *understanding* rather than just rote knowledge abjures any specific formalism. Instead, we develop multiple proofs for the topics of discourse, involving multiple representations of the objects being discussed and multiple modes of argumentation. The reader will be able to see a variety of arguments for the same topic, which will hopefully enhance the likelihood of discovering an approach that is congenial to each reader's individual way of thinking. By practicing proofs and analyses regarding more and more topics, the reader will begin to find it increasingly easy to state informally what ultimately needs to be analyzed rigorously and to intuit how to embark on the path toward such rigor.

2.2.2 What is a "modern" proof?

We are interested in helping the reader learn intuitively compelling, perspicacious techniques for proving interesting mathematical results. But what exactly is a "mathematical proof"—especially within the context of computational systems and artifacts? To answer this question operationally, we follow the lead of the French mathematician René Thom, the Fields medal winning inventor of catastrophe theory. Thom famously defined a proof as an argument that "creates a corpus of evidence for educated readers that leads to their adherence."

> *Est rigoureuse toute démonstration, qui, chez tout lecteur suffisamment instruit et préparé, suscite un état d'évidence qui entraine l'adhésion.*

2.2.2.1 A discussion of "rigor"

2.2.2.2 Sample proofs

A. The average length of a carry in a binary counter

The problem. You add from 1 to n, in increments of 1 using a counter of binary (or, base-2) numerals. Each time you increment the counter, there is a *carry*. These carries have varying lengths; for instance, when $n = 32 = 100000_2$, the carry-lengths range from 0—whenever you increment an even integer—to 5—when you increment $31 = 11111_2$ to achieve $32 = 100000_2$.
Prove that the average carry as you go from 1 *to n hs length* 2*.*

The solution.
Half of the increments add 1 to an even number, i.e., a number whose binary numeral ends in "…0". These increments generate no carry—or, equivalently, a carry of length 0.
One-quarter of the increments, which form half of the remaining increments, execute a carry of length 1, because they add 1 to a numeral that ends in "…01".
One-eighth of the increments, which form half of the remaining increments, execute a carry of length 2, because they add 1 to a numeral that ends in "…011".

Continuing in this way, one can show that the average length of a carry can be expressed in the form

$$\frac{1}{2}\cdot 0 \;+\; \frac{1}{4}\cdot 1 \;+\; \frac{1}{8}\cdot 3 \;+\; \frac{1}{16}\cdot 4 \;+\; \cdots$$

Using techniques that we cover in Chapter 6, one verifies that this infinite series converges with the sum 2. □

B. On meeting new people

The problem. You are attending a cocktail party that is populated by n couples. In order to create a warm atmosphere, the host requests that each attendee shake the hand of every attendee that he or she does not know.
Prove that some two attendees shake the same number of hands.

The solution. This observation follows from the *pigeonhole principle*, which states the following.

If $n+1$ *pigeons occupy n pigeonholes, then some hole contains* 2 *pigeons.*

This principle guarantees that some two attendees shake the same number of hands. To wit, the number of people that each attendee *does not know* belongs to the set $\{0, 1, \ldots, 2n-2\}$, because each person knows him/herself and his/her partner. Because there are $2n$ handshakers (the pigeons) and $2n-1$ numbers of hands to shake (the boxes), some two shakers must shake the same numbers of hands. □

Should we add here the concept of Proof by computer? Used for instance in the 4-color theorem...

2.2.3 Proof by (Finite) Induction

It is crucial that the reader appreciate the fact that proofs by induction, such as we discuss in this section—and exemplify in Section 6.3.2.2.C—are important tools for *verifying* the correctness of alleged results, but induction by itself is not a tool for *discovering* new results.

2.2.3.1 The proof technique

2.2.3.2 Sample proofs: verifying summation formulas

We illustrate the proof technique of (Finite) Induction by proving the correctness of two familiar summation formulas: (1) the sum of the first n positive integers and (2) the sum of the first n odd positive integers.

Proposition 2.1 *For all $n \in \mathbb{N}$,*

$$\begin{aligned} S_n \stackrel{def}{=} \sum_{i=1}^{n} i &\stackrel{def}{=} 1+2+\cdots+(n-1)+n \\ &= \frac{1}{2}n(n+1) \\ &= \binom{n+1}{2}. \end{aligned} \tag{2.1}$$

Proof. For every positive integer m, let $\mathbf{P}(m)$ be the proposition

$$1+2+\cdot+m \;=\; \binom{m+1}{2}.$$

Let us proceed according to the standard format of an inductive argument.

1. Because $\binom{2}{2} = 1$, proposition $\mathbf{P}(1)$ is true.

2. Let us assume, for the sake of induction, that proposition $\mathbf{P}(m)$ is true for all positive integers strictly smaller than n. In particular, then, $\mathbf{P}(n-1)$ is true.

3. Consider now the summation

$$1+2+\cdots+(n-1)+n.$$

Because $\mathbf{P}(n-1)$ is true, we know that

$$1+2+\cdots+(n-1) \;=\; \binom{n}{2}.$$

By direct calculation, we see that

$$\begin{aligned}\binom{n}{2}+n &= \frac{n(n-1)}{2} \;+\; n \\ &= \frac{n^2-n+2n}{2} \\ &= \frac{n^2+n}{2} \\ &= \binom{n+1}{2}\end{aligned}$$

Because n is an arbitrary positive integer, we conclude that $\mathbf{P}(n)$ is true whenever

- $\mathbf{P}(1)$ is true
- *and* $\mathbf{P}(m)$ is true for all $m < n$.

By the Principle of (Finite) Induction, then, we conclude that $\mathbf{P}(n)$ is true for all $n \in \mathbb{N}^+$. $\square$

We turn now to our second summation, which asserts that each perfect square of a positive integer, say, n^2, is the sum of the first n odd integers, $1,3,5,\ldots,2n-1$. This proof complements the constructive proofs of the same result in Proposition 6.3.

Proposition 2.2 *For all $n \in \mathbb{N}^+$,*

$$\sum_{k=1}^{n} (2k-1) \;=\; 1+3+5+\cdots+(2n-1) \;=\; n^2.$$

That, is, the nth perfect square is the sum of the first n odd integers.

Verification. For every positive integer m, let $\mathbf{P}(m)$ be the proposition

$$m^2 \;=\; 1+3+5+\cdots+2m-1.$$

Let us proceed according to the standard format of an inductive argument.

1. Because $1 \cdot 1 = 1$, proposition $\mathbf{P}(1)$ is true.

2. Let us assume, for the sake of induction, that proposition $\mathbf{P}(m)$ is true for all positive integers strictly smaller than n. In particular, then, $\mathbf{P}(n-1)$ is true.

3. Consider now the summation

$$1+3+5+\cdots+2n-3+2n-1$$

Because $\mathbf{P}(n-1)$ is true, we know that

$$1+3+5+\cdots+2n-3+2n-1 \;=\; (n-1)^2+2n-1.$$

By direct calculation, we see that

$$(n-1)^2+2n-1 \;=\; (n^2-2n+1)+(2n-1) \;=\; n^2.$$

Because n is an arbitrary positive integer, we conclude that $\mathbf{P}(n)$ is true whenever

- $\mathbf{P}(1)$ is true
- *and* $\mathbf{P}(m)$ is true for all $m < n$.

By the Principle of (Finite) Induction, then, we conclude that $\mathbf{P}(n)$ is true for all $n \in \mathbb{N}^+$. $\square$

2.2.3.3 Making guesses: the method of undetermined coefficients

Proofs by induction, as provided in Section 6.3.2.2.C, are important tools for *verifying* the correctness of alleged results, but induction by itself is not a tool for *discovering* new results. This section is devoted to the *Method of Undetermined Coefficients*, a tool that sometimes yields the "guesses" that can then be verified via induction. We illustrate the method by deriving a formula for the sum of the first n perfect squares. Our derivation builds on prior knowledge of two facts:

1. A trivial proof by counting verifies that

$$\sum_{k=0}^{n} k^0 \;=\; \sum_{k=1}^{n} 1 \;=\; n.$$

2. We know from Proposition 6.1 that

$$\sum_{k=0}^{n} k^1 \;=\; \sum_{k=1}^{n} k \;=\; \frac{1}{2}(n^2+n)$$

It seems to be silly to include the case $n = 0$ in our summations, since that term does not affect the sum, but that case tells us that the "constant term" in all of these polynomials—i.e., the coefficient of k^0—is always $c_0 = 0$.

Proposition 2.3 *For all* $n \in \mathbb{N}$,

$$\begin{aligned} S_n^{(2)} \;\stackrel{def}{=}\; \sum_{i=1}^{n} i^2 &\stackrel{def}{=} 1+4+\cdots+(n-1)^2+n^2 \\ &= \frac{1}{3}n^3 \;+\; \frac{1}{2}n^2 \;+\; \frac{1}{6}n \end{aligned} \tag{2.2}$$

The target quantity $S_n^{(2)}$ in the proposition is often expressed in a more aesthetic form:

$$S_n^{(2)} \;=\; \frac{1}{6}n(n+1)(2n+1) \;=\; \frac{2n+1}{3}\cdot\binom{n}{2}.$$

Proof. Since summing 0th powers thus gives us a degree-1 polynomial, and summing 1st powers gives us a degree-2 polynomial, it is not unreasonable to guess that summing 2nd powers would give us a degree-3 polynomial. This turns out to be a good guess! To prove this assertion, we must determine values for constants c_1, c_2, c_3 such that

$$\sum_{i=0}^{n} k^2 \;=\; c_3 n^3 + c_2 n^2 + c_1 n. \tag{2.3}$$

(Including the case $n = 0$ leaves us with *three* constants to determine rather than four: we know that the coefficient of k^0 is $c_0 = 0$.)

We begin our determination of the constants c_1, c_2, c_3 by instantiating the symbolic polynomial in (2.3) at the smallest three values of n. Any three values will work; using the *smallest* ones simplifies our calculations. These instantiations leaves us with the following system of linear equations. (The summations in (2.4) indicate where each linear equation in the system comes from.)

$$\begin{aligned}
&1.\ \sum_{i=0}^{1} k^2 = \;\; c_3 \;\; + \;\; c_2 \;\; + \;\; c_1 \;\; = \;\; 1 \\
&2.\ \sum_{i=0}^{2} k^2 = \;\; 8c_3 \;\; + 4c_2 \;\; + 2c_1 = \;\; 5 \\
&3.\ \sum_{i=0}^{3} k^2 = 27c_3 \;\; + \;\; 9c_2 \;\; + \;\; 3c_1 = \;\; 14
\end{aligned} \tag{2.4}$$

We use a form of the *Gaussian elimination* algorithm[4] to solve the system by "eliminating variables." First, we rewrite equation 1 as

$$8c_3 + 8c_2 + 8c_1 \;=\; 8$$

and substract equation 2 from it, thereby obtaining the 2-variable equation

$$4c_2 + 6c_1 \;=\; 3. \tag{2.5}$$

We perform a similar calculation based on equations 1 and 3 in system (2.4): We rewrite equation 1 as

$$27c_3 + 27c_2 + 27c_1 \;=\; 27$$

and substract equation 3 from it, thereby obtaining the 2-variable equation

$$18c_2 + 24c_1 \;=\; 13. \tag{2.6}$$

We now rewrite equations (2.5) and (2.6) to obtain the simplified system

$$\begin{aligned}
72c_2 \;+\; 108c_1 &= 54 \\
72c_2 \;+\; 96c_1 \;\; &= 52
\end{aligned}$$

[4] This algorithm is defined and validated in sources such as [28].

We now see, via subtraction, that

$$12c_1 \;=\; 2$$

or, equivalently,

$$c_1 \;=\; 1/6. \tag{2.7}$$

Now that we know the value of c_1, we can make the indicated substitution and further simplify system (2.4). Since we now have only two variables, we can also eliminate any one of the three equations in (2.4). We eliminate equation 3 in the system in order to simplify our calculations from this point on. We now have the system

$$\begin{array}{l} 1.\;\; c_3 \;+\; c_2 \;=\; 5/6 \\ 2.\; 8c_3 + 4c_2 = 14/3 \end{array} \tag{2.8}$$

We rewrite equation 1 in (2.8) as

$$8c_3 + 8c_2 \;=\; 20/3$$

and subtract equation 2 from this version of equation 1. We thereby discover that

$$4c_2 \;=\; 2,$$

or, equivalently,

$$c_2 \;=\; 1/2. \tag{2.9}$$

Using the values we have discovered for c_1 and c_2, in equations (2.7) and (2.9), respectively, we finally use equation 1 of system (2.4) to determine the value of c_3:

$$c_3 \;=\; 1 - \;1/2 \;-\; 1/6 \;=\; 1/3. \tag{2.10}$$

We have, thus, derived equation (2.2).

The careful reader will note that we are not really done yet. We have derived our expression for $S_n^{(2)}$ under the as-yet unjustified assumption that $S_n^{(2)}$ really is a cubic (i.e., degree-3) polynomial. What we need now is an induction that will verify our result. With our previous illustrations as models, we shall leave this final task to the reader.

****

IS THIS OK?

********** □

With more (calculational) work, but no new (mathematical) ideas, one can derive explicit expressions for the sum of the first n kth powers, i.e., the quantity $S_n^{(k)}$, for any positive integer k.

2.2.4 Proof by Contradiction

The importance of this proof technique has been recognized since antiquity, under the Latin names contradictio in contrarium *and, perhaps less accurately,* reductio ad absurdum.

2.2.4.1 The Proof Technique

The basic principle that underlies proof by contradiction is that the following *metamathematical* assertions are logically equivalent.

- Proposition P IMPLIES Proposition Q
- Proposition $\sim Q$ IMPLIES Proposition $\sim P$

As with many of these *metamathematical* principles, there is a corresponding *mathematical* equivalence, in this case, Proposition 3.4.

$$[P \Rightarrow Q] \;\equiv\; [(\sim Q) \Rightarrow (\sim P)]$$

2.2.4.2 Sample Proofs

A. There are infinitely many primes

The following result is traditionally attributed to the Greek mathematician Euclid, one of the patriarchs of mathematics.

Proposition 2.4 *There are infinitely many prime numbers.*

Proof. Let us assume, contrarily, that there are only finitely many primes. Say, in particular, that the following r-element sequence enumerates all (and only) primes, in order of magnitude:

Prime-Numbers $= \langle P_1,\, P_2,\, \ldots,\, P_r \rangle$
where

- $P_1 = 2$
- $P_2 = 3$
- $P_i < P_{i+1}$ for all $i \in \{1, 2, \ldots, r-1\}$.

We verify the *falseness* of the alleged completeness of the sequence **Prime-Numbers** by analyzing the positive integer

$$n \;=\; 1 + \prod_{i=1}^{r} P_i \;=\; 1 \;+\; (P_1 \cdot P_2 \cdot \cdots \cdot P_r)\,.$$

We make three crucial observations.

1. We note first that *the number n is not divisible by any prime number in the sequence* **Prime-Numbers**.
   To see this, note that for each P_k in the sequence,
   $$n/P_k \;=\; \frac{1}{P_k} \;+\; \prod_{i\neq k} P_i.$$
   Because $P_k \geq 2$, we see that n/P_k obeys the inequalities
   $$\prod_{i\neq k} P_i \;<\; n/P_k \;<\; 1+\prod_{i\neq k} P_i.$$
   The discreteness of the set $\mathbb{Z}$—see Section 4.3.A—implies that n/P_k is not an integer, because it lies strictly between two adjacent integers.
2. We note next that, because of assertion 1, if the sequence **Prime-Numbers** actually contained *all* of the prime numbers, then we would have to conclude that *the number n is not divisible by any prime number.*
3. Finally, we remark that the Fundamental Theorem of Arithmetic (Theorem 7.1) implies that *every integer is divisible by (at least one) prime number*.

We have a chain of assertions that lead to a mutual inconsistency: on the one hand, the integer n has no prime-integer divisor; on the other hand, no integer can fail to have a prime-integer divisor! Let us analyze how we arrived at this uncomfortable place.

- At the front end of this uncomfortable string of assertions we have the assumption that there are only finitely many prime numbers. We have (as yet) no substantiation for this assertion.
- At the back end of this uncomfortable string of assertions we have the (*rock solid*) Fundamental Theorem of Arithmetic (Theorem 7.1).
- In between these two assertions we have a sequence of assertions, each of which follows from its predecessors via irrefutable rules of inference.

It follows that the *only* brick in this edifice that could be faulty—i.e., the only assertion that could be false—is the assumption that there are only finitely many prime numbers. Since this assumption leads to an inconsistent set of assertions, we must conclude that the assumption is false! We conclude from this classical proof by contradiction that there are infinitely many prime numbers. □

2.2.5 Geometrical and graphical proofs

2.2.5.1 An old and simple example

2.2.5.2 Fubini's principle

[35]

2.2.6 Proofs via the Pigeonhole Principle

Is it a technique by itself like the other ones or should it be integrated into another one – on thus, which one?

The proof technique we discuss now builds on an observation that is almost embarrassingly obvious—yet its simplicity is exceeded by its importance as a source of strikingly surprising results.

2.2.6.1 The Proof Technique

The technique, known variously as *the pigeonhole principle* or *Dirichlet's Box Principle* (after the French mathematician Peter Gustav Lejeune Dirichlet), exploits the fact that if one has n objects (say, pigeons) and $m < n$ boxes (they're the pigeonholes), then any way of putting pigeons into boxes must place at least two pigeons into the same box.

2.2.6.2 Sample (Fun) Applications/Proofs

A. Choosing a pair of matching socks

You have n pairs of socks, the socks in each pair having a distinct color (one pair of red socks, one pair of blue socks, ...). Since you wake up "very slowly", you want to grab some number of unpaired socks that is certain to yield at least one pair of same-color socks. Clearly, if you grab any $n+1$ socks (the pigeons), the pigeonhole principle guarantees that you have at least one monochromatic pair, because there are only n distinct sock-colors (the boxes).

B. Finding birthday-mates

You are attending a conference and wander into a lecture that has 367 attendees (including you). It is certain that at least two attendees share the same birthday: there are 366 possible birthdays (the boxes for a leap year) and 367 birthday-possessors (the pigeons).

May be we can put this in exercice and add the anniversary paradox which state a similar question with probabilities?

D. Friends and strangers at a party

We turn now to a somewhat more surprising result that can be proved using the pigeonhole principle. While we phrase the result in anthropomorphic, "homely",

terms, its formal statement identifies it as a genre of "unavoidable subgraph" phenomenon within the theory of *graphs*.

> Graphs are an immensely important mathematical construct that models myriad situations that involve objects (possibly people) and interrelationships between pairs of objects. Chapter 10 is devoted to studying graphs and the situations they can be used to model—including the problem discussed here.

Here is the "homely" version of the *Friends and Strangers* problem.

Proposition 2.5 *In any gathering of six people, at least one of the following assertions is true.*
A. *There is a group of three people who know each other.*
B. *There is a group of three people none of whom knows either of the others.*

Proof. Let the gathering consist of six indistinguishable people, named P_1, P_2, P_3, P_4, P_5, P_6. Focus on an arbitrary person, say P_5. (This choice "sounds" more arbitrary than P_1—but, of course, is not.) Now, there are 5 people, namely, P_1, P_2, P_3, P_4, P_6, each of whom P_5 either *knows* or *doesn't know*.

Clearly, some 3 of these 5 people "lie on the same side of the *know/don't-know* fence." This follows from the pigeonhole principle: we have *two* boxes (*know* and *doesn't know*) and *five* pigeons (the people P_1, P_2, P_3, P_4, P_6). Any way of putting the pigeons into the boxes will place three people into one of the boxes.

Say, with no loss of generality, that P_5 *knows* P_1, P_2, P_3.

> Why can we claim that the selected situation— "P_5 *knows* P_1, P_2, P_3"—can be assumed "with no loss of generality"? One should *always* ask this question about such a claim! In the current case, the claim follows from the following facts.
>
> (a) The names that we use to refer to the six assembled people are just for our expository benefit. The names carry no inherent meaning related to the *Friends and Strangers* problem. You can repeat our argument while choosing arbitrary replacements for P_1, P_2, P_3, P_5, with no change to the logical outcome.
>
> You can also interchange the *know* and *don't-know* labels. The underlying logic will not change, although the conclusions regarding options A and B in the statement of the proposition will clearly "flip".

Having decided that P_5 *knows* P_1, P_2, and P_3, we now consider the implications of the possible relations between each of the three pairs of people chosen from $\{P_1, P_2, P_3\}$. There are two logical possibilities.

- Some two of P_1, P_2, P_3 know each other—say, with no loss of generality, P_1 and P_2. In this case, P_1, P_2, and P_5 form a trio of people who know one another (option A in the statement of the proposition).
- No two of P_1, P_2, P_3 know each other. In this case, P_1, P_2, and P_3 form a trio of people none of whom knows either of the others (option B in the statement of the proposition).

This disjunction completes the proof.

We close the proof by noting that nothing we have stated precludes the possibility that *both* option A *and* option B are true! □

2.3 Bijections between Sets and Combinatorial Proofs

A nice example but rather complicated here may be the proof of the little Fermat theorem

2.4 Reasoning via Mathematical Analysis

I am not clear about this section. What should it contain?

2.4.1 Asymptotics

Asymptotics can be viewed as a language and a system of reasoning that allow one to talk in a *qualitative* voice about *quantitative* topics. We thereby generalize to arbitrary growth functions terms such as "linear", "quadratic", "exponential", and "logarithmic".

Such a language and system are indispensable if one needs to reason about computational topics over a range of situations, such as a range ("all existing"?) computer architectures and software systems. As two simple examples: (1) Carry-ripple adders perform additions in time linear in the lengths n of the summands (measured in number of bits) no matter what these lengths are. (2) Comparison-based sorting algorithms can sort lists of n keys in time proportional to $n \log n$, but no faster—where the base of the logarithm depends on the characteristics of the computing platform. More precise versions of the preceding statements require specication of the number n and other details, possibly down to the clock speeds of the host computer's circuitry.

2.4.1.1 The language of asymptotics

The language of asymptotics, which has its origins in the field of Number Theory in the late 19th century, builds on the following terminology, which is likely what one would cover in an early undergraduate course. More advanced aspects of the language would likely by beyond the needs of most students of computing, aside from specialists in advanced courses. The basics of the language build on three primitive notations and notions. Standard sources, such as any text on algorithm design and analysis, flesh out the following ideas.

- *The big-O notation*. The assertion $f(x) = O(g(x))$ says, intuitively, that the function f grows no faster than function g. It is, thus, the asymptotic analogue of "less than".
  Formally: $f(x) = O(g(x))$

means
$(\exists c > 0)(\exists x^{\#})(\forall x > x^{\#})[f(x) \leq c \cdot g(x)]$

- *The big-Ω notation*. The assertion $f(x) = \Omega(g(x))$ says, intuitively, that the function f grows at least as fast as function g. It is, thus, the asymptotic analogue of "greater than".
  Formally: $f(x) = \Omega(g(x))$
  means
  $(\exists c > 0)(\exists x^{\#})(\forall x > x^{\#})[f(x) \geq c \cdot g(x)]$

- *The big-Θ notation*. The assertion $f(n) = \Theta(g(n))$ says, intuitively, that the function f grows at the same rate as does function g. It is, thus, the asymptotic analogue of "equal to".
  Formally: $f(x) = \Theta(g(x))$
  means
  $(\exists c_1 > 0)(\exists c_2 > 0)(\exists x^{\#})(\forall x > x^{\#})[c_1 \cdot g(x) \leq f(x) \leq c_2 \cdot g(x)]$

One renders the preceding intuitive explanations precise by pointing out that the three specifies relations (*a*) take hold *eventually*, i.e., only for large arguments to the functions f and g, and (*b*) hold up to an unspecified constant of proportionality.

2.4.1.2 The "uncertainties" in asymptotic relationships

The formal definitions of all three of our asymptotic relationships are bracketed by two important quantifiers:

$$\text{“}(\exists c > 0)\text{”} \quad \text{and “}(\forall x > x^{\#})\text{”}.$$

The former, *uncertain-size* quantifier, asserts that asymptotic notions describe functional behavior "in the large". Thus, in common with more common qualitative descriptors of quantitative growth such as linear, quadratic, cubic, quartic, exponential, logarithmic, etc., asymptotic relationships give no infomation about constants of proportionality. *We are not saying that constant factors do not matter! We are, rather, saying that we want to discuss growth patterns in the large.*

The latter, *uncertain-time* quantifier asserts that asymptotic relationships between functions are promised to hold only "eventually", i.e., "for sufficiently large values of the argument x". Therefore, in particular, asymptotic notions cannot be employed to discuss or analyze quantities that can never grow beyond a fixed finite value. The fact that all instances of a quantity throughout history have been below N is immaterial, as long as it is conceivable that an instance larger than N could appear at some time in the future.

These quantifiers in particular distinguishes claims of asymptotic relationship from the more familiar definite inequalities such as "$f(x) \leq g(x)$" or $f(x) \geq 7 \cdot g(x)$. In fact, it is often easier to think about our three asymptotic bounding assertions as establishing *envelopes* for $f(x)$:

- Say that $f(x) = O(g(x))$. If one draws the graphs of the functions $f(x)$ and $c \cdot g(x)$, then as one traces the graphs with increasing values of x, one eventually reaches a point $x^{\#}$ beyond which the graph of $f(x)$ never enters the territory *above* the graph of $c \cdot g(x)$.
- Say that $f(x) = \Omega(g(x))$. This situation is the up-down mirror image of the preceding one: just replace the highlighted "*above*" with "*below*."
- Say that $f(x) = \Theta(g(x))$. We now have a two-sided envelope: beyond $x^{\#}$, the graph of $f(x)$ never enters the territory *above* the graph of $c_1 \cdot g(x)$ and never enters the territory *below* the graph of $c_2 \cdot g(x)$.

In addition to allowing one to make familiar growth-rate comparisons such as "$n^{14} = O(n^{15})$" and "$1.001^n = \Omega(n^{1000})$," we can now also make assertions such as "$\sin x = \Theta(1)$," which are much clumsier to explain in words.

Beyond the big letters. There are "small"-letter analogues of the preceding "big"-letter asymptotic notations, but they are only rarely encountered in discourse about real computations (although they do arise in the analysis of algorithms).

2.4.1.3 Inescapable complications

The story we have told thus far is covered in many sources and courses. Two complications to the story are covered less faithfully, although lacking them, one cannot perform cogent asymptotic reasoning. Both complications involve the notion of *uniformity*.

1. *Multiple functions*. Say that we have four functions, f, g, h, k, and we know that both

$$f(n) = O(g(n)) \quad \text{and} \quad h(n) = O(k(n))$$

It is intuitive that

$$f(n) + h(n) = O(g(n) + k(n))$$

— but is it true?

In short, the answer is YES, but verifying that requires a bit of subtlety, because, absent hitherto undisclosed information, the proportionality constants $c_{f,g}$ and $N_{f,g}$ that witness the big-O relationship between functions f and g have no connection with the constants, $c_{h,k}, N_{h,k}$ that witness the analogous relationship between functions h and k. Therefore, in order to verify the posited relationship between functions $f+h$ and $g+k$, one much find witnessing constants $c_{f+h,g+k}$ and $N_{f+h,g+k}$. Of course, this task requires only elementary reasoning and manipulation — but it must be done!

2. *Multivariate functions*. Finally, we discuss the scenario that almost automatically accompanies the transition from a focus on sequential, single-agent computing to a focus on PDC. Within this broadened context, most functions that describe a system have one or more variables that describe the computing system — its number of processors or of agents or the sizes of its memory modules or the communication-radii of its transponders or ..., in addition to the one or more variables that describe

the data that the system is processing. Within such scenarios, every assertion of an asymptotic relationship, of the form

$$f(\mathbf{m};\mathbf{n}) = O(g(\mathbf{m};\mathbf{n}))$$

must explicitly specify the following information:

- which variables can grow without bound;
- among such unbounded variables, which participate in the posited asymptotic relation;
- for each participating unbounded variable x, what are the constants c_x and N_x that witness the posited asymptotic relationship(s).

Clearly the complexity of cogent asymptotic reasoning — hence also the complexity of teaching about such reasoning — gets much more complicated in the multivariate settings engendered by PDC. But, the benefits of being able to reason qualitatively about the quantitative aspects of computing increase at least commensurately!

2.5 Coping with Infinity

is it the right place here?

2.5.1 Reasoning about Infinity

This part should be carefully read

An immediate problem is the forward reference to SUMMATION. Do we need to reorganize?

We presented in the chapter Summations several ways to compute the sum of numbers in a given sequence (a_n). If the sequence contains infinitely many, the sum can be finite or not. Some of the classical techniques used while calculating finite sums are no longer valid for infinite sums.

A natural definition is $\sum_{k\geq 1} a_k = lim_{n\to\infty} \sum_{k=1}^{n} a_k$. But we have to be careful with this definition. For instance, let us consider the following paradoxal situation: we aim to determine the value of $S = \sum_{k\geq 0} \frac{1}{2^k}$. Defining the infinite sum as the limit leads to the value 2. This is obtained $2S = S + 2$... But what happens if we apply the same reasoning to to the sum: $\sum_{k\geq 0} 2^k$? Te obtain the value -1! This is obviously not correct since the sum of increasing positive numbers should be positive. The reason is that the terms of the series grows to $+\infty$.

2.5.2 The "Point at Infinity"

In large part, the difficulties encountered when dealing with infinite objects result from the conceptual fiction that there is, in fact, a "point at infinity"—i.e., that one can treat infinity as just another number. In many mathematical environments, this fiction is an aid to reasoning which can be handled with totally rigor—but often only when accompanied by rather sophisticated mathematical machinery. Two familiar examples of such mathematical machinery are the notions of *limit* and *continuity* (of a function). A more advanced example of such machinery is the *Riemann sphere*, an invention of the 19th-century German mathematician Bernhard Riemann, which allows one to reason about the infinite two-dimensional plane by "conformally" wrapping the plane into a sphere whose "south pole" represents the zero-point (i.e., the origin) of the plane and whose "north pole" represents the "point at infinity." There are many other, less-familiar, examples of such mathematical machinery, including advanced topics such as *types* in the domain of mathematical logic.

In the remainder of this text beyond the current section, we deal successfully with a number of quite sophisticated topics related to infinity. To cite just two examples:

1. In Chapter 4 we successfully answer questions such as
   a. *Are there more rational numbers than integers?*
   b. *Are there more real numbers than rational numbers?*
2. In Chapters 4 and 6, we successfully sum and manipulate infinite summations.

The lessons of the preceding paragraphs is that there is no need to avoid dealing with infinity and its related notions—as long as one has the mathematical machinery necessary to define all needed notions unambiguously, obtain well-defined results from all required operations and manipulations, and reason cogently about all concepts and processes. That said, the scenarios described in the following two subsections warn us to treat all aspects of infinity with care and respect. The subsections point out two challenges one can encounter when reasoning about the infinite. Both challenges leave us with a *paradox*, i.e., "a statement or proposition that, despite sound (or apparently sound) reasoning from acceptable premises, leads to a conclusion that seems senseless, logically unacceptable, or self-contradictory." (Apple dictionary)

2.5.2.1 Underspecified problems

The paradoxes we present now require refined groundrules for their resolution. The underlying problems *seem* to be totally specified—until one tries to develop their solutions.

A. An infinite summation

The first question we tackle was the subject of much concern as the topic of infinite summations emerged from its infancy in the 18th century. The overriding question is, What can one learn from infinite series that do not have a unique sum? Much valuable work was done on this question, most being beyond the scope of this text. But the conundrum presented by the following, particularly vexing, infinite summation is valuable to consider here.

$$S = 1 - 1 + 1 - 1 + 1 - 1 + \cdots$$

There are many conflicting, but well-reasoned, answers to the following questions.
Questions: *Does S have a finite value? If not, then is S positive or negative?*
Here are three plausible answers to these questions; the third merits some strong contemplation.

1. $S = 0$
   This response is justified by the following association of terms in the summation that defines S.

$$\begin{aligned} S &= (1 - 1) + (1 - 1) + (1 - 1) + \cdots \\ &= 0 + 0 + 0 + \cdots \\ &= 0 \end{aligned}$$

2. $S = -\infty$
   This response is justified by the following association of terms in the summation that defines S.

$$\begin{aligned} S &= 1 - (1 + 1) - (1 + 1) - (1 + 1) + \cdots \\ &= 1 - 2 - 2 - 2 - \cdots \\ &= -\infty \end{aligned}$$

3. There is no valid answer, because the problem statement does not specify how to associate terms. Indeed, by mischievously playing with parentheses, one can arrive at many "plausible" values for S. This is one concrete example of how summations with infinitely many terms behave differently from those with finitely nany terms.

B. The Ross-Littlewood paradox

The following story about balls and bins is known as the *Ross-Littlewood paradox*, after its creators: A version of the story appeared first in John Littlewood's enlightening and entertaining book *Littlewood's Miscellany;* [54] the story was amplified to its present form by S.M. Ross [74].

Let us imagine a system that contains

- a *really big* bin (in fact, one whose capacity grows as needed, as the story progresses)
- an unbounded sequence of ordered balls, labelled 1, 2, ...
- a *very* (read: infinitely) precise clock.

The system is watched over by a *Keeper*. We observe the *Keeper* executing the following process.

Step 1 of the process occurs at time *midnight minus* 1 *minute*. The *Keeper* places the first ten balls in the sequence (balls #$1,\ldots,10$) into the bin, and *immediately* removes the first ball (ball #1).

Step 2 of the process occurs at time *midnight minus* $1/2$ *minute*. The *Keeper* places the next ten balls in the sequence (balls #$11,\ldots,20$) into the bin, and *immediately* removes the second ball (ball #2).

The *Keeper* repeats this process endlessley, at midnight minus $1/4$ minute (putting balls #$21,\ldots,30$ into the bin and removing ball #3), then at midnight minus $1/8$ minute (putting balls #$31,\ldots,40$ into the bin and removing ball #4), and on, and on.

Question: *How many balls are present in the bin at midnight?* (Note that "infinity" now measures the number of steps executed in the process.)

As in paragraph A, there are many plausible answers to this question. We provide just three.

1. There are infinitely many balls.
   This response is justified by the following reasoning. Each step of the process inserts 10 balls into the bin but removes only 1 ball. Hence, the population of the bin grows by 9 balls after every step of the process—and it never decreases! Hence, the bin's population after infinitely many steps must be infinite.
2. There are 0 balls—the bin is empty!.
   This response is justified by the following reasoning. Every ball is eventually removed from the bin at some (finite) step of the process. Specifically, ball #n is removed at step n, i.e., at time midnight minus 2^{-n} seconds.
3. There is no valid answer, because there is no "moment at infinity" that is encountered after an infinite number of steps of the process. In other words, infinity is not a number!

C. Zeno's paradox: Achilles and the tortoise

In his celebrated *Paradox of Achilles and the Tortoise*, Zeno of Elea presented a problem whose solution had to await the 17th century. In the story, the slow-footed Tortoise (T) tries to convince the speedy Achilles (A) of the futility of trying to win any race in which A gives T even the smallest head start. As long as T is ahead of A, says T, every time A traverses half the distance between himself and T, T will respond by moving a bit further ahead. Thereby, T will always be a positive distance ahead of A, so that A can *never* catch T. At first glance, this story seems to call into question the physical reality of all motion.

The resolution of the apparent paradox resides in the notion of *infinitesimals*—quantities that dynamically grow smaller than any finite number. While familiar today to anyone who has studied subjects such as the differential calculus, the notion of infinitesimal actually dates back only a few hundred years, to the 17th century. Underlying the discovery/invention of infinitesimals is one of the great real-life mysteries of all time: Who invented/discovered infinitesimals. The parties to this dispute were the German mathematician Gottfried Leibniz [51] and the English polymath (Sir) Isaac Newton [59]. The cases favoring each of these great men contains enough merit to guarantee that the dispute will likely never be settled. We therefore list Leibniz and Newton alphabetically and give a coarse dating of the 17th century for the discovery. This real-life mystery is as full of intrigues and suspense as any that one encounters in fiction.

D. Hilbert's hotel paradox

While the final story of this section does not actually provide a paradox, it does point out a fundamental difference between the real world of finite capacities and an idealized world that is not so encumbered.

Imagine that you are running a hotel that has an infinite number of rooms which are labeled by the (entire set of) positive integers: there is a room #1, a room #2, a room #3, and so on. Say that on a particular evening, every room of the hotel is occupied by a guest—and then a new guest arrives!

In a desire to accommodate the newcomer, you initiate the following procedure, which was first proposed by the eminent German mathematician David Hilbert.

By means of a broadcast message to all current guests, you move each guest who currently occupies room #k into room #$k+1$. Of course, this total shift renders room #1 available—so you assign this room to the newly arrived guest. The world is quiet once more!

Of course, this humorous story has its roots in a fundamental distinction between the world of finite-capacity hotels that we live in and the idealized infinite-capacity hotel proposed in Hilbert's story. In a word, every finite set of integers—think of the integers as the room numbers in a finite-capacity hotel—*has a largest number*, while an infinite set of positive integers *does not have a largest number*.

2.5.2.2 Foundational paradoxes

The "foundational" paradoxes that we present now can be resolved only via the development of new, sophisticated, mathematical machinery.

A. Gödel's paradox: Self-referentiality in language

Let us focus on the following utterance, which we call "Sentence *S*".

Sentence *S*: *The sentence you are reading at this moment is false.*
Questions: *Is Sentence S true, or not?*
Let us analyze the options.

- On the one hand:
  If Sentence *S* is true, then one must accept its assertion that Sentence *S* is false.
- On the other hand:
  If Sentence *S* is false, then one must reject its assertion that Sentence *S* is false. In other words, one must conclude that Sentence *S* is true.

In the 1930s, the revolutionary philosopher/ logician Kurt Gödel turned the mathematical world on its head with his demonstration that, roughly speaking,

> *Any language that is self-referential—i.e., that can refer to its own sentences as objects of discourse—must contain a sentence such as Sentence S, which is neither true nor false.* [38]

The shocking implication of Gödel's work is that in any sufficiently sophisticated language *L*, the notions *true* and *false* do not totally partition (into two pieces) the set of legitimate utterances in language *L*. The simplicity of Sentence *S* and encodings thereof—see, e.g., [70]—can be used to show that the following classes of languages, and their kin, are "sufficiently sophisticated":

- Natural languages (Swahili, German, Urdu, etc.)
- General-purpose programming languages (assembly language, Basic, C, Java, LISP, Python etc.)
- Quantified mathematical languages—i.e., ones that contain logical quantifiers such as FOR ALL ($\forall$), THERE EXIST ($\exists$), etc.

Of course, the world proceeded before Gödel's earthshaking proof, and it is still spinning after the proof. We are just aware now that we must be more careful in our use of language. For instance, we must employ pre-validated transformations in our compilers and pre-justified "small steps" in our schedulers.

B. Russell's paradox: The absence of an "anti-universal" set

The notion of set is perhaps the most primitive one in mathematics. Even before delving into Chapter 3's survey of the intricacies of sets and their mathematical kin, the reader probably has at least an informal command of many of the relevant notions. One of the most basic features of sets is that their elements are not governed by any *a priori* restrictions. Most specifically for our purposes here, a set can have sets as elements. Indeed, there is no inherent reason why a set cannot contain itself as an element! At first blush, this possibility for self-membership seems to be a rather innocuous freedom. But, the 20th-century English philosopher/logician Bertrand Russell pointed out in [76] that, when coupled within the world of potentially-infinite sets, the capacity for self-membership is (intellectually) hazardous.

Russell had the foresight to observe that, absent any restrictions on the formations of sets, one could talk about the following set *A*, which we shall call "anti-universal."

A is the set of precisely those sets that do not *contain themselves (as elements).*
Question. *Is the set A a member of itself?*
Let us consider the possibilities.

- If set A is a member of itself, then by definition, A *does not* contain itself—since sets belonging to A *do not* contain themselves.
- If set A is not a member of itself, then by definition, A is an element of A.

There have been many attempts to reseolve the dilemma inherent in the preceding analysis. Many have striven for logical edifices that declare the question "*Is the set A a member of itself?*" somehow illegitimate. One options that appeals to many is to assign each sentence within a language L a *type* (say, a positive integer). One then allows a sentence of L to refer only to sentences of lower type-number.

The strategem of typing utterances within a language L disables self-reference within L, hence defines away both Russel's paradox and Gödel's paradox.

Chapter 3
SETS, BOOLEAN ALGEBRA, AND LOGIC

3.1 Sets

3.1.1 Fundamental Set-Related Concepts

Sets are probably the most basic object of mathematical discourse. Sets exist to have *elements*, or *members*, the entities that *belong to* the set. The notion of set is surprisingly difficult to specify formally, so we just assume that *the reader knows what a set is and recognizes that some sets are finite, while others are infinite*. Speaking informally—a formal treatment will follow in later chapters—here are a few illustrative finite sets:

- the set of words in this book
  I do not know how big this set is, but I imagine that you as a reader have a better idea than I as an author.
- the set of characters in any JAVA program
  Note that while we are sure that this set is finite, we are not so confident about the number of seconds the program will run!
- the set consisting of *you*
  Paraphrasing the iconic television figure Mister Rogers, "You are unique." This set has just one element.
- the set of unicorns in New York City
  I will not argue with you about this, but I believe that this is the *empty set* $\emptyset$, which has zero members.

Some familiar infinite sets are:

- the set of *nonnegative integers*
- the set of *positive integers*
- the set of *all integers*
- the set of nonnegative *rational numbers*—which are quotients of integers
- the set of nonnegative *real numbers*—which can be viewed computationally as the set of numbers that admit infinite decimal expansions,

- the set of nonnegative *complex numbers*—which can be viewed as ordered pairs of real numbers,
- the set of *all* finite-length binary strings.
  A *binary string* is a sequence of 0s and 1s. When discussing computer-related matters, one often calls each 0 and 1 that occurs in a binary string a *bit* (for *binary digit*). The term "bit" leads to the term *bit string* as a synonym of *binary string*.

Despite this assumption, we begin the chapter by reviewing some basic concepts concerning sets and operations thereon.

As noted early, sets were created to contain members/elements. We denote the fact that element t *belongs to*, or, *is an element of* set T by the notation $t \in T$. A *subset* of a set T is a set S each of whose members belongs to T. The subset relation occurs in two forms, The *strong* form of the relation, denoted $S \subset T$, says that every element of S is an element of T, but *not* conversely; i.e., T contains (one or more) elements that S does not. The *weak* form of the relation, denoted $S \subseteq T$, is defined as follows:

$$[S \subseteq T] \quad \text{means:} \quad \Big[\textit{either} \quad [S = T] \quad \textit{or} \quad [S \subset T]\Big].$$

For any finite set S, we denote by $|S|$ the *cardinality* of S, which is the number of elements in S. Finite sets having three special cardinalities are singled out with special names. The limiting case of finite sets is the unique *empty set*, which we denote by $\emptyset$; thus, $\emptyset$ is characterized by the equation $|\emptyset| = 0$. (The empty set is often a limiting case of set-defined entities.) If $|S| = 1$, then we call S a *singleton*; and if $|S| = 2$, then we call S a *doubleton*.

It is often useful to have a convenient term and notation for *the set of all subsets of a set S*. This bigger set—it contains $2^{|S|}$ elements when S is finite—is denoted by $\mathscr{P}(S)$ and is called the *power set* of S.[1] Note carefully the two set-relations that we are talking about here:

A set T that is a subset *of set S is an* element *of the set* $\mathscr{P}(S)$.

You should satisfy yourself that the biggest and smallest elements of $\mathscr{P}(S)$ are, respectively, the set S itself and the empty set $\emptyset$.

3.1.2 Operations on Sets

Given two sets S and T, we denote by:

- $S \cap T$ the *intersection* of S and T: the set of elements that belong to *both* S and T.

$$[s \in S \cap T] \quad \text{means} \quad \Big[[s \in S] \textbf{ and } [s \in T]\Big]$$

- $S \cup T$ the *union* of S and T: the set of elements that belong to S, or to T, *or to both*. (Because of the "or both" qualifier, this operation is sometimes called *inclusive*

[1] The name "power set" arises from the relative cardinalities of S and $\mathscr{P}(S)$ for finite S.

union.)

$$[s \in S \cup T] \quad \text{means} \quad \Big[[s \in S] \textbf{ or } [s \in T] \textbf{ or } [s \in S \cap T]\Big]$$

- $S \setminus T$ is the *(set) difference* of S and T: the set of elements that belong to S but not to T.

$$[s \in S \setminus T] \quad \text{means} \quad \Big[[s \in S] \textbf{ and } [s \notin T]\Big]$$

(Particularly in the United States, one often encounters the notation "$S - T$" instead of "$S \setminus T$.")

We illustrate the preceding operations with the sets $S = \{a, b, c\}$ and $T = \{c, d\}$. For these sets:

$$\begin{aligned} S \cap T &= \{c\}, \\ S \cup T &= \{a, b, c, d\}, \\ S \setminus T &= \{a, b\}. \end{aligned}$$

In many set-related situations, the sets of interest will be subsets of some fixed "universal" set U.

> We use the term "universal" as in "universe of discourse," not in the self-referencing sense of a set that contains all other sets as members, a construct (discussed by philosopher-logician Bertrand Russell) which leads to mind-bending paradoxes.

Given a universal set U and a *subset* $S \subseteq U$, we observe the set-inequalities

$$\emptyset \subseteq S \subseteq U.$$

When studying a context within which there exists a universal set U that contains all other sets of interest, we include within our repertoire of set-related operations also the operation of *complementation*

- $\overline{S} \stackrel{\text{def}}{=} U \setminus S$, the *complement* of S (relative to the universal set U).
  For instance, the set of odd positive integers is the complement of the set of even positive integers, relative to the set of all positive integers.

We note a number of basic identities involving sets and operations on them.

- $S \setminus T = S \cap \overline{T}$,
- If $S \subseteq T$, then
  1. $S \setminus T = \emptyset$,
  2. $S \cap T = S$,
  3. $S \cup T = T$.

Note, in particular, that[2]

$$[S = T] \text{ iff } \Big[[S \subseteq T] \text{ and } [T \subseteq S]\Big] \text{ iff } \Big[(S \setminus T) \cup (T \setminus S) = \emptyset\Big].$$

[2] "iff" abbreviates the common mathematical phrase, "if and only if."

The operations union, intersection, and complementation—and operations formed from them, such as set difference—are usually called the *Boolean (set) operations*, (named for the 19th-century English mathematician George Boole). There are several important identities involving the Boolean set operations. Among the most frequently invoked are the two "laws" attributed to the 19th-century French mathematician Auguste De Morgan:

$$\text{For all sets } S \text{ and } T\text{:} \quad \begin{cases} \overline{S \cup T} = \overline{S} \cap \overline{T}, \\ \\ \overline{S \cap T} = \overline{S} \cup \overline{T}. \end{cases} \tag{3.1}$$

(Algebraic) Closure. We end this section with a set-theoretic definition that occurs in many contexts. Let $\mathscr{C}$ be any (finite or infinite) collection of sets, and let S and T be two elements of $\mathscr{C}$. (Note that $\mathscr{C}$ is a set whose elements are sets.) Think, e.g., of the concrete example of set intersection.

We say that $\mathscr{C}$ is *closed* under intersection if whenever sets S and T (which could be the same set) both belong to $\mathscr{C}$, the set $S \cap T$ also belongs to $\mathscr{C}$. By De Morgan's laws, $\mathscr{C}$'s closure under union implies also its closure under intersection.

3.2 Binary Relations

3.2.1 The Formal Notion of Binary Relation

We begin our discussion of relations by adding a new (binary) set operation to our earlier repertoire. Given (finite or infinite) sets S and T we denote by $S \times T$ the *direct product* of S and T, which is the set of all *ordered pairs* whose first coordinate contains an element of S and whose second coordinate contains an element of T. For example, if $S = \{a, b, c\}$ and $T = \{c, d\}$, then

$$S \times T \ = \{\langle a,c\rangle, \langle b,c\rangle, \langle c,c\rangle, \langle a,d\rangle, \langle b,d\rangle, \langle c,d\rangle\}$$

The direct-product operation on sets affords us a simple, yet powerful, formal notion of binary relation.

Given (finite or infinite) sets S and T, a *relation* ρ *on* S *and* T (in that order) is any subset

$$\rho \ \subseteq \ S \times T.$$

When $S = T$, we often call ρ a *binary relation on (the set)* S ("*binary*" because there are *two* copies of set S being related by ρ).

Relations are so common that we use them in every aspect of our lives without even noticing them. The relations "equal to," "less than," and "greater than or equal to" are simple examples of binary relations on the integers. These same three relations apply also to other familiar number systems such as the rational and real numbers; only "equal," though, holds (in the natural way) for the complex numbers.

Some subset of the three relations "is a parent of," "is a child of," and "is a sibling of" probably are binary relations on (the set of people constituting) your family. To mention just one relation with distinct sets S and T, the relation "A is taking course X" is a relation on

$$(\text{the set of all students}) \times (\text{the set of all courses}).$$

By convention, when dealing with a binary relation $\rho \subseteq S \times T$, we often write "$s\rho t$" in place of the more stilted notation "$\langle s,t\rangle \in \rho$." For instance we (almost always) write "$5 < 7$" in place of the strange-looking (but formally correct) "$\langle 5,7\rangle \in <$."

The following operation on relations occurs in many guises, in almost all mathematical theories. Let ρ and ρ' be binary relations on a set S. The *composition* of ρ and ρ' (in that order) is the relation

$$\rho'' \stackrel{\text{def}}{=} \Big\{ \langle s,t\rangle \in S \times S \mid (\exists t \in S) \Big[[s\rho t] \text{ and } [t\rho' u] \Big] \Big\}.$$

Note that we have used both of our notational conventions for relations here. We also encounter here, for the first time in the text, but certainly not the last, a new notational convention: the common "shorthand" compound symbol "$\stackrel{\text{def}}{=}$": The sentence "$X \stackrel{\text{def}}{=} Y$" should be read "*$X$ is (or, equals), by definition, Y*."

The operation of composition of relations is quite important in the study of "relational" databases.

It is important to be able to assert that elements $s,t \in S$ are *not* ρ-related, i.e., $\langle s,t\rangle \notin S \times S$. Several notations have been developed for this purpose.

Relation	Notation	Negation	Standard?
set membership	$\in$	$\notin$	yes
equality	$=$	$\neq$	yes
less than (strong)	$<$	$\not< \text{ or } \geq$	yes
less than (weak)	$\leq$	$\not\leq \text{ or } >$	yes
greater than (strong)	$>$	$\not> \text{ or } \leq$	yes
greater than (weak)	$\geq$	$\not\geq \text{ or } <$	yes
generic	ρ	$\sim \rho$ or $\widetilde{\rho}$	no

(3.2)

There are several special classes of binary relations that are so important that we must single them out immediately, in the following subsections.

3.2.2 Order Relations

A binary relation ρ on a set S is a *partial order relation*, or, more briefly, is a *partial order* if ρ is transitive. This means that, for all elements $s,t,u \in S$,

$$\text{if } \; sRt \; \text{ and } \; tRu \; \text{ then } \; sRu. \tag{3.3}$$

The qualifier "partial" warns us that some pairs of elements of S do not occur in relation ρ. Number-related orders supply an easy illustrative example. Given any two distinct integers, m and n, one of them must be less than the other: either $m < n$, or $n < m$. In contrast, if we consider *ordered pairs* of integers, then there are pairs of pairs that are not related by the "less than" relation in any natural way. For instance, even though we may agree that, by a natural extension of the number-ordering relation "less than", $\langle 4, 17\rangle$ is "less than" $\langle 22, 19\rangle$, we might well not agree on which of $\langle 4, 22\rangle$ and $\langle 19, 17\rangle$ is less than the other—or, indeed, whether either is "less than" the other.

In many domains, order relations occur in two "flavors", *strong* and *weak*. For many such relations ρ—consider, e.g., "less than" on the integers—the weak version is denoted by underscoring the strong one's symbol. This will be our convention. Just as $\leq$ denotes the weak version of $<$, and $\geq$ denotes the weak version of $>$, we shall denote the weak version of a generic order ρ by $\underline{\rho}$.. Strong and weak versions of an order relation ρ (denoted, respectively, ρ and $\underline{\rho}$) are distinguished by their behavior under simultaneous membership. For illustration, instantiate the following template with ρ being "$<$" and with ρ being "$>$":

For a strong order ρ: **if** $[s \; \rho \; t]$, **then** $[t \; \widetilde{\rho} \; s]$
For the weak version $\underline{\rho}$ of ρ: **if** $[s \; \underline{\rho} \; t]$ **and** $[t \; \underline{\rho} \; s]$, **then** $[s = t]$.

3.2.3 Equivalence Relations

A binary relation R on a set S is an *equivalence relation* if it enjoys the following three properties:

1. R is *reflexive:* for all $s \in S$, we have sRs.
2. R is *symmetric:* for all $s, s' \in S$, we have sRs' whenever $s'Rs$.
3. R is *transitive:* for all $s, s', s'' \in S$, whenever we have sRs' and $s'Rs''$, we also have sRs''.

Sample familiar equivalence relations are:

- The equality relation, $=$, on a set S which relates each $s \in S$ with itself but with no other element of S.
- The relations $\equiv_{12}$ and $\equiv_{24}$ on integers, where[3]

  1. $n_1 \equiv_{12} n_2$ if and only if $|n_1 - n_2|$ is divisible by 12.
  2. $n_1 \equiv_{24} n_2$ if and only if $|n_1 - n_2|$ is divisible by 24.

  We use relation $\equiv_{12}$ (without formally knowing it) whenever we tell time using a 12-hour clock and relation $\equiv_{24}$ whenever we tell time using a 24-hour clock.

[3] As usual, $|x|$ is the *absolute value*, or, *magnitude* of the number x. That is, if $x \geq 0$, then $|x| = x$; if $x < 0$, then $|x| = -x$.

Closely related to the notion of an equivalence relation on a set S is the notion of a *partition* of S. A partition of S is a nonempty collection of subsets $S_1, S_2, \ldots$ of S that are

1. *mutually exclusive:* for distinct indices i and j, $S_i \cap S_j = \emptyset$;
2. *collectively exhaustive:* $S_1 \cup S_2 \cup \cdots = S$.

We call each set S_i a *block* of the partition.

One verifies the following Proposition easily.

Proposition 3.1 *A partition of a set S and an equivalence relation on S are just two ways of looking at the same concept.*

To verify this, we note the following.

Getting an equivalence relation from a partition. Given any partition $S_1, S_2, \ldots$ of a set S, define the following relation R on S:

sRs' if and only if s and s' belong to the same block of the partition.

Relation R is an equivalence relation on S. To wit, R is reflexive, symmetric, and transitive because collective exhaustiveness ensures that each $s \in S$ belongs to some block of the partition, while mutual exclusivity ensures that it belongs to only one block.

Getting a partition from an equivalence relation. To obtain the converse, focus on any equivalence relation R on a set S. For each $s \in S$, denote by $[s]_R$ the set

$$[s]_R \stackrel{\text{def}}{=} \{s' \in S \mid sRs'\};$$

we call $[s]_R$ *the equivalence class of s under relation R.*

The equivalence classes under R form a partition of S. To wit: R's reflexivity ensures that the equivalence classes collectively exhaust S; R's symmetry and transitivity ensure that equivalence classes are mutually disjoint.

The *index* of the equivalence relation R is its number of classes—which can be finite or infinite.

Let[4] $\equiv_1$ and $\equiv_2$ be two equivalence relations on a set S. We say that the relation $\equiv_1$ *is a refinement of* (or, *refines*) the relation $\equiv_2$ just when each block of $\equiv_1$ is a subset of some block of $\equiv_2$. We leave to the reader the simple verification of the following basic result.

Theorem 3.1. *The equality relation, $=$, on a set S refines every equivalence relation on S. In this sense, it is the finest equivalence relation on S.*

3.2.4 Functions

One learns early in school that a function from a set A to a set B is a rule that assigns a unique value from B to every value from A. Simple examples illustrate that this

[4] Conforming to common usage, we typically use the symbol $\equiv$, possibly embellished by a subscript or superscript, to denote an equivalence relation.

notion of function is more restrictive than necessary. Think, e.g., of the operation *division* on integers. We learn that division, like multiplication, is a function that assigns a number to a given pair of numbers. Yet we are warned almost immediately not to "divide by 0": The quotient upon division by 0 is "undefined." So, division is not quite a function as envisioned our initial definition of the notion. Indeed, in contrast to an expression such as "$4 \div 2$," which should lead to the result 2 in any programming environment,[5] expressions such as "$4 \div 0$" will lead to wildly different results in different programming environments. Since "wildly different" is anathema in any mathematical setting, we deal with situations such as just described by broadening the definition of "function" in a way that behaves like our initial simple definition under "well-behaved" circumstances and that extends the notion in an intellectually consistent way under "ill-behaved" circumstances. Let us begin to get formal.

A *(partial) function from set S to set T* is a relation $F \subseteq S \times T$ that is *single-valued;* i.e., for each $s \in S$, there is *at most* one $t \in T$ such that sFt. We traditionally write "$F : S \to T$" as shorthand for the assertion, "F is a function from the set S to the set T"; we also traditionally write "$F(s) = t$" for the more conservative "sFt." (The single-valuedness of F makes the nonconservative notation safe.) We often call the set S the *source (set)*, or, the *domain* of function F, and we call set T the *target (set)* or, the *range* of function F. When there is always a (perforce, unique) $t \in T$ for each $s \in S$, then we call F a *total* function.

You may be surprised to encounter functions that are not total, because most of the functions you deal with daily are *total*. Our mathematical ancestors had to do some fancy footwork in order to make your world so neat. Their choreography took two complementary forms.

1. They expanded the target set T on numerous occasions. As just two instances:
   - They appended both 0 and the negative integers to the preexisting positive integers[6] in order to make subtraction a total function.
   - They appended the rationals to the preexisting integers in order to make division (by nonzero numbers!) a total function.

   The irrational algebraic numbers, the nonalgebraic real numbers, and the nonreal complex numbers were similarly appended, in turn, to our number system in order to make certain (more complicated) functions total.
2. They adapted the function. In programming languages, in particular, true undefinedness is anathema, so such languages typically have ways of making functions total, via devices such as "integer division" (so that odd integers can be "divided by 2") as well as various ploys for accommodating "division by 0."

The (20th-century) inventors of *Computation Theory* insisted on a theory of functions on nonnegative integers (or some transparent encoding thereof). The price for such "pureness" is that we must allow functions to be undefined on some arguments.

[5] We are, of course, ignoring demons such as round-off error.

[6] The great mathematician Leopold Kronecker said, "God made the integers, all else is the work of man"; Kronecker was referring, of course, to the *positive* integers.

Thus the theory renders such functions as "division by 2" and "taking square roots" as being *nontotal*: both are defined only on subsets of the positive integers (the even integers and the perfect squares, respectively).

Three special classes of functions merit explicit mention. For each, we give both a down-to-earth name and a more scholarly Latinate one.

A function $F : S \to T$ is:

1. *one-to-one* (or *injective*) if for each $t \in T$, there is at most one $s \in S$ such that $F(s) = t$;

   Example:

   - "multiplication by 2" is injective: If I give you an even integer $2n$, you can always respond by giving me n.
   - "integer division by 2" is not injective—because performing the operation on arguments $2n$ and $2n+1$ yields the same answer (namely, n).

   An injective function F is called an *injection*.

   Importantly, each injection F has a *functional inverse*, which is commonly denoted F^{-1}. and which is defined as follows. For each $t \in T$:

   - If there is an $s \in S$ such that $F(s) = t$, then

   $$F^{-1}(t) \;=\; s$$

   - If there is no $s \in S$ such that $F(s) = t$, then $F^{-1}(t)$ is not defined.

   Because F is an *injection*, there is at most $s \in S$ such that $F(s) = t$. In other words, an element $t \in T$ can occur in the range of F only because of a single element $s \in S$ in the domain of F. This means that the preceding definition of F^{-1} is a valid definition (it is "well-defined") and that F^{-1} is a (partial) function $F^{-1} : T \to S$ whose domain is the range of F.
2. *onto* (or *surjective*) if for each $t \in T$, there is at least one $s \in S$ such that $F(s) = t$;

   Example:

   - Two surjective functions on the nonnegative integers:
     - "subtraction of 1" is surjective, because "addition of 1" is a total function.
     - "taking the square root" is surjective because the operation of squaring is a total function.
   - Two functions on the nonnegative integers that are *not* surjective:
     - "addition of 1" is not surjective, because, e.g., 0 is not "1 greater" than any nonnegative integer.
     - "squaring" is not surjective, because, e.g., 2 is not the square of any integer. (We prove this as Proposition 4.7.)

   A surjective function F is called a *surjection*.
3. *one-to-one, onto* (or *bijective*) if for each $t \in T$, there is precisely one $s \in S$ such that $F(s) = t$.

A bijective function F is called a *bijection*. When F is a bijection from S onto T, we often write $F : S \leftrightarrow T$.

There is a marvelous theorem that must be mentioned here, even though it is beyond the scope of the book.[7] The theorem says that, given sets S and T: *if* there is an injection F_1 that maps elements of S one-to-one to elements of T *and* there is an injection F_2 that maps elements of T one-to-one to elements of S, *then* there is a single bijection F such that

- F maps elements of S one-to-one to elements of T;
- F^{-1}, the *functional inverse* of F, maps elements of T one-to-one to elements of S.

Theorem 3.2 (The Schröder-Bernstein Theorem). *For any sets S and T, if there is an* injection $F^{(S\to T)} : S \to T$ *and an* injection $F^{(T\to S)} : T \to S$, *then there is a* bijection $F^{(S\leftrightarrow T)} : S \to T$.

While the operation of *composition*, as introduced in Section 3.2.1, is important for general binary relations, it is a daily staple with relations that are functions! Let us be given two functions on a set S

$$F : S \to S \quad \text{and} \quad G : S \to S$$

The composition of F and G, *in that order*, is the function

$$F \circ G : S \to S$$

defined as follows.

$$\text{For each } s \in S \quad F \circ G(s) \;=\; G(F(s)). \tag{3.4}$$

The unexpected change in the orders of writing F and G on the two sides of equation (3.4) results from the existence of two historical schools that both contributed to the formulation of this material. One school cleaved to the tradition of *abstract algebra*; they wanted all expressions to be written with all operators—including the composition operator $\circ$—in infix notation. Another school, which could be called *applicative algebraic*, wanted to view functions as "applying" to their arguments. Both notations have significant advantages in certain contexts, so both have survived. It is a good idea for neophyte readers to be prepared to encounter both notations—but they have to keep their eyes open regarding the relative orders of F and G.

Before progressing with new material, it is worth taking a moment to verify that the important operation of composition behaves the way one would want and expect it to—by preserving the type of function being composed.

Proposition 3.2 *Let us be given functions $F : S \to S$ and $G : S \to S$ on the set S, together with their composition $F \circ G$, as defined in (3.4).*

[7] The theorem can be found in texts such as [14].

(a) *$F \circ G$ is a function on S.*
(b) *If F and G are injections, then so also is $F \circ G$.*
(c) *If F and G are surjections, then so also is $F \circ G$.*
(d) *If F and G are bijections, then so also is $F \circ G$.*

Proof. We prove each of the four assertions by invoking the underlying definitions. (a) Because F and G are functions on S: For each $s \in S$, there is at most one $t_1 \in S$ such that $F(s) = t_1$ and at most one $t_2 \in S$ such that $G(s) = t_2$. Hence, we identify three possibilities.

if F is defined at $s \in S$	**then** $F(s) \in S$ is unique
and if G is defined at $F(s) \in S$	**then** $G(F(s)) = F \circ G(s) \in S$ is unique
if F is not defined at $s \in S$	**then** $F \circ G$ is not defined at $s \in S$
if F is defined at $s \in S$	**then** $F(s) \in S$ is unique
and if G is not defined at $F(s) \in S$	**then** $F \circ G$ not defined at $s \in S$

Hence, for each $s \in S$, there is at most one $t \in S$ such that $F \circ G(s) = t$; in other words, $F \circ G$ is a function on S.

(b) Focus on any $s \in S$. Because F is an injection on S, there exists at most one $t \in S$ such that $F(t) = s$. Because G is an injection on S, there exists at most one $u \in S$ such that $G(u) = t$. Thus, there exists at most one $u \in S$ such that $F \circ G(u) = s$. Hence, $F \circ G(u)$ is an injection on S.

(c) Focus on any $s \in S$. Because F is a surjection on S, there exists $t \in S$ such that $F(t) = s$. Because G is a surjection on S, there exists $u \in S$ such that $G(u) = t$. This means, however, that $F \circ G(u) = s$. Because $s \in S$ was arbitrary, it follows that $F \circ G(u)$ is a surjection on S.

(d) If each of F and G is a bijection on S, then each is an injection on S, and each is a surjection on S. Then Part (b) tells us that $F \circ G$ is an injection on S, and Part (c) tells us that $F \circ G$ is a surjection on S. Hence, $F \circ G$ is a bijection on S. □

3.3 Boolean Algebras

A *Boolean algebra* is a mathematical system **HERE

3.3.1 The Boolean Operations

3.3.2 The Axioms of Boolean Algebras

3.3.3 Two Special Boolean Algebras

3.3.3.1 The Algebra of Sets

3.3.3.2 Propositional Logic as an Algebra: The Propositional Calculus

A. The basic logical connectives. The Boolean set-related operations we discussed in Section 3.1.2 have important analogues within the context of Propositional logic. *logical* analogues of these operations for logical sentences and their logical *truth values*, TRUE and FALSE, often denoted 1 and 0, respectively:

- logical not ($\sim$) The operation not is the logical analogue of the set-theoretic operation of complementation. Because always writing "not" makes logical expressions long and cumbersome, we usually use the prefix-operator $\sim$ to denote not; i.e., we write
  $\sim P$ rather than the more cumbersome not P
  to denote the logical complementation of proposition P. Whichever notation we use, the defining properties of logical complementation are encapsulated in the following pair of equations
  $$[\sim \text{TRUE} = \text{FALSE}] \quad \text{and} \quad [\sim \text{FALSE} = \text{TRUE}].$$
- logical or ($\vee$) The operation or—which is also called *disjunction* or *logical sum*—is the logical analogue of the set-theoretic operation of union. For convenience and brevity, we usually use the infix-operator $\vee$ to denote or in expressions. Whichever notation we use, the defining properties of logical disjunction are encapsulated as follows.
  $$[[P \vee Q] = \text{TRUE}] \quad \text{if, and only if,} \quad [P = \text{TRUE}] \text{ or } [Q = \text{TRUE}] \text{ or both.}$$
  Note that, as with union, logical or is *inclusive:* The assertion
  $[P \vee Q]$ is TRUE
  is true when *both* P and Q are true, as well as when only one of them is. Because such inclusivity does not always capture one's intended meaning, another, *exclusive* version of disjunction also exists, as we see next.
- logical xor ($\oplus$) The operation *exclusive or*—which is also called xor—is a version of disjunction that does not allow both disjuncts to be true simultaneously. For convenience and brevity, we usually use the infix-operator $\oplus$ to denote xor in expressions. Whichever notation we use, the defining properties of exclusive or are encapsulated as follows.

$$[[P \oplus Q] = \text{TRUE}] \quad \text{if, and only if,} \quad [P = \text{TRUE}] \text{ or } [Q = \text{TRUE}] \textit{ but not } \text{both.}$$

To emphasize the distinction between $\vee$ and $\oplus$: The assertion

$[P \oplus Q]$ is TRUE

is *false* when *both* P and Q are true.

- logical and ($\wedge$) The operation and—which is also called *conjunction* or *logical product*—is the logical analogue of the set-theoretic operation of intersection. For convenience and brevity, we usually use the infix-operator $\wedge$ to denote and in expressions. Whichever notation we use, the defining properties of logical conjunction are encapsulated as follows.

$$[[P \wedge Q] = \text{TRUE}] \quad \text{if, and only if, } \textit{both} \quad [P = \text{TRUE}] \text{ and } [Q = \text{TRUE}]$$

- logical implication ($\Rightarrow$) The logical operation of implication, which is often called *conditional* and which we usually denote in expressions via the infix-operator $\Rightarrow$ differs from the other logical operations we have discussed in a way that the reader must always keep in mind. In contrast to not, or, and and, whose formal versions pretty much coincide with their informal versions, the formal version of implication, being formal, fixed, and precise, carries connotations that we do not always associate with the informal word "implies." The formal version of implication is defined as follows.

$$[[P \Rightarrow Q] = \text{TRUE}] \quad \text{if, and only if,} \quad [[\sim P] = \text{TRUE}] \text{ (inclusive) or } [Q = \text{TRUE}]$$

This definition means, in particular, that

  - If proposition P is false, then it implies *every* proposition.
  - If proposition Q is true, then it is implied by *every* proposition.

- logical equivalence ($\equiv$) The final logical operation in our toolbox is variously called *equivalence,* as in:

  Proposition P is (logically) equivalent to Proposition Q

  and *biconditional*. We usually denote the operation in expressions via the infix-operator $\equiv$. The operation is defined as follows.

$$[[P \equiv Q] = \text{TRUE}] \quad \text{if, and only if } [[P \Rightarrow Q] = \text{TRUE}] \quad \text{and} \quad [[Q \Rightarrow P] = \text{TRUE}]$$

We often use the term "connective" rather than "operation" to refer to what we have here called the logical operations of the Propositional Calclulus. This is because one often feels that logical propositions are static statements rather than active prescriptions for computations.

B. The logical connectives via truth tables If one is willing to view statements in the Propositional Calclulus as prescriptions for computations, then one can encapsulate the definitions of the basic logical connectives via functions that map logical expressions int the truth values true, or 1, and false, or 0. The following tables reproduce the definitions of Subsection A within this computational framework.

P	$\sim P$
0	1
1	0

P	Q	$P \vee Q$	$P \oplus Q$	$P \wedge Q$	$P \Rightarrow Q$	$P \equiv Q$
0	0	0	0	0	1	1
0	1	1	1	0	1	0
1	0	1	1	0	0	0
1	1	1	0	1	1	1

(3.5)

Note how the truth-value entries in the righthand truth table of (3.5) reinforce the lessons of Subsection A. This awareness is especially essential with the implication-related operations ($\Rightarrow$ and $\equiv$) that have informal counterparts in "real-life" reasoning. Think about whether your intuition agrees with the formal specifications of these operations.

C. The (Boolean) algebra of logical operations The operations of the Propositional Calclulus give rise to a Boolean algebra that manipulates (logical) propositions in much the way that the set-oriented Boolean algebra of Section 3.3.3.1 manipulates sets.

D. Logic via truth values As demonstrated in Subsection C, the Propositional Calculus is a Boolean Algebra when one interprets the logical connectives of Subsection A as operations on logical expressions that evaluate to either TRUE (or, 1) or FALSE (or, 0). But, the calculus forms a very special genre of Boolean Algebra, namely, a *free Boolean Algebra*.

The meaning of the qualifier *free* is easy to state but not so easy to understand. Here is a very informal try.

Any sort of algebra is defined as a set of objects accompanied by a set of basic operations, wherein the operations obey certain "self-evident" axioms.[8] In rough, but evocative, terms, the algebra is *free* if its operations do not relate to one another in any way that is not mandated by the axioms.

An easy illustration of this decidedly *not easy* concept is the following. There is a (quite important) genre of algebra called a *semi-group*. A semi-group $\mathscr{S}$ is specified via a set (of objects) S, together with a binary operation on S, which we denote $\oplus$. For the (algebraic) structure $\mathscr{S} = \langle S, \oplus \rangle$ to be a semi-group, it must obey the following two axioms.

1. $\mathscr{S}$ must be *closed* under operation $\oplus$; i.e., for all $s_1, s_2 \in S$, we must have
$$(s_1 \oplus s_2) \in S.$$
2. The operation $\oplus$ must obey the *associative law;* i.e., for all $s_1, s_2, s_3 \in S$, we must have
$$(s_1 \oplus s_2) \oplus s_3 \;=\; s_1 \oplus (s_2 \oplus s_3)$$

Now, it is easy to verify—we shall discuss this in Chapter 5—that the (algebraic) structure formed by the integers coupled with the operation of addition forms a semi-group (which the notation of Chapter 5 would denote $\langle \mathbb{Z}, + \rangle$). But this semi-group is *not* free, because integer addition obeys laws beyond just closure and associativity, namely:

1. Addition of integers is *commutative*.
2. Integer addition has an *identity*, namely 0.

[8] Indeed, that is what the word "axiom" means.

3. Every integer $z \in \mathbb{Z}$ has an *additive inverse* (namely, $-z$).

So, it really is meaningful that the Boolean Algebra built upon the Porpositional calculus *is* free.

For us, the impact of the "freeness" of the Propositional algebra, *qua* Boolean Algebra, is manifest in the following *meta-theorem*[9] which is discussed and proved in [75].

Theorem 3.3. *A propositional expression $E(P,Q,\ldots,R)$ is a theorem of Propositional logic if, and only if, it to* TRUE *under all truth assignments to the propositions $P,Q,\ldots,R$.*

Proving Theorem 3.3 is beyond the scope of this book, but we now present a few illustrative instantiations, annotated to suggest their "real-world" messages. Each example is accompanied by a "real-life" interpretation and a verifying truth table.

The law of double negation

This is a formal analogue of the homely adage that "a double negative is a positive."

Proposition 3.3 *For any proposition P,*

$$P \equiv \sim[\sim P]$$

Proof via truth table.

P	$\sim P$	$\sim[\sim P]$
0	1	0
1	0	1

(3.6)

Note that columns 1 and 3 of truth table (3.6) are identical. By Theorem 3.3, this fact verifies Proposition 3.3, the law of double negation. □

The law of contraposition

This is a very exciting example! Let us immediately convert this to a mathematical statement about the Boolean Algebra of Propositional logic and then prove the statement. We shall then contemplate the implications of this law for *logic* rather than *mathematics*.

Proposition 3.4 *For any propositions P and Q,*

$$[[P \Rightarrow Q] \equiv [\sim Q \Rightarrow \sim P]]$$

Proof via truth table.

[9] A *theorem* exposes some truth within the mathematical structure being discussed. A *meta-theorem* exposes some truth about the way the discussion can proceed.

P	$\sim P$	Q	$\sim Q$	$P \Rightarrow Q$	$\sim Q \Rightarrow \sim P$
0	1	0	1	1	1
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	0	1	1

(3.7)

Note that columns 5 and 6 of truth table (3.7) are identical. By Theorem 3.3, this fact verifies Proposition 3.4, the law of contraposition. □

Now let us reconsider the whole concept of contraposition, including this law, in the light of *logic and reasoning*.

asserts that the assertion

"Proposition Q implies Proposition Q"

is *logically equivalent* to the assertion

"the negation of Proposition Q implies the negation of Proposition P".

Think about this! In any system of reasoning in which a given proposition is either true or false

De Morgan's Laws:

Proposition 3.5 *For any propositions P and Q:*

- $[P \wedge Q] \equiv \sim [[\sim P] \vee [\sim Q]]$
- $[P \vee Q] \equiv \sim [[\sim P] \wedge [\sim Q]]$

Proof via truth table.

P	$\sim P$	Q	$\sim Q$	$[P \wedge Q]$	$[\sim P] \vee [\sim Q]$	$[P \vee Q]$	$[\sim P] \wedge [\sim Q]$
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	1
1	0	0	1	0	1	1	1
1	0	1	0	1	0	1	0

(3.8)

Note that columns 5 and 6 of truth table (3.8) are mutually complementary, as are columns 7 and 8. If we negate (or, complement) the entries of columns 6 and 8, then we can invoke Theorem 3.3 to verify proposition 3.5, which encapsulates De Morgan's laws for Propositional logic. □

The distributive laws for Propositional logic

In numerical arithmetic, multiplication distributes over addition, but not conversely, so we have a single distributive law for arithmetic (see Section 5.1.2). In contrast, each of logical multiplication and logical addition distributes over the other, so we have two distributive laws for Propositional logic.

- $P \vee [Q \wedge R] \equiv [P \vee Q] \wedge R$
- $P \wedge [Q \vee R] \equiv [P \wedge Q] \vee R$

P	Q	R	$[P\vee Q]$	$[P\wedge Q]$	$[Q\wedge R]$	$[Q\vee R]$	$P\vee[Q\wedge R]$	$[P\vee Q]\wedge[P\vee R]$	$P\wedge[Q\vee R]$	$[P\wedge Q]\vee[P\wedge R]$
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0
0	1	0	1	0	0	1	0	0	0	0
0	1	1	1	0	1	1	1	1	0	0
1	0	0	1	0	0	0	1	1	0	0
1	0	1	1	0	1	0	1	1	1	1
1	1	0	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

(3.9)

Note that columns 8 and 9 of truth table (3.9) are identical, as are columns 10 and 11. By Theorem 3.3 this fact verifies the distributive laws for Propositional logic.

3.3.4 Connecting Mathematical Logic with Logical Reasoning

3.3.4.1 A formal notion of *implication*, and its implications

In everyday discourse, we all employ an intuitive notion of *implication*. When we make the assertion

Proposition A *implies* Proposition B

what we usually have in mind is

If Proposition A is true, then Proposition B istrue.

But this, or any, homespun meaning of the word/concept "implies" raises many questions.

- What if Proposition A is *not* true? Are there any inferences we can draw?
- Is there any relation between the assertion
  Proposition A *implies* Proposition B
  and its *converse* assertion
  Proposition B *implies* Proposition A?
- If we know that Proposition B is true, shouldn't it be "implied by every other proposition—perhaps even a *false* one?

This section is devoted to discussing the *formal* notion of implication that was adopted by mathematical philosophers in the 19th century. The *advantage* of having such a formal notion is that it will answer all questions of the sort we have just posed. The *disadvantage* of having such a formal notion is that the way in which the notion answers some of our questions may be rather counter to one's untutored intuition.

Chapter 4
NUMBERS AND NUMERALS

4.1 Introduction

Many, perhaps most, of us take for granted the brilliant notations that have been developed for the myriad arithmetic constructs that we use daily. Our mathematical ancestors have bequeathed us notations that are not only perspicuous but also convenient for computing and for discovering and verifying new mathematical truths. Several chapters in this text are devoted to sharing the elements of this legacy with the reader. This chapter focuses on elementary concepts and techniques relating to the most familiar objects of mathematical discourse, numbers and the numerals that we use to manipulate them. We extend this chapter's introduction to our number system with more advanced topics concerning numbers in Chapter 7.

Every reader will be familiar with the notion of *number* and with the familiar strings, called *numerals*, that name numbers.

> Numbers and numerals embody what is certainly the most familiar instance of a very important dichotomy that pervades our intellectual lives: the distinction between objects and their names:
> *Numbers are intangible, abstract objects.*
> *Numerals are the names we use to refer to and manipulate numbers.*
> This is a critically important distinction! You can "touch" a numeral: break it into pieces, combine two (or more) numerals via a large range of operations. When numerals are *operational*, then you can compute with them. Numbers are intangible abstractions, or, conceptualizations: you reason with numbers.

Historically, we have employed a broad range of mechanisms for naming numbers. *Nicknames for "popular" numbers*. At one extreme, we have endowed certain numbers that we are "fond of" with names that do not even hint at the "meaning of" the named number. To cite just a few examples, we talk about

- π: the ratio of the circumference of a circle to its diameter
- e: the base of so-called natural logarithms

- ϕ: the *golden ratio* (one of several word-names for ϕ) that can be observed in nature, e.g., in the leaf patterns of plants such as pineapples and cauliflower
- Avogadro's number: a fundamental quantity discussed in chemistry and physics. We include this number-name to indicate that not all special numbers' nicknames are single letters.

Nickname-based numerals give no information about the named number: they do not help anyone (except possibly the *cognoscenti:* the "in-crowd") *identify* the named number, and they do not help anyone manipulate—e.g., compute with—the named number. These names are valuable only for *cultural* purposes, not mathematical ones.

We want to be totally clear about our intended message. It is the *names* of the numbers we have mentioned that convey no operational information. Regarding the numbers themselves, each is attached to valuable science and/or mathematics! We shall expose some of this mathematics as we discuss e in the current chapter, as we revisit π and e in Chapter 6, and as we revisit ϕ in Chapter 8.

Alphabet-based systems. Several cultures have developed systems for naming arbitrary integers by using their alphabets in some manner. One such system that is still visible in European cultures within constrained contexts comprises *Roman numerals*. One stills encounters these, e.g., as the hour markers on the faces of "classical" clocks and as timestamps on the cornerstones of official buildings. Roman numerals are formed from a constrained set of letters from the Latin alphabet:

Letter	*Numerical value*
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

The formation rules for Roman numerals of length exceeding 2 are a bit complicated, but *roughly*, a letter to the right of a higher valued letter augments the value of the numeral (e.g., DCL $= 650$, XVI $= 16$), while a letter to the left of a higher valued letter lowers the value (e.g., MCM $= 1900$, XLIV $= 44$).

Yet another way to craft numerals from letters is observable in the based on, e.g., the Hebrew alphabet; this system assimilates ideas used by the ancient Egyptians, Phoenicians, and Greeks. Hebrew assigns the following respective values to the 22 letters of its alphabet.

$$1,2,3,4,5,6,7,8,9,10,20,30,40,50,60,70,80,90,100,200,300,400$$

Numerals are then formed as strings of single occurrences of letters, by accumulating the letters' numerical values. Numbers that are too large to be named using strings of single letter-instances often allow repeated letter instances or incorporate auxiliary words, in a mixed-mode manner similar to our writing the number 5000 as "5 thousand".

Alphabet-based systems for creating numerals are more useful than nickname-based systems: they *do* allow anyone to *identify* any named number. Indeed, one can (algorithmically) convert any Roman numeral or any Hebrew numeral to a decimal numeral for the same number. However, any reader who is familiar with alphabet-based systems will recognize a major drawback of such systems: It is *exceedingly difficult* to do any but the most trivial arithmetic using such systems' numerals. Two simple examples using Roman numerals will make our case:

- Square CC. This is, of course, trivial using our familiar decimal numerals: an elementary school student can compute $200 \times 200 = 40,000$. But even in an early course on programming, one would not assign the general "multiply numbers using Roman numerals" problem as a first assignment.
- Substract MCMXCVIII from MMII. We all know, of course, that the answer is IV, but determining this is usually done by converting to decimal numerals $(2002 - 1998)$.

Positional number systems. In our daily commerce, we typically deal with numerals that are formed within a *base-b positional number system,* i.e., by strings of *digits* from a set of the form $\{0, 1, 2, \ldots, \overline{b-1}\}$, often embellished with other symbols, such as a *radix point*,[1] and sometimes a leading "$+$" or "$-$" to indicate, respectively, the denoted number's positivity or negativity.

> We employ the notation "$\overline{b-1}$" to remind ourselves that in this context "$b-1$" is a digit , not a string; for instance, when $b = 10$ (the common *decimal* base), $\overline{b-1}$ is the digit 9.

We discuss positional number systems in detail in Section 4.7. For now, we settle for a few examples:

- Most of our daily work employs the base-10 (*decimal*) system, whose digits comprise the set $\{0, 1, 2, 3, 4, 5., 6, 7, 8, 9\}$; the system's radix point is usually called a *decimal point*.
- Because electrical and electronic circuitry are (for the most part) built using *bistable* devices—e.g., switches that are either *on* or *off*—the system most often employed when dealing with such circuitry and its end products (say, computers) is the base-2 (*binary*) system, whose digits—usually called *bits*—comprise the set $\{0, 1\}$. The term bit is the contraction of binary digit.
- Because of its small repertoire of digits, the binary system's numerals are quite long—roughly 3 times longer than decimal numerals. For instance,

$$32,768 \text{ base } 10 \;=\; 1,000,000,000,000,000 \text{ base } 2$$

  In order to make these numerals easier for humans to deal with, small sequences of bits are often aggregated to form larger number bases—but still powers of 2. Two aggregations have been particularly popular:

[1] The use of a period as the radix point is a US convention; in much of Europe, a comma usually denotes the radix point.

- By aggregating length-3 sequences of bits, one converts binary numerals to base-8 (*octal*) numerals; the octal digits comprise the set $\{0,1,2,3,4,5,6,7\}$.
- By aggregating length-4 sequences of bits, one converts binary numerals to base-16 (*hexadecimal*) numerals; the hexadecimal digits comprise the set

$$\{0,1,2,3,4,5,6,7,8,9,\overline{10},\overline{11},\overline{12},\overline{13},\overline{14},\overline{15}\}.$$

  Note: We have written the hexadecimal digits in decimal, to make them easy to read, but we have placed overlines above the 2-digit (decimal) numerals "10", "11", "12", "13", "14", and "15" as a reminder that each represents a single hexadecimal digit, not a 2-digit numeral.

4.2 A Brief Biography of Our Number System

We begin our study of numbers and numerals with a short taxonomy of our number system. Although we assume that the reader is familiar with the most common classes of numbers, we do spend some time highlighting important features of each class, partly, at least, in the hope of heightening the reader's interest in this most basic object of mathematical discourse.

We present the four most common classes of numbers in what is almost certainly the chronological order of their discovery/invention.

> Did humans *invent* these classes of numbers to fill specific needs, or did we just *discover* their pre-existing selves as needs prompted us to search for them? The great German mathematician Leopold Kronecker, as cited in [10] (page 477), shared his viewpoint on this question: "God made the integers; all else is the work of man."

A pleasing narrative can be fabricated to account for our multi-class system of numbers. In the beginning, the story goes, we needed to count things (sheep, bottles of oil, weapons, ...), and the positive *integers* were discovered to serve this need. As accounting practices matured, we needed to augment this class with the number zero (0), to allow merchant A to keep track of the inventory remaining after the last flask of wine is sold, and with the negative integers, to allow the merchant's banker to record A's credit balance after taking a loan. The class of integers was now complete, even though a variety of special classes of integers that warranted special attention for reasons ranging from the religious to the intellectual remained to be discovered. As society matured, humans began to share materials that had to be subdivided—cloth, grain, etc.—rather than partitioned in discrete units. We needed to invent the *rational numbers* to deal with such materials. Happily for the mathematically inclined, the rational numbers could be developed in a way that allowed one to view an integer as a special type of rational. This meant that our ancestors could build upon the systems they had developed to deal with integers, rather than scrapping those systems and starting anew.

This quest for *extendible* frameworks rather than isolated unrelated frameworks is a hallmark of mathematical thinking.

We now enter the realm of "semi-recorded" history in the West: the Babylonians, the Egyptians, the Greeks, and others. "Practical" mathematics was invented—and reinvented—to accommodate pursuits as varied as astronomy, commerce, navigation and architecture. Our mathematical stalwarts, the integers and the rationals, did not seem to be able to deal with all of the measurements that we wanted to make, calculations that we wanted to do, structures that we wanted to design. So, we approximated and "fudged" and got pretty much where we wanted to get. The standard story at this point (at least in the West) is that the ancient Greeks began to try to systematize mathematical knowledge and practice. The Greek mathematician and geometer Euclid and members of his school verified—via one of the first *proofs* in recorded history—the uncomfortable fact that the lengths of portions of eminently buildable structures were not "measurable," by which they meant *not rational*. The poster child for this phenomenon was the hypotenuse of the isosceles right triangle T with unit-length legs. Thanks to the well-known theorem of the Greek mathematician Pythagoras, even schoolchildren nowadays know that the length of this eminently buildable line is $\sqrt{2}$. What Euclid discovered—and what is still likely not known by *all* schoolchildren—is: *There is—provably!—no way to find integers p and q whose ratio is $\sqrt{2}$, the length of T's hypotenuse.*[2]

Digression

It is worthwhile digressing here with some supplementary material so that you do not have to take our word for the mathematical facts underlying our story.

Even though the so-called *Pythagorean Theorem* is widely known, at least informally, it is worthwhile to provide a formal statement of this seminal result. Quite aside from reviewing the Theorem's important content, this statement will provide the reader one more opportunity to ponder the "music" of mathematical discourse.

Let us be given a triangle T with vertices A, B, and C. Use the lefthand grey triangle in Fig. 4.1 as a model. Say that the angle at vertex A of T is a *right angle*, i.e., that its measure is 90° (90 degrees or, equivalently, $\pi/2$ radians). In this case, we say that T is a *right triangle*. Because T is a right triangle, the line from vertex A to vertex B and the line from vertex A to vertex C are called the *sides* (or the *legs*) of T, while the line from vertex B to vertex C is called the *hypotenuse* of T. Triangle T is *isosceles* precisely when its two sides have the same length. The grey triangle in Fig. 4.1 is an isosceles right triangle.

Theorem 4.1 (The Pythagorean Theorem). *Let T be a right triangle whose two sides have respective lengths s_1 and s_2, and whose hypotenuse has length h. Then*

$$h^2 \;=\; s_1^2 + s_2^2.$$

Consequently, when T is an isosceles *right triangle, then $h^2 = 2s_1^2$.*

[2] We present a version of Euclid's proof in Proposition 4.7.

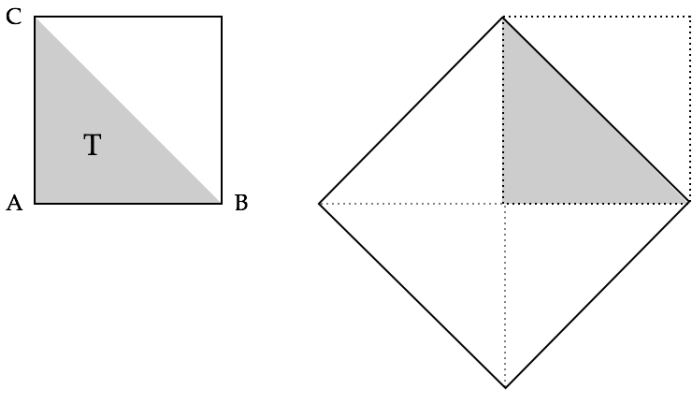


Fig. 4.1 A pictorial schematic proof of The Pythagorian Theorem for the special case of an isosceles right triangle.

The special case of the Pythagorean Theorem, which deals with isosceles right triangles admits the very perspicuous proof presented pictorially in Fig. 4.1. Let us review this proof.

We begin with the unit-side square S on the left of the figure. We partition S by its diagonal into two unit-side isosceles right triangles, one grey and one white. In this construction the diagonal of S is the (common) hypotenuse of the two triangles. On the righthand side of Fig. 4.1, we use our partitioned version of S to construct a new, bigger square, call it $\widehat{S}$, whose side-length is the hypotenuse-length of the grey triangle. The dotted lines in the figure tell us how big $\widehat{S}$ is (measured in area).

- Square S is unit-sided, hence has unit area.
- The grey triangle is (geometrically) half of S, hence has area $1/2$.
- Square $\widehat{S}$ is built from four copies of the grey triangle, hence has area $4 \cdot 1/2 = 2$.

Because the hypotenuse of the grey triangle is a side of an area-2 square, we have just proved the following special case of the Pythagorean Theorem.

The hypotenuse of the unit-side isosceles right triangle has length $\sqrt{2}$.

End of digression

As part of the same movement toward formal mathematics, the Sicilian-based Greek mathematician and polymath Archimedes was systematically observing that squares are better approximations to circles than triangles are; regular pentagons are better than squares; regular hexagons are better than pentagons; and so on. Fig. 4.2 illustrates this evidence. In fact, observed Archimedes, as the number of sides, n, in a regular polygon grows without bound (or, as we might say today, tends to infinity), each increase in n brings a regular polygon closer to being a circle.

In order to pursue their respective observations to their completion, both Euclid and Archimedes would have to leave the world of the rationals and enter the world of the *real numbers* (so named by the French mathematician-philosopher

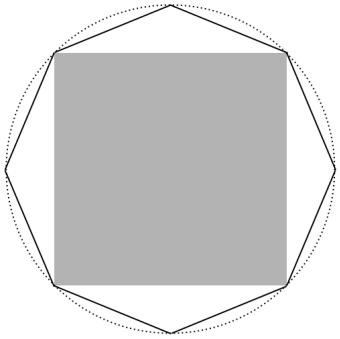


Fig. 4.2 *Octagons approximate circles much better than squares do.*

René Descartes). It would take roughly two-thousand years from the time of Euclid and Archimedes before the real numbers were *formally* introduced to the world, by mathematical luminaries such as the early-19th-century French mathematician-scientist Augustin-Louis Cauchy and the late-19th-century German mathematician Richard Dedekind. It turned out to be much easier to recognize instances of non-rational real numbers than to formally delimit the entire family of such numbers. Once again, happily, one could develop the real numbers in a way that allowed one to view a rational number as a special type of real number.

During the millennia between the discoveries of Euclid, Archimedes, and their friends and the full development of the real numbers, mathematics was enriched repeatedly by the discovery of new conceptual structures. One of these—polynomials and their roots—ultimately led to the final major subsystem of our number system. In Chapter 3.2.4, we discussed the important notion of *function*. Polynomials are a practically important class of functions that are delimited by the operations needed to compute them. Specifically, an n-argument polynomial function—typically just called a polynomial—is a function $P(x_1, x_2, \ldots, x_n)$ whose values can be calculated using just the basic operations of arithmetic: addition/subtraction and multiplication/division.[3] A *root* of a polynomial (function) $P(x_1, x_2, \ldots, x_n)$ is an argument $\langle r_1, r_2, \ldots, r_n \rangle$ that causes P to *vanish*, meaning that $P(r_1, r_2, \ldots, r_n) = 0$. Here are a few illustrative, examples of *univariate* (i.e., *single-variable*) polynomials:

[3] We pair the operations in this way because addition and subtraction are *mutually inverse operations*, as are multiplication and division. This means that one can undo an operation (say, an addition) by performing its inverse operation (in this case, a subtraction). But *be careful:* one cannot undo multiplication by 0.

	Polynomial $P(x)$	Root(s)
1	$x+1$	$x=-1$
2	$x-1$	$x=1$
3	$x^2+2x+1=(x+1)^2$	$x=-1$
4	$x^2-2x+1=(x-1)^2$	$x=1$
5	x^2-1	$x=1$ and $x=-1$
6	x^2+1	*no real root*
7	x^2-2	$x=\sqrt{2}$ and $x=-\sqrt{2}$
8	x^2+2	*no real root*

There are lessons, both major and minor, to be gleaned from this table.

- Entries 3 and 4 in the table illustrate that even simple polynomials can often be written in several different ways. These entries illustrate also that roots can occur *with multiplicity:* one can view the value $x=-1$ as causing the polynomial $x^2+2x+1=(x+1)\cdot(x+1)$ to vanish in two ways—(1) by setting the lefthand factor $x+1$ to 0 and (2) by setting the righthand factor $x+1$ to 0.
- Entries 5 and 7 in the table illustrate that polynomials can have multiple distinct roots.
- Perhaps most importantly, entries 6 and 8 in the table provide explicit, simple polynomials—whose expressions involve only positive integers—that fail to have any real roots!
  The fact that the indicated polynomials have no real roots is immediate, because the square of a real number can never be negative. Hence, for instance, there is no positive integer c such that $c^2=-1$, or, equivalently, $c^2+1=0$.

For both applied and purely intellectual reasons, there has always been considerable interest in developing techniques for finding the roots of polynomials. Indeed, much seminal mathematics was developed in the quest for such techniques;[4] we study the topic at length in Chapter 5.3.

Closely related to this interest in a polynomial's roots was the considerable discomfort within the mathematical and technical world at the fact that the then-current number system—built upon the integers, the rationals, and the reals—was inadequate to the important task of providing roots for every polynomial. The reaction to this deficiency was similar in kind to all earlier recognized deficiencies: a way was found to expand the number system! Centuries would pass before mathematics developed adequately to find the needed expansion. Once discovered, the expansion was based in the conception, in the 16th century, of a new *imaginary* number, so designated by Descartes.[5] This new number was named i (for "imaginary") and was defined to be a root of the polynomial $P_{-1}(x)=x^2+1$. The number i was evocatively often defined via the equation, $i=\sqrt{-1}$. By keeping our extended arithmetic

[4] A giant in the development and transmission of this work was the 9th-century mathematician and astronomer whose name is traditionally anglicized as Muhammad ibn Musa al-Khwarizmi . He is usually credited with introducing Hindu-Arabic numerals (ancestors of the ones we use to this day) and the elements of algebra into Europe. And, he is the eponym of our word "*algorithm*".

[5] The term "imaginary" is reputed to be a derogation of these numbers that flouted tradition.

consistent with our former arithmetic, $-i$ also became a root of $P_{-1}(x)$. When the imaginary number i was added to the real number system, and the combination was blended via the rules of arithmetic, the *complex number system* was born.

Thankfully, the imaginary number i was the only totally new concept that was needed to mend the observed deficiency in the real numbers. In formal terms, the complex numbers were shown to be *algebraically complete* in the sense expressed in the landmark *Fundamental Theorem of Algebra*:

Theorem 4.2 (The Fundamental Theorem of Algebra). *Every polynomial of degree n with complex coefficients has n roots over the complex numbers.*

The proof of the Theorem is beyond the scope of our introductory text, but the result is a notable milestone in our mathematical/technical culture.

Our historical tour is now complete, so we can—finally—begin to get acquainted with our number system and the operations that bring it to life in applications.

4.3 Integers: The "Whole" Numbers

The most basic class of numbers are the *integers*, which are also referred to as the *whole numbers* or the *counting numbers*. As suggested in our "biography" of our number system (Section 4.2), integers are certainly the numbers that our prehistoric ancestors employed in the earliest days of our species. This section is devoted to exploring some of the basic properties of the class of integers. The details we provide in Section 7.2 regarding the building blocks of the integers, the *prime numbers*, will prepare the reader for myriad applications of the integers, including important security-related applications. Our introduction to *pairing functions*, in Section 7.3, will open the door toward myriad applications of the integers that build on the ordering properties of the integers, coupled with tools for encoding highly structured data as integers. We begin to explore the basics of the integers.

4.3.1 The Basics of the Integers: The Number Line

We survey a number of the most important properties of the following three sets, which collectively comprise "the integers."

- The set $\mathbb{Z}$ comprises *all integers*—the positive and negative integers and zero (0).
- The set $\mathbb{N}$ comprises the *nonnegative integers*—the positive integers and zero (0).
- The set $\mathbb{N}^+$ comprises the *positive integers*.

There is no universal default when one refers to "the integers" with no qualifying adjective; therefore, we shall always be careful to indicate which set we are discussing at any moment—often by supplying the set-name: $\mathbb{Z}$ or $\mathbb{N}$ or $\mathbb{N}^+$.

4.3.1.1 Natural orderings of the integers

Several essential properties of the sets $\mathbb{Z}$, $\mathbb{N}$, and $\mathbb{N}^+$ are consequences of the integers' behaviors under their natural order relations:

- the two *less-than* relations:
  - the *strict* relation ($<$). We articulate "$a < b$" as "a is (strictly) less than b" or "a is (strictly) smaller than b."
  - the *nonstrict* relation ($\leq$). We articulate "$a \leq b$" as "a is less than or equal to b" or "a is no larger than b."
- their *converses,* the *greater-than* relations:
  - the *strict* relation ($>$). We articulate "$a > b$" as "a is (strictly) larger than b."
  - the *nonstrict* relation ($\geq$). We articulate "$a \geq b$" as "a is greater than or equal to b" or "a is no smaller than b."

One sometimes encounters *emphatic* versions of the strict relations: $a << b$ and $a >> b$ indicate that a is, respectively, *much smaller than* or *much larger than* b.

The reader will note throughout the text that order within a number system is among one's biggest friends when reasoning about the numbers within the system.

4.3.1.2 The order-related laws of the integers

A. Total order and the Trichotomy Laws

The sets $\mathbb{Z}$, $\mathbb{N}$, and $\mathbb{N}^+$ are *totally ordered*, also termed *linearly ordered* .

These facts are embodied in the *Trichotomy Laws for integers*.

The Trichotomy Laws for integers.

(i) *For each integer $a \in \mathbb{Z}$, precisely one of the following is true.*
a equals 0: $(a = 0)$ a is *positive*: $(a > 0)$ a is *negative*: $(a < 0)$

(ii) *For each integer $a \in \mathbb{N}$, precisely one of the following is true.*
a equals 0: $(a = 0)$ a is *positive*: $(a > 0)$

(iii) *For each integer $a \in \mathbb{N}^+$:*
a is *positive*: $(a > 0)$

Consequently:

(i') $\mathbb{Z}$ can be visualized via the (2-way infinite) number line:

$$\ldots, -3,\ -2,\ -1,\ 0,\ 1,\ 2, \ldots$$

(ii') $\mathbb{N}$ can be visualized via the (1-way infinite) number line:

$$0,\ 1,\ 2,\ 3,\ldots$$

(iii') $\mathbb{N}^+$ can be visualized via the (1-way infinite) number line:

$$1,\ 2,\ 3,\ldots$$

The Trichotomy Laws can be expressed using arbitrary pairs of integers, rather than insisting that one of the integers be zero. For the set $\mathbb{Z}$, for instance, this version of the Laws takes the following form:

For any integers $a, b \in \mathbb{Z}$, precisely one of the following is true.

a equals b: $(a = b)$ a is less than b: $(a < b)$ a is greater than b: $(a > b)$

B. Well-ordering

The sets $\mathbb{N}$ and $\mathbb{N}^+$ are *well-ordered*.

The Well-ordering law for nonnegative and positive integers.
Every subset of $\mathbb{N}$ or of $\mathbb{N}^+$ has a smallest element (under the ordering $<$).

C. Discreteness

The set $\mathbb{Z}$ is *discrete*.

The discreteness of the integers.
For every integer $a \in \mathbb{Z}$, there is no integer between a and $a+1$; i.e., there is no $b \in \mathbb{Z}$ such that $a < b < a+1$.

D. The law of "between-ness"

The "between-ness" law for the set $\mathbb{Z}$:

For any integers $a, b \in \mathbb{Z}$, there are finitely many $c \in \mathbb{Z}$ such that $a < c < b$.

Any such c lies *between* a and b along the number line, whence the name of the law.

E. The cancellation laws

There are two *cancellation laws* for the set $\mathbb{Z}$, one for the operation of addition and one for the operation of multiplication.

The cancellation law for addition.
For any integers $a, b, c \in \mathbb{Z}$, if $a + c = b + c$, then $a = b$.

The cancellation law for multiplication.
For any integers $a, b \in \mathbb{Z}$ and $c \in \mathbb{Z} \setminus \{0\}$, if $a \cdot c = b \cdot c$, then $a = b$.

The cancellation laws provide limited versions of the algebraic notion of mutually inverse arithmetic operations (Section 5.1.1.2).

4.3.2 Divisibility: Quotients, Remainders, Divisors

This section is devoted to studying the fundamental relation of *divisibility* between two integers. Let $m, n \in \mathbb{N}$ be nonnegative integers. We use any of the following locutions to assert the existence of a positive integer q such that $n = q \cdot m$.

- *m divides n*
- *m is a divisor of n*
- *n is divisible by m*
- $m \mid n$.

This section is devoted to studying the possible *divisibility* relations between integers m and n. We begin by noting some general facts.

- *Every integer m divides* 0.
  This is because of the universal equations $m \cdot 0 = 0 \cdot m = 0$. The same equations verify that 0 *does not divide any integer.*
- 1 *divides every integer.*
  This is because of the universal equation $1 \cdot m = m$.
- *Every nonzero integer divides itself.*
  This is because of the universal equation $m \cdot 1 = m$.

Some nonzero integers have many distinct divisors, while some have very few. Consider, for illustration, the first twelve positive integers.

Number	Divisors
1	$\{1\}$
2	$\{1,2\}$
3	$\{1,3\}$
4	$\{1,2,4\}$
5	$\{1,5\}$
6	$\{1,2,3\}$
7	$\{1,7\}$
8	$\{1,2,4,8\}$
9	$\{1,3,9\}$
10	$\{1,2,5,10\}$
11	$\{1,11\}$
12	$\{1,2,3,4,6,12\}$

All nonzero integers (but 1) have at least two divisors, 1 and themselves. The "sparsely divisible" integers that have only these two divisors are called *primes* or

prime integers or *prime numbers*. We study these "building blocks of the positive integers" in more detail in Section 7.2. While there is, indeed, much of interest to discuss about prime numbers, the *composite*—or, nonprime—integers are also quite interesting, particularly, when we focus of *pairs* of integers. The next section looks at the defining property of composite integers, namely, *divisibility*.

In order to better understand the fundamental concept of divisibility, we must broaden our perspective somewhat and consider the notion of *Euclidean division*, i.e., division with remainders. The notion of "perfect" divisibility that we have been discussing thus far is the special case in which the remainder is 0. The next subsection studies Euclidean division; the remainder of this section investigates the ramifications of "perfect" divisibility.

4.3.2.1 Euclidian division

Divisibility is not always perfect: Given a pair of integers, neither needs be an integer multiple of the other. As we learned in elementary school, if an integer $m > 0$ does not "evenly" divide an integer $n > m$, then we are left with a "remainder" when we attempt to divide n by m. it Euclidean *division*—so named for the Greek mathematician Euclid, whose writings introduced the process in the West—is the process of producing, given integers $m > 0$ and $n > m$, an integer *quotient* q and an integer *remainder* r (where $0 \leq r < m$) such that $n = q \cdot m + r$. The process of Euclidean division always succeeds, in the following strong sense.

Proposition 4.1 (The Division Theorem) *Given any integers n and $m > 0$, there exists a unique pair of integers q and r, with $0 \leq r < m$ such that*

$$n \;=\; q \cdot m + r. \tag{4.1}$$

Proof. We first prove that a result-pair $\langle q, r \rangle$, as described in the Proposition, exists for each argument-pair $\langle m, n \rangle$. Then we prove that the result-pair is unique.

There exists at least one argument-pair. Given any argument-pair $\langle m, n \rangle$, let $N_{m,n}$ be the set of all integers of the form $(n - a \cdot m)$ for some integer a. Symbolically,

$$N_{m,n} \stackrel{\text{def}}{=} \{(n - a \cdot m) \;\mid\; \text{both } a,\ (n - a \cdot m) \in \mathbb{N}\}$$

Each such set $N_{m,n}$ is a nonempty set of nonnegative integers: the nonemptiness follows because $n \in N_{m,n}$ (via the case $a = 0$). By the Well-ordering law, $N_{m,n}$ contains a (perforce, unique) *smallest* element. Let us denote this smallest element by r, and let us denote by a_r the value of a that yields r; i.e.,

$$n - a_r \cdot m \;=\; r.$$

We complete this section of the proof, we need show that $r < m$. Say, for contradiction, that $r = m + r'$ for some $r' \in \mathbb{N}^+$. We then find

$$n-(a_r+1)\cdot m \;=\; r-m \;=r'$$

to be an element of $N_{m,n}$ that is strictly smaller than r. This contradiction completes the first part of the proof.

2. There exists at most one argument-pair. Turn now to the issue of uniqueness. Say, for the sake of contradiction, that there exists an argument-pair $\langle m,n\rangle$, for which there exist distinct result-pairs $\langle q_1,r_1\rangle$ and $\langle q_2,r_2\rangle$. We therefore have

$$n \;=\; q_1\cdot m+r_1 \;=\; q_2\cdot m+r_2, \tag{4.2}$$

where both r_1 and r_2 satisfy the inequalities $0\leq r_1,r_2<m$. We consider two cases.

- **Case 1.** Assume first that $r_2=r_1$. In this case, the equations (4.2) tell us that

  $$q_1\cdot m \;=\; q_2\cdot m.$$

  The cancellation law for multiplication then tells us that $q_1=q_2$. Therefore, the allegedly distinct result-pairs are, in fact, identical.
- **Case 2.** If $r_2\neq r_1$, then say, with no loss of generality, that $r_2>r_1$. In this case, the equations (4.2) tell us that

  $$(q_1-q_2)m \;=\; r_2-r_1.$$

  Because the righthand quantity is positive, so also must be the lefthand quantity; i.e., $q_1>q_2$ because $r_2>r_1$.
  On the one hand, the lefthand quantity, $(q_1-q_2)m$, is no smaller than m. This is because q_1 and $q_1>q_2$ are integers. On the other hand, the lefthand quantity, r_2-r_1 is strictly smaller than m. This is because $r_1\geq 0$ so that r_2-r_1 is no larger than r_2.

Both of the relevant cases thus lead to contradictions, so we must conclude that no argument-pair gives rise to more than one result-pair. □

4.3.2.2 Divisibility, divisors, GCDs

We begin to study the several important aspects of integer divisibility by considering a variety of simple, yet significant, consequences of an integer n's being divisible by an integer m. We leave the following applications of the basic definitions as exercises for the reader.

Proposition 4.2 .

1. *If m divides n, then m divides all integer multiples of n. Symbolically: If m divides n, then m divides cn for all integers c.*
2. *The relation "divides" is* transitive*; see Chapter 3.2.2. Specifically, if [m divides n], and [n divides q for some integer q], then n divides q.*

3. *The relation "divides" distributes over addition.*[6] *Specifically, if [m divides n], and [m divides $(n+q)$ for some integer q], then [m divides q].*
   Hint: $\dfrac{n+q}{m} = \dfrac{n}{m} + \dfrac{q}{m}$.
4. *For any integer $c \neq 0$,*

$$[[m \text{ divides } n] \quad \text{if, and only if,} \quad [cm \text{ divides } cn]].$$

The following result follows from the preceding facts.

Proposition 4.3 *Given integers m, n, and q, if m divides both n and q, then m divides all linear combinations of n and q; i.e., m divides $(sn+tq)$ for all integers s and t.*

Proof. Because m divides both n and q, there exist integers k_1 and k_2 such that $k_1 \cdot m = n$ and $k_2 \cdot m = q$. By the distributive law, we therefore have:

$$(k_1 \cdot s \; + \; k_2 \cdot t)m \; = \; sn + tq$$

for any s and t. □

Among the common divisors of integers n and q, a particularly significant one is their *greatest common divisor*, which is the largest integer that divides both n and q. We abbreviate "greatest common divisor" by GCD , and we write

$$m \; = \; \text{GCD}(n,q)$$

to identify an integer m as the GCD of n and q.

We are finally ready for our first major result about integer division and divisors.

Proposition 4.4 (Bézout's identity) *For positive integers n and q, $\text{GCD}(n,q)$ is the smallest positive linear combination of n and q.*
Stated alternatively: For any positive integers n and q, there exist integers s and t, not necessarily positive, such that

$$sn + bq \; = \; \text{GCD}(n,q).$$

Proof. Consider the set of all integer linear combinations of n and q:

$$L_{n,q} \stackrel{\text{def}}{=} \{sn+tq \mid s,s \in \mathbb{Z}\} \; \subseteq \; \mathbb{Z}.$$

Note that both n and q belong to $L_{n,q}$, because of the respective cases $(s=1,t=0)$ and $(s=0,t=1)$. One consequence of this is that $L_{n,q}$ has a nonempty subset, call it $L_{n,q}^{(>0)}$, all of whose elements are *positive* integers.

By the *Well-Ordering law of the positive integers*, the set $L_{n,q}^{(>0)}$ has a smallest element, call it m_0. By definition of $L_{n,q}^{(>0)}$, m_0 is a positive integer, and there exist integers s_0 and t_0 such that

[6] We use the term "distributes" in the sense of the Distributive Law; see Chapter 5.1.2.C.

$$m_0 \;=\; s_0 n + t_0 q.$$

We claim that $L_{n,q}^{(>0)}$ in fact *consists precisely of all positive-integer multiples of* m_0. Were this not the case, there would be an element m of $L_{n,q}^{(>0)}$ that is not a (positive-integer) multiple of m_0. Let m_1 be the *smallest* such element m. We then have

1. Because $m_1 \in L_{n,q}^{(>0)}$, there exist integers s_1 and t_1 such that

$$m_1 \;=\; s_1 n + t_1 q.$$

2. Because m_0 is the *smallest* element of $L_{n,q}^{(>0)}$, the difference have $m_2 \stackrel{\text{def}}{=} m_1 - m_0$ must be positive, so that $m_2 \;=\; (s_1 - s_0)n + (t_1 - t_0)q$ must belong to $L_{n,q}^{(>0)}$.

Now, we are in trouble because of the following incompatible facts.

- On the one hand, m_2 *is not* a multiple of m_0
  If it were a multiple, then we would have m_0 dividing both m_0 (trivially) and $m_2 = m_1 - m_0$. But this would imply that m_0 divides m_1, contrary to assumption.
- On the other hand, m_2 *is* a multiple of m_0
  This is because $m_2 < m_1$, while m_1 is the *smallest* element of $L_{n,q}^{(>0)}$ that is not a multiple of m_0.

This contradiction forces us to conclude that integer m_1 does not exist; in other words: all elements of $L_{n,q}^{(>0)}$ are multiples of m_0.

Let us summarize. The set of positive integer linear combinations of n and q consists entirely of integer multiples of a single integer m_0. This means, in particular, that m_0 is a common divisor of n and q. The only way this situation could hold is if $m_0 = \text{GCD}(n,q)$, as claimed in the proposition. □

Bézout's identity has the following significant corollary.

Corollary 4.1 *Every linear combination of n and q is a multiple of* $\text{GCD}(n,q)$*, and vice-versa.*

Greatest common divisors are fundamental companions of pairs of integers, with manifold computational applications. How does one compute them? This question was addressed millennia ago, by Euclid, who authored the following result, which led to the GCD-computing algorithm that bears his name.

For any integers n and $m > 0$, let $\text{REM}(m,n)$ denote the remainder in the Euclidean division expression (4.1) for m and n; i.e.,

$$n \;=\; q \cdot m \;+\; \text{REM}(m,n).$$

Notice here that we use an integer function to express the reminder since it will be used as the result of an operation on two integers (similarly to what we did for the GCD).

Proposition 4.5 *For any integers n and $m > 0$,*

$$\text{GCD}(m,n) \;=\; \text{GCD}(m,\ \text{REM}(m,n)).$$

Proof. For integers $x > 0$ and $y \geq 0$, we denote by $D(x,y)$ the set of common divisors of x and y, i.e., the set of integers that divide both x and y. We prove the proposition by showing, as follows, that the sets $D(m,n)$ and $D(m,\text{REM})$ actually contain precisely the same elements.

- $D(m,n) \;\subseteq\; D(m,\text{REM}(m,n))$.
  Say that the integer d divides both m and n, i.e., that $d \in D(m,n)$. By Proposition 4.1, we know that $n \;=\; q\cdot m \;+\; \text{REM}(m,n)$. By property 3 in Proposition 4.2, we must then have $d \mid \text{REM}(m,n)$. This means that $d \in D(m,\text{REM}(m,n))$.
- $D(m,\text{REM}(m,n)) \;\subseteq\; D(m,n)$.
  Say that the integer d divides both m and $\text{REM}(m,n)$, i.e., that $d \in D(m,\text{REM}(m,n))$. By Proposition 4.3, d divides every linear combination of m and $\text{REM}(m,n)$. In particular, d divides the specific combination $q\cdot m \;+\; 1\cdot\text{REM}(m,n) \;=\; n$. Thus, d divides n, so that $d \in D(m,n)$.

Since we thus have $D(m,n) \;=\; D(m,\text{REM}(m,n))$, we know that the sets contain the same largest element:

$$\max\left(D(m,n)\right) \;=\; \max\left(D(m,\text{REM}(m,n))\right);$$

the proposition follows. □

4.4 The Rational Numbers

Each enrichment of our number system throughout history has been a response to a deficiency with the then-current system. The deficiency that instigated the introduction of the rational numbers was the fact that many integers do not divide certain other integers.

This situation led to practical problems as civilization developed to the point where communities strove to share commodities that were physically divisible. You can always cut a pizza into any desired number of slices—but mandating such an action is awkward if you lack the terminology to describe what you want to achieve.

The situation also led to an intellectual problem, when viewed from a modern perspective. The arithmetic operation *multiplication* was surely recognized not long after its slightly more fundamental sibling operation *addition*. In many ways, these two operations mimic one another. Both are *total bivariate functions* that take a pair of numbers and produce a number; both are *commutative*, in that the argument numbers can be presented in either order without changing the result:

$$(\forall a,b)\ \left[[a+b \;=\; b+a] \quad \text{and} \quad [a\cdot b \;=\; b\cdot a]\right]$$

both are *associative*, in the sense asserted by the equations

$$(\forall a,b)\ \big[[a+(b+c)\ =\ (a+b)+c]\quad\text{and}\quad[a\cdot(b\cdot c)\ =\ (a\cdot b)\cdot c]\big]$$

If we restrict focus to the *integers*, however, there is a glaring difference between addition and multiplication. To wit, addition has a "partner operation", *subtraction*, that operates as a type of *inverse operation*:

$$(\forall a,b,c)\big[\text{if}\quad[c=a+b]\quad\text{then}\quad[a=c-b]\big]$$

(We call $c-b$ the *difference* between c and b.) Within the context of the integers, multiplication has no such "partner". We respond to this imbalance by inventing a "partner" for multiplication, and we call it *division*, denoted $\div$. Now, division cannot completely mimic subtraction because of the technical problems that arise from the *multiplicative annihilation* properties of the integer 0:

$$(\forall a)\,[a\cdot 0\ =\ 0\cdot a\ =\ 0]$$

There is no way to "undo", or "invert" the operation multiplication-by-0, because that operation is not one-to-one. But if we frame the operation of division carefully—specifically, by avoiding division by 0, then we can endow multiplication with the desired "partner":

$$(\forall a,b,c)\big[\text{if}\quad[c=a\cdot b]\quad\text{and if}\quad[b\neq 0]\quad\text{then}\quad[a=c\div b]\big]$$

(We call $c\div b$ the *quotient of c by b*.) We are almost at the end of our journey. All we need is a way to speak about specific quotients. When integer b divides integer c, as when $c=12$ and $b=4$, it is natural to write $12\div 4\ =\ 3$, but how should we denote the quotient $12\div 5$ which is not an integer? Enter the rational numbers!

4.4.1 The Rationals: Special Ordered Pairs of Integers

The set $\mathbb{Q}$ of *rational* numbers—often abbreviated as just "the rationals"—was invented to name the quotients referred to in the preceding paragraph. Formally:

$$\mathbb{Q}\ \stackrel{\text{def}}{=}\ \{0\}\ \cup\ \{p/q\mid p,q\in\mathbb{Z}\setminus\{0\}\}$$

Each element of $\mathbb{Q}$ is called a *rational* number; each *nonzero* rational number p/q is often called a *fraction*; some people reserve the word "fraction" for the case $q>p$, because the word seems to connote "less than the whole", but this does not seem to be a valuable distinction.

In analogy with our treatment of integers, we reserve the notation $\mathbb{Q}^+$ for the *positive* rationals.

An alternative, mathematically more advanced, way of defining the set $\mathbb{Q}$ is as *the smallest set of numbers that contains the integers and is closed under the operation*

of dividing any number by any nonzero number. The word "*closed*" here means that, for every two numbers $p \in \mathbb{Q}$ and $q \in \mathbb{Q} \setminus \{0\}$, the quotient p/q belongs to $\mathbb{Q}$.

Numerous notations have been proposed for denoting rational numbers in terms of the integers they are "built from." Most of these notations continue our custom of employing the single symbol "0" for the number 0, but notations such as $0/q$ (where $q \neq 0$) are permissible when they arise as part of a calculation or an analysis. For the nonzero elements of $\mathbb{Q}$, we traditionally employ some notation for the operation of division and denote the quotient of p by q using one of the following:

$$p/q \quad \text{or} \quad \frac{p}{q} \quad \text{or} \quad p \div q \tag{4.3}$$

The integer p in any of the expressions in (4.3) is the *numerator* of the fraction; the integer q is the *denominator*.

4.4.2 The Rational Number line versus the Integer Number Line

There are many ways to compare the sets $\mathbb{Z}$ and $\mathbb{Q}$ in ways that enhance our understanding of both sets. We craft a comparison that focuses on the similarities and differences in the two sets' number lines, using Section 4.3.1 as the reference for the integer number line.

As the first point in our comparison, we remark that every integer $n \in \mathbb{Z}$ can be encoded as a rational number. Specifically, we represent/encode the integer $n \in \mathbb{Z}$ by the rational p/q whose numerator is $p = n$ and whose denominator is $q = 1$. This encoding is so intuitive that most people would write "$n = n/1$" and ignore the fact that this is expressing an encoding rather than an equality. We know with hindsight that this intellectual shortcut can cause no problems, but it is important to be aware that we are using a shortcut, for (at least) two reasons.

1. We should contemplate *why* the encoding "can cause no problems." Answering this question will enhance our understanding of both $\mathbb{Z}$ and $\mathbb{Q}$. *What essential properties of rationals and integers does the proposed encoding preserve?* To get started, note that the encoding preserves the special characters of the numbers 0 and 1—because the following equations hold: $0/1 = 0$ and $1/1 = 1$.
2. There are intuitively similar situations wherein one's intuition turns out to be wrong! One such situation occupies Section 4.4.2.2, wherein we demonstrate that the sets $\mathbb{Z}$ and $\mathbb{Q}$ "have the same size", and the more advanced Section 4.5.4, wherein we show that the set $\mathbb{R}$ of real numbers is (in a formal sense) "larger" than sets $\mathbb{Z}$ and $\mathbb{Q}$. (*Even the fact that we can discuss the relative "sizes" of infinite sets is interesting!*)

4.4.2.1 Comparing $\mathbb{Z}$ and $\mathbb{Q}$ via their number-line laws

The rational numbers share some, but not all, of the number-line laws of the integers, as enumerated in Section 4.3.A. We now adapt for $\mathbb{Q}$ that section's discussion of $\mathbb{Z}$'s number line.

The sets $\mathbb{Q}$ and $\mathbb{Q}^+$ are both *totally ordered*, in the manners expressed by the Trichotomy laws for rational numbers.

The Trichotomy laws for the rational numbers
(i) *For each rational $a \in \mathbb{Q}$, precisely one of the following is true.*

$$a \text{ equals 0: } (a = 0) \qquad a \text{ is } \textit{positive}\text{: } (a > 0) \qquad a \text{ is } \textit{negative}\text{: } (a < 0)$$

(ii) *Every rational $a \in \mathbb{Q}^+$ is positive $(a > 0)$.*

The total ordering of $\mathbb{Q}$ is expressed as follows
(iii) *For any rationals $a, b \in \mathbb{Q}$, precisely one of the following is true.*

$$a = b \quad \text{or} \quad a < b \quad \text{or} \quad a > b$$

As with the integers, the rationals can be visualized via a (2-way infinite) number line. But the rational line is much harder to visualize, mainly because the rationals do *not* enjoy the well-ordering or discreteness or "between-ness" of the integers.

The set $\mathbb{Q}$ is not *well-ordered.*

For illustration: The set

$$S \;=\; \{a \in \mathbb{Q} \mid 0 < a \leq 1\}$$

has no smallest element. If you give me a rational $p \in S$ that you claim is the smallest element of the set, then I shall give you $p/2$ as a smaller one.

The set $\mathbb{Q}$ does not *obey the "Between" laws.*

In fact, $\mathbb{Q}$ violates the "Between" laws in a very strong way: *For any two unequal rationals, a and $b > a$, there are infinitely many rationals between a and b.*

One can specify such an infinite set for the pair a, b in myriad ways. Here is a simple such set, call it $S_{a,b}$.

$$S_{a,b} \;=\; \left\{ \frac{a+b}{k} \;\mid\; k \in \mathbb{Z} \right\}. \tag{4.4}$$

4.4.2.2 Comparing $\mathbb{Z}$ and $\mathbb{Q}$ via their cardinalities

Our final comparison between the rationals and the integers compares the relative "sizes", or, *cardinalities* of $\mathbb{Z}$ and $\mathbb{Q}$. Informally, *Are there "more" rationals than integers?*

Consider the following facts.

- Every integer is a rational number, as attested to by the "encoding"

$$\text{Encode} \quad n \in \mathbb{Z} \quad \text{by} \quad \frac{n}{1} \in \mathbb{Q}. \tag{4.5}$$

- There are infinitely many non-integer rational numbers between every pair of adjacent integers, as attested to by every set $S_{n,n+1}$ as defined in (4.4).

Thus, the set $\mathbb{Z}$ of integers is a *proper* subset of the set $\mathbb{Q}$ of rationals: symbolically, $\mathbb{Z} \subset \mathbb{Q}$. To many, this subset relation provides an intuitively compelling argument that

there are more rational numbers than integers.

For us—and for the general mathematical community—the preceding intuition provides a compelling argument only for the fact that reasoning about infinite sets demands subtlety and care. For this community, only the formal setting of Section 7.3.3.A allows us to reason cogently about the relative "sizes" of infinite sets. Within this setting, we show that

the set $\mathbb{N}$ *has the same cardinality as the set* $\mathbb{Q}$.

Mirroring Proposition 7.9, we have

Proposition 4.6 $|\mathbb{Q}| = |\mathbb{N}|$.

Proof. Since the proof of this result is adapted from that of Proposition 7.9, we provide only a sketch, leaving details to an exercise.

First, we note that the encoding f defined by

$$(\forall n \in \mathbb{N}) \left[f(n) = \frac{n}{1} \right]$$

provides an injection from $\mathbb{N}$ into $\mathbb{Q}$. This injection verifies that $|\mathbb{Q}| \geq |\mathbb{N}|$.

For the converse relation, we proceed in two steps.

1. Let the function g associate each rational $p/q \in \mathbb{Q}$ with the ordered pair $\langle a, b \rangle \in \mathbb{N} \times \mathbb{N}$ that is obtained by expressing p/q in *lowest terms*; that is,
   - $\dfrac{p}{q} = \dfrac{a}{b}$.
   - The rational $\dfrac{a}{b}$ is in *lowest terms*, in the sense that a and b share no non-unit common divisor.

   Clearly, g is an injection from $\mathbb{Q}$ into $\mathbb{N} \times \mathbb{N}$.
2. Let the function h be an injection from $\mathbb{N} \times \mathbb{N}$ into $\mathbb{N}$. Sample such injections can be found in the proof of Proposition 7.9.

Since the composition of injections is again an injection, the composite injection $g \circ h$ verifies that $|\mathbb{N}| \geq |\mathbb{Q}|$.

Combining the preceding derived inequalities completes the proof. □

4.5 The Real Numbers

4.5.1 Inventing the Real Numbers

Each subsequent augmentation of our system of numbers inevitably gets more complicated than the last: one solves the easy problems first. The deficiency in the system of rational numbers harkens back to historical time, roughly $2\frac{1}{2}$ millennia ago. The ancient Egyptians were prodigious builders who mastered truly sophisticated mathematics in order to engineer their temples and pyramids. The ancient Greeks perpetuated this engineering tradition, but they added to it the philosophical "soul" of mathematics.

Numbers were (literally) sacred objects to many (philosophically oriented) Greeks, and they invented ways of thinking about mathematical phenomena that are quite "modern" to our perspective, in order to understand *why* certain facts were true, in addition to knowing *that* they were true. One intellectual project in this spirit had to do with the way they designed constructions such as temples. They were attracted to geometric contructions that could be accomplished using only *straightedges and compasses*. And—most relevant to our story—they preferred that the relative lengths of linear sections of their structures be *commensurable*, in the following sense. *Integers* $x, y \in \mathbb{N}$ *are* commensurable *if there exist* $a, b \in \mathbb{N}$ *such that*

$$ax = by \quad \text{or, equivalently,} \quad x = \frac{b}{a}y.$$

As Greek philosophers contemplated their desire to employ commensurable pairs of integers in constructions, they discovered that this goal was impossible even in moderately simple constructions. The "poster child" of this assertion is perceptible in *the diagonal of the square with unit-length sides* or, equivalently, in *the hypotenuse of the isosceles right triangle with unit-length legs*. In both situations, one found that the unit lengths of the structure's sides or of its legs were accompanied by the inevitably *non-commensurability* of the length of the square's diagonal or the triangle's hypotenuse; in current terminology, the length of the diagonal and the hypotenuse is $\sqrt{2}$. The Greek mathematicians, as reported by the renowned mathematician Euclid,[7] proved, using current terminology, that $\sqrt{2}$ is not rational. (We rephrase Euclid's proof imminently, in Proposition 4.7.) The conclusion from this proof is that a number system based solely on the integers and rationals was inadequate. In response, the philosophers augmented our number system by introducing *surds* or, as we more commonly term them, *radicals* . The augmentation thus begun culminated in what we know as the real number system. Since our intention in this introduction has been to justify the journey along that trajectory, we leave our historical digression and turn to our real focus, the set $\mathbb{R}$ of *real numbers*.

[7] Euclid wrote extensively on this and related subjects, especially regarding geometry and what is currently known as number theory.

4.5.2 Defining the Real Numbers via Their Numerals

For any integer $b > 1$, the real numbers are the numbers that can be named by *infinite* strings built out of the digits $\{0, 1, \ldots, b-1\}$;[8] the resulting strings are called *b-ary numerals*. There are a couple of ways to form b-ary numerals; we shall discuss some of the most common ones in Section 4.7. For now, we define real numbers as those that can be represented by a *base-b numeral*, for some integer *base* $b > 1$. Such a numeral is a string of the form

$$\alpha_n \alpha_{n-1} \cdots \alpha_1 \alpha_0 . \beta_0 \beta_1 \beta_2 \cdots$$

and represents the (real) number

$$NUM_b(alpha_n \alpha_{n-1} \cdots \alpha_1 \alpha_0 . \beta_0 \beta_1 \beta_2 \cdots) \stackrel{\text{def}}{=} \sum_{i=0}^{n} \alpha_i \cdot b^i + \sum_{j \geq 0} \beta_j \cdot b^{-j}.$$

By prepending a "negative sign" (or, "minus sign") $-$ to a numeral or a number, one renders the thus-embellished entity as negative.

I put NUM here without telling what it is, there is a reference to 4.7 where you defined NUM and the positional numerals, but I think the whole definition should be placed here and 4.7 should refer to this section, are you OK with this?

4.5.3 Not All Real Numbers Are Rational

We close this section by verifying the earlier-mentioned assertion about the non-commensurability of the length of the diagonal of a square with the (common) length of its sides—or, equivalently, the leg-length of an isosceles right triangle with the length of its hypotenuse.

Proposition 4.7 *The real number* $\sqrt{2} = 2^{1/2}$ *is not rational.*

As with many results that arise in our mathematical journey, we provide multiple—in this case, two—proofs for Proposition 4.7, which build upon quite different mathematical insights. In Section 4.5.3 we provide the classical proof of the result. This proof invokes a simple provision of Theorem 7.1, to exploit the divisibility properties of integers. In Section 4.5.3, we provide a proof of the results that builds on the Pythagorean Theorem (Theorem 4.1) to develop geometric insights.

[8] This is not the traditional way that a mathematician would define the set of real numbers, but it is correct and adequate for thinking about the set.

Using divisibility to prove that $\sqrt{2} \notin \mathbb{Q}$

Proof. We now prove Proposition 4.7 *by contradiction*, a proof technique described in Chapter 3.3.4.

I don't think you should refer to the type of proof here (contradiction).... I let you choose if we keep the remark or not. Or should we systematically refer to the type of proof used?

Let us assume, for contradiction, that $\sqrt{2}$ is rational. By definition, then, $\sqrt{2}$ can be written as a quotient

$$\sqrt{2} = \frac{a}{b}$$

for positive integers a and b. In fact, we can also insist that a and b *share no common prime factor*. For, if a and b shared the prime factor p, then we would have $a = p \cdot c$ and $b = p \cdot d$. In this case, though, we would have

$$\sqrt{2} = \frac{a}{b} = \frac{p \cdot c}{p \cdot d} = \frac{c}{d},$$

by cancellation of the common factor p. We can eliminate further common prime factors if necessary until, finally, we find a quotient for $\sqrt{2}$ whose numerator and denominator share no common prime factor. This must occur eventually because each elimination of a common factor leaves us with smaller integers, so the iterative elimination of common factors must terminate.

Let us say that, finally,

$$\sqrt{2} = \frac{k}{\ell} \tag{4.6}$$

where k and ℓ share no common prime factor. Let us square both expressions in (4.6) and multiply both sides of the resulting equation by ℓ^2. We thereby discover that

$$2\ell^2 = k^2. \tag{4.7}$$

This rewriting exposes the fact that k^2 is *even*, i.e., *divisible by* 2. But, Theorem 7.1 tells us that *if k^2 is divisible by* 2*, then so also is k*. This means that $k = 2m$ for some positive integer m, which allows us to rewrite (4.7) in the form

$$2\ell^2 = k^2 = (2m)^2 = 4m^2. \tag{4.8}$$

Hence, we can divide the first and last quantities in (4.8) by 2, to discover that

$$\ell^2 = 2m^2.$$

Repeating the invocation of Theorem 7.1 now tells us that the integer ℓ must be even. We now see that *both k and ℓ are even, i.e., divisible by* 2. This contradicts our assumption that k and ℓ share no common prime divisor!

Since every step of our argument is ironclad—except for our assumption that $\sqrt{2}$ is rational, we conclude that that assumption is false! The Proposition is verified. □

The proof of Proposition 4.7 is a classical (and early) example of *proof by contradiction*, as discussed in Section 3.3.4.

A geometric proof that $\sqrt{2} \notin \mathbb{Q}$

Proof. Our geometric proof repeatedly invokes Fig. 4.3, which suggestively invokes

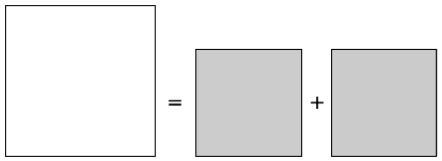


Fig. 4.3 *A geometric depiction the Pythagorean Theorem and its underlying equation:* $a^2 = b^2 + b^2$.

the Pythagorean Theorem. The figure presents three squares. The intention is that the two small grey squares are identical, with common area A, while the large white square has double this area. By the Pythagorean Theorem, if the small squares have (common) side-lengths $b \in \mathbb{N}^+$, hence shared area $A = b^2$ each, then the large square has side-lengths $a \stackrel{\text{def}}{=} \sqrt{2}b$, hence area $a^2 = 2b^2$.

Thus set up, we begin to pursue our contradiction. As in the classical proof shown above, we start with the assumption that $\sqrt{2}$ is rational. Within the context of Fig. 4.3, this means that

$$\sqrt{2} \;=\; \frac{a}{b}$$

for $a, b \in \mathbb{N}^+$. Since all that we have said thus far holds for arbitrary a and b, we are free to consider the assumption's implications for the same situation as before, namely, that a and b do not share any common prime factor. Note additionally that, because[9]

$$\sqrt{2} \;>\; 1.4,$$

we know that $a > b$.

Of course, our demands on the relationship between the numerator a and the denominator b lose no generality in our argument. To wit, "$\frac{a}{b}$" is just one name for the depicted rational number, and choosing any specific name has no impact on the number itself.

Now that we have the suggestive "equation" presented in Fig. 4.3, we can manipulate the depicted squares. Let us embed both of the grey $b \times b$ squares of Fig. 4.3 into the white $a \times a$ square, in the overlapped manner depicted in Fig. 4.4: one grey

[9] If this inequality is new to you, then just note that $(1.4)^2 = 1.96$, which is less than 2.

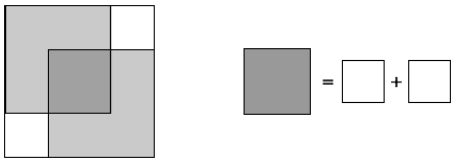


Fig. 4.4 *Construction of a smaller pythagorian equation.*

square is nestled into the northwestern corner of the white square, while the other is nestled into the southeastern corner. The overlapping of the grey squares under this embedding creates a new square—depicted in dark grey—in the center of the white square, while it leaves unoccupied two small squares, which remain white in the figure.

Now, let us get quantitative.

- On the first side, the fact that the combined areas of the two grey squares equal the area of the white square guarantees that the area of the dark grey overlap-square is equal to the combined areas of the small unoccupied white squares.
- On the other side, because the side-lengths of the large white square is a, while those of the grey squares is b, the side-lengths of the small white square is $a-b$, and the side-lengths of the dark grey overlap-square is $2b-a$. All of these side-lengths are positive because of the value of $\sqrt{2}$: (*i*) $a > b$ because $\sqrt{2} > 1$; (*ii*) $2b > a$ because $\sqrt{2} < 2$.

The preceding facts allow us to label the squares of Fig. 4.3 differently than we did earlier—and derive a different valid "equation". As we did at the beginning of this discussion, we again invoke the Pythagorian Theorem, but now we do so while focusing—cf., Fig. 4.4—on the dark grey overlap-square (which plays the role of the large square in Fig. 4.3) and the two small white squares (which play the role of the two small squares in Fig. 4.3). Whereas our original focus led to the putative rational value $\frac{a}{b}$ for $\sqrt{2}$, the new focus yields the putative rational value $\frac{2b-a}{a-b}$ for $\sqrt{2}$. We thus have

$$\sqrt{2} \;=\; \frac{a}{b} \;=\; \frac{2b-a}{a-b},$$

where $2b-a < a$ and $a-b < b$. In the light of the Fundamental Theorem of Arithmetic (Theorem 7.1), this new rational name for $\sqrt{2}$ contradicts our beginning assumption that $\frac{a}{b}$ is the simplest name. □

I did not find a better word for simplest...

I have another geometric proof for the same result, I added it in the separate file which will serve as exercices.

4.5.4 $\mathbb{R}$ is uncountable, hence is "bigger than" $\mathbb{Z}$ and $\mathbb{Q}$

The main result of this section establishes the uncountability of the set $\mathbb{R}$. We thereby have an argument that the infinitude of real numbers is *of a higher order* than the infinitude of the integers. In fact, Georg Cantor used this result as the base of his study of orders of infinity.

Proposition 4.8 *The set $\mathbb{R}$ of real numbers is not countable. In particular, there is no injection $f : \mathbb{R} \to \mathbb{N}$.*

Proof. Our multi-step proof shows that the assumption of $\mathbb{R}$'s countability leads to a contradiction. Being built around Georg Cantor's renowned *diagonalization construction*, this argument provides the most sophisticated proof by contradiction in our text. The reader might want to review the reasoning underlying such argumentation, in Section 2.2.4, as a "warm-up."

4.5.4.1 Plotting a strategy to prove uncountability

Invoking the definition of countability, our proof begins with the assumption that $|\mathbb{R}| \leq |\mathbb{N}|$ and demonstrates that this assumption leads to a contradiction. We simplify our goal in several steps.

(a) Recasting the problem in terms of bijections. We now set out to recast our goal into a form that will be easier to manage mathematically, by invoking our prior knowledge about the sets $\mathbb{N}$ and $\mathbb{R}$. We begin with two simplifying lemmas.

Lemma 4.1. *There exists an injection $f : \mathbb{N} \to \mathbb{R}$; i.e., $|\mathbb{N}| \leq |\mathbb{R}|$.*

Verification.

It sounds like we have a confusion for the definition of NUMb... In the other parts, NUM is the integer For each nonnegative integer $n \in \mathbb{N}$, let $\text{NUM}_b(n)$ denote the shortest base-b numeral for n, i.e., a numeral with no leading 0s. The mapping that associates each $n \in \mathbb{N}$ with the infinite string

$$\text{NUM}_b(n) \; . \; 00\cdots$$

is an injection from $\mathbb{N}$ into $\mathbb{R}$. To wit, when given a real numeral that has only 0s to the right of its radix point, one produces the integer n by stripping the numeral of its radix point and all 0s to the right of the point, and then evaluating the remaining string of digits, which is $\text{NUM}_b(n)$, according to Section 4.5.2's rules for evaluating integer numerals. □

Lemma 4.2. *If there exists an injection $g : \mathbb{R} \to \mathbb{N}$, then there exists a bijection $h : \mathbb{N} \leftrightarrow \mathbb{R}$. In other words, if $|\mathbb{R}| \leq |\mathbb{N}|$, then $|\mathbb{N}| = |\mathbb{R}|$.*

Verification. This is an immediate consequence of the Schröder-Bernstein Theorem (Theorem 7.3). □

It is somewhat surprising that our proof is simplified by converting our initial assumption

$$|\mathbb{R}| \leq |\mathbb{N}|$$

to the stronger assumption

$$|\mathbb{R}| = |\mathbb{N}|,$$

but Cantor's diagonal argument deals quite gracefully with the latter assumption.

(b) Focus on real numbers between 0 *and* 1. Our next refinement replaces our target set $\mathbb{R}$ by its proper subset $\mathbb{R}_{(0,1)}$, all of whose elements have infinite decimal numerals of the form

$$0 \,.\, \delta_0\delta_1\cdots$$

where each δ_i is a decimal digit: $\delta_i \in \{0,1,2,3,4,5,6,7,8,9\}$.[10]

Of course, if we prove that the proper subset $\mathbb{R}_{(0,1)} \subset \mathbb{R}$ is uncountable, then it will follow that $\mathbb{R}$ is uncountable. (In informal terms which can be made formal, any putative injection $f : \mathbb{R} \to \mathbb{N}$ "contains" an injection $f_{(0,1)} : \mathbb{R}_{(0,1)} \to \mathbb{N}$.)

A. Seeking a bijection $h : \mathbb{N} \leftrightarrow \mathbb{R}_{(0,1)}$

We assume, for contradiction, that the targeted bijection g exists. As part of this two-way mapping, there exists an *injection*

$$h : \mathbb{N} \to \mathbb{R}_{(0,1)},$$

which we view as an *enumeration* of the elements of $\mathbb{R}_{(0,1)}$. Specifically, for each integer $k \in \mathbb{N}$, we can think of $h(k)$ as the "kth number in the set $\mathbb{R}_{(0,1)}$." We thereby view h as producing an "infinite-by-infinite" matrix Δ of decimal digits, whose kth row is the infinite string of decimal digits $\text{NUM}_{10}(h(k))$. Let us visualize Δ:

$$\Delta = \begin{array}{l} \delta_0 = \delta_{0,0}\ \delta_{0,1}\ \delta_{0,2}\ \delta_{0,3}\ \delta_{0,4}\ \cdots \\ \delta_1 = \delta_{1,0}\ \delta_{1,1}\ \delta_{1,2}\ \delta_{1,3}\ \delta_{1,4}\ \cdots \\ \delta_2 = \delta_{2,0}\ \delta_{2,1}\ \delta_{2,2}\ \delta_{2,3}\ \delta_{2,4}\ \cdots \\ \delta_3 = \delta_{3,0}\ \delta_{3,1}\ \delta_{3,2}\ \delta_{3,3}\ \delta_{3,4}\ \cdots \\ \delta_4 = \delta_{4,0}\ \delta_{4,1}\ \delta_{4,2}\ \delta_{4,3}\ \delta_{4,4}\ \cdots \\ \quad\vdots\quad\ \vdots\quad\ \vdots\quad\ \vdots\quad\ \vdots\quad\ \vdots\quad\ \ddots \end{array}$$

We summarize, for emphasis:

- Each row of Δ consists of the decimal numeral for a number in the set $\mathbb{R}_{(0,1)}$.
- Each number in the set $\mathbb{R}_{(0,1)}$ contributes at least one numeral to the rows of Δ. A number may contribute more than one numeral because of an artifact of positional number systems, which is exemplified by equations such as

$$0.25 = 0.24999\cdots$$

[10] We choose *decimal* numerals for convenience: converting the argument to other number bases—especially the base $b = 2$—slightly complicates clerical details.

We can, thus, view the successive rows of Δ, $h(0)$, $h(1)$, ..., as an enumeration (with possible repetitions) of all of the real numbers in the set $\mathbb{R}_{(0,1)}$.

B. Every bijection $h : \mathbb{N} \leftrightarrow \mathbb{R}$ "misses" some real

We are finally poised to find the contradiction to our assumption that $|\mathbb{R}_{(0,1)}| \leq |\mathbb{N}|$. Specifically, we define from Δ an infinite decimal numeral

$$\Psi \;=\; \psi_0\, \psi_1\, \psi_2\, \psi_3\, \psi_4 \cdots,$$

that *does not* appear in Δ, even though $\text{NUM}_{10}(\Psi) \in \mathbb{R}_{(0,1)}$. For each index $i \in \mathbb{N}$, we define the ith digit ψ_i of Ψ from the ith *diagonal digit*[11] $\delta_{i,i}$ of Δ in the following manner.

$$\psi_i \stackrel{\text{def}}{=} \begin{cases} 0 \text{ if } \delta_{i,i} > 5 \\ 9 \text{ if } \delta_{i,i} \leq 5 \end{cases}$$

The important feature of the definition is the following.

Lemma 4.3. *The string Ψ does not occur as a row of Δ.*

Verification. Focus on an arbitrary row of Δ, say row k, and on the numeral, δ_k, in that row.

If $\delta_{k,k} > 5$ then $\text{NUM}_{10}(\delta_k) - \text{NUM}_{10}(\Psi) > 4 \cdot 10^{-k}$

If $\delta_{k,k} \leq 5$ then $\text{NUM}_{10}(\Psi) - \text{NUM}_{10}(\delta_k) > 4 \cdot 10^{-k}$

In either case, we have $\text{NUM}_{10}(\Psi) \neq \text{NUM}_{10}(\delta_k)$ so that Ψ does not appear as row k of Δ. Since $k \in \mathbb{N}$ is an arbitrary row-index of Δ, we conclude that Ψ does not occur as any row of Δ. □

4.5.4.2 The denouement: There is no bijection $h : \mathbb{N} \leftrightarrow \mathbb{R}$

Because the infinite decimal string Ψ differs from every row of Δ, even though $\text{NUM}_{10}(\Psi) \in \mathbb{R}_{(0,1)}$, we have shown that Δ *does not* contain as a row *every* infinite decimal numeral of a number in $\mathbb{R}_{(0,1)}$. But this contradicts Δ's assumed defining characteristic!

Where could we have gone wrong? Every step of our argument, save one, is backed up by a proof—so the one step that is not so bolstered must be the link that has broken. This one unsubstantiated step is our assumption that the set $\mathbb{R}_{(0,1)}$ is countable. Since this assumption has led us to a contradiction, we must conclude that the set $\mathbb{R}_{(0,1)}$, and hence, the set $\mathbb{R}$, is *not* countable! □

The result and the proof are fine, however, I am sure we can simplify the presentation of the successive steps more clearly. I will come back later at this...

[11] Our use of Δ's "diagonal digits" in this definition is the origin of the term "*diagonal argument*" to describe this proof and its intellectual kin.

4.6 The Complex Numbers

4.6.1 The Basics of the Complex Numbers

Let us denote by $\mathbb{C}$ the set of complex numbers. Each number $\kappa = a + bi \in \mathbb{C}$ has a *real part*—the part that *does not* involve the imaginary unit i—and an *imaginary part*—the part that *does* involve i. To be explicit: the real part of our number κ, is $\mathrm{Re}(\kappa) = a$; the *imaginary part* of our number κ,is $\mathrm{Im}(\kappa) = b$. The notation $\mathrm{Re}(\kappa)$ and $\mathrm{Im}(\kappa)$ is common but not universal.

Using the basic arithmetic laws that we have discussed thus far, plus the defining equation, $i^2 = -1$, of the imaginary unit i, we find that the *product* of two complex numbers, $a + bi \in \mathbb{C}$ and $c + di \in \mathbb{C}$ is the complex number, call it κ,

$$\kappa \;=\; (a+bi)\cdot(c+di) \;=\; (ac-bd)+(ad+bc)i. \tag{4.9}$$

We note that a "direct" implementation of complex multiplication, i.e., one that implements (4.9) literally, requires four real multiplications—namely, ac, bd, ad, bc.

During the 1960s, people first began to pay close attention to the costs associated with various ways of achieving computational results. They sought—and found—a number of procedures that replaced computations involving k real multiplications (a relatively expensive operation) and ℓ real additions (a relatively inexpensive operation) by computations that achieved the same result but used fewer multiplications and not too many more additions. Complex multiplication was one of the operations they studied. Here is the result.

Proposition 4.9 *One can compute the product of two complex numbers using* three *real multiplications rather than four.*

Proof. Although implementing (4.9) "directly" correctly produces the product $\kappa = (a+bi)\cdot(c+di)$, there is another implementation that is *more efficient*. Specifically, the following recipe computes κ using only *three* real multiplications instead of the four real multiplications of the "direct" implementation. We begin to search for this recipe by noting that our immediate goal is to compute both $\mathrm{Re}(\kappa) = ac - bd$ and $\mathrm{Im}(\kappa) = ad + bc$. We can accomplish this by computing the *three* real products

$$(a+b)\cdot(c+d); \qquad ac; \qquad bd \tag{4.10}$$

and then noting that

$$\begin{aligned} \mathrm{Im}(\kappa) &= (a+b)\cdot(c+d) - ac - bd, \\ \mathrm{Re}(\kappa) &= ac - bd \end{aligned} \tag{4.11}$$

We thereby achieve the result of the complex multiplication described in (4.9) while using only *three* real multiplications.

Of course, a full reckoning of the costs of the two implementations we have discussed exposes the fact that the implementation that invokes (4.10) and (4.11) uses

three real additions rather than the *two* real additions of the "direct" implementation. But this entire exercise was predicated on the observation that each real addition is much less costly than a real multiplication, so trading one multiplication for one addition is an unqualified "win". □

I added a sentence to refer to an exercice dealing with karatsuba which uses the same idea... Notice that this technique is classical and it has been used in many other situations. For instance while multiplying two integers in base 2 (see exercice **??**).

4.7 Numerals We Can Work With

4.7.1 Positional Number Systems

The most common family of *operational* numerals for real numbers—i.e., numerals that enable one to do things such as perform arithmetic (add, multiply, etc.) on the named numbers—is the family of *positional number systems*. Each system in this family is identified by its *base*, which is usually[12] an integer $b > 1$. For any base b, we define the set $B_b = \{0, 1, \ldots, b-1\}$ of *digits in base b*. To aid legibility, *within the context of base-b positional numerals*, we denote the digit $b-1$ as a single character, $\bar{b}$. We then form base-b numerals in the following way.

The entire edifice of positional numerals builds on the concept of geometric summations*, a mathematical structure that we shall learn to manipulate, evaluate, and compute with in Section 6.2.2. We need only special aspents of this topic here.*

the following should be put before in 4.5.2

A base-b numeral is a string having three segments.

1. The numeral begins with its *integer part*, which is a *finite* string of digits from B_b. We denote the integer-part string as: $\alpha_n\alpha_{n-1}\cdots\alpha_1\alpha_0$.
2. The numeral continues with a single occurrence of the *radix point* "."
3. The numeral ends with its *fractional part*, which is a string—*finite or infinite*—of digits from B_b. We denote the fractional-part string as: $\beta_0\beta_1\beta_2\cdots$.

Our completed numeral now has the form

$$\alpha_n\alpha_{n-1}\cdots\alpha_1\alpha_0 \,.\, \beta_0\beta_1\beta_2\cdots \tag{4.12}$$

where the α_i and the β_j are *base-b digits* i.e., elements of the set B_b, and "." represents the *(base-b) radix point*.
The numeral depicted in (4.12) represents the (real) number[13]

[12] In rather specialized contexts one encounters number bases that are not positive integers.

[13] The notation "$\text{NUM}_{10}(x)$" in (4.13) denotes an operator that produces the *numerical value* of the numeral x written in base-b.

$$\text{NUM}_{10}(\alpha_n\alpha_{n-1}\cdots\alpha_1\alpha_0\,.\,\beta_0\beta_1\beta_2\cdots) \stackrel{\text{def}}{=} \sum_{i=0}^{n}\alpha_i\cdot b^i + \sum_{j\geq 0}\beta_j\cdot b^{-j}. \qquad (4.13)$$

For emphasis, we note that the base-b integer represented by the numeral's integer part is

$$\text{NUM}_{10}(\alpha_n\alpha_{n-1}\cdots\alpha_1\alpha_0) = \sum_{i=0}^{n}\alpha_i\cdot b^i$$

and the base-b fraction represented by the numeral's fractional part is

$$\text{NUM}_{10}(.\beta_0\beta_1\beta_2\cdots) = \sum_{j\geq 0}\beta_j\cdot b^{-j}$$

By prepending a "minus sign" (or, "negative sign") $-$ to a numeral or a number, one renders the thus-embellished entity as negative.

Note that *two types of sequences of* 0*s do not affect the value of the number represented by a numeral:* (1) an *initial* sequence of 0s to the *left* of the radix point and of all non-0 digits; (2) a *terminal* sequence of 0s to the *right* of the radix point and of all non-0 digits.

One consequence of this fact is that we lose no generality by insisting that every numeral have the following *normal form:*

– a finite sequence of digits,
– followed by a radix point,
– followed by an infinite sequence of digits

4.7.2 Recognizing Integers and Rationals from Their Numerals

We have provided an adequate, albeit inelegant, characterization of the real numbers: a number r is real if, and only if, it can be represented by an infinite-length numeral in a positional number system. Because every rational number—hence, also, every integer—is also a real number, every rational number and every integer can also be written as a b-ary numeral, in the form (4.12). For rational numbers and integers, we can make much stronger statements about the forms of their positional numerals.

We have to add a brief discussion about k-ary and k-adic systems...

4.7.2.1 Positional numerals for integers

Proposition 4.10 *A real number is an integer if, and only if, it can be represented by a* finite-length *numeral all of whose nonzero digits are to the left of the radix point.*

Proof. The result is immediate by definition (4.13). In the indicated form, if any β_i is nonzero, then the VALUE of the numeral is non-integral: it has a nonzero fractional part. □

We can go beyond the simple statement of Proposition 4.10 and present the following algorithm that computes the normal-form base-b numeral for an integer n.

I think it would be better to give the process in literal form as the rest of the chapter...

Procedure Normal-Form Numeral(n)
/*Compute the normal-form numeral for a given integer n*/
Initialization.
Set CURRENT-RESIDUE to n
Iterate until CURRENT-RESIDUE $= 1$
Divide CURRENT-RESIDUE by b

The *remainder* upon each division is the next lowest-order digit in the base-b numeral for n.

Validating the procedure. The procedure is an implementation of a method of rewriting univariate polynomials, known variously as *Horner's rule* or *Horner's scheme* [41]. Among its other virtues, the "rule" provides a recipe for computing a degree-d univariate polynomial using $O(d)$ multiplications, rather than the $\Theta(d^2)$ multiplications that appear at first to be needed.

- The "standard" way of writing the polynomial $P(x)$.
  General degree d:
  $P(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{d-1}x^{d-1} + a_dx^d$
  Degree 3:
  $P(x) = a_0 + a_1x + a_2x^2 + a_3x^3$
- Rewriting $P(x)$ using Horner's rule.
  General degree d:
  $P(x) = a_0 + x\cdot(a_1 + x\cdot(a_2 + \cdots + x\cdot(a_{d-2} + x\cdot(a_{d-1} + a_dx))\cdots))$
  Degree 3:
  $P(x) = a_0 + x\cdot(a_1 + x\cdot(a_2 + a_3x))$

The "rule" is so natural that its origins certainly predate the cited 1819 publication, but they are impossible to trace definitively. □

Illustrating the procedure. We use the procedure to produce the normal-form base-2 (binary) numeral for $n = 143$.

Step	CURRENT-RESIDUE	Remainder
1.	143	1
2.	71	1
3.	35	1
4.	17	1
5.	8	0
6.	4	0
7.	2	0
8.	1	1

We have thus derived the following equation which specifies the base-2 normal-form numeral for 143.

$$143_{10} \;=\; 10001111_2$$

4.7.2.2 Positional numerals for rationals

We can completely characterize the positional numerals that represent rational numbers, in terms of the auxiliary notion of an *ultimately periodic* infinite sequence of digits.

An infinite sequence S of numbers is *ultimately periodic* if there exist two *finite* sequences of numbers, A and B, such that S can be written in the following form (we have added spaces to enhance legibility):

$$S \;=\; A\,B\,B\,B\cdots B\,B\cdots \tag{4.14}$$

The intention here is that the sequence B is repeated *ad infinitum*.

Proposition 4.11 *A positional numeral denotes a rational number if, and only if, it is ultimately periodic.*

Proof. 1. Part 1: the "if" clause.

Say first that the real number r has an ultimately periodic infinite base-b numeral. Since the exact lengths of the finite sequences A and B that constitute the numeral, as in (4.14), are not germane to the argument, we arbitrarily denote r by the following normal-form numeral (spaces added to enhance legibility):

$$a_2a_1a_0\,.\,b_0b_1\;c_0c_1c_2\;c_0c_1c_2\cdots c_0c_1c_2\cdots$$

so that

$$A = a_2a_1a_0\,.\,b_0b_1$$
$$B = c_0c_1c_2$$

(Choosing specific lengths for A and B cuts down on the number of "ellipsis dots" we need to denote the numeral, as in "$123123\cdots123\cdots$", hence enhances legibility.)

If we now invoke the evaluation rules of (4.13), we find that

$$\begin{aligned} r &= \text{VALUE}(a_2a_1a_0 \,.\, b_0b_1\; c_0c_1c_2\; c_0c_1c_2\cdots c_0c_1c_2\cdots) \\ &= \text{VALUE}(a_2a_1a_0) \;+\; \text{VALUE}(b_0b_1)\cdot b^{-2} \;+\; \text{VALUE}(c_0c_1c_2)\cdot b^{-5} \\ &\quad +\; \text{VALUE}(c_0c_1c_2)\cdot b^{-8} \;+\; \text{VALUE}(\gamma_0\gamma_1\gamma_2)\cdot b^{-11} \;+\cdots \qquad (4.15) \\ &= \text{VALUE}(a_2a_1a_0) \;+\; \text{VALUE}(b_0b_1)\cdot b^{-2} \;+\; \text{VALUE}(c_0c_1c_2)\cdot \sum_{i=1}^{\infty} b^{-2-3i} \end{aligned}$$

We must change VALUE, is corresponds to NUM_{10} right?

We shall learn in Section 6.2.2 that infinite summations such as the one in (4.15), namely, $\sum_{i=1}^{\infty} b^{-2-3i}$, *converge*—meaning that *they have finite rational sums*—and we learn how to compute these sums. For the purposes of the current proof, we just take this fact on faith, and we denote the summation's finite rational sum by p/q.

Collecting all of this information, we find that there exist *integers* m, n, p, and q such that

$$r \;=\; m \;+\; n/b^2 \;+\; p/q \;=\; \frac{mqb^2+nq+p}{qb^2}.$$

The number r is, thus, the ratio of two integers; hence, by definition, it is rational.

2. Part 2: the " only if" clause.

Say next that the real number r is rational—specifically,

$$r \;=\; s+\frac{t}{q}$$

for nonnegative integers $t < q$ and s. It is only the fraction $t/q < 1$ that can produce an infinite numeral, so it suffices for us to verify the special case

$$r \;=\; \frac{t}{q} \;<\; 1$$

of the proposition.

We prove that $r = t/q$ has an ultimately periodic infinite numeral by using *synthetic division*—the algorithm taught in elementary school—to compute the ratio t/q. As we proceed, keep in mind that we are working in base b. Each of the following successive divisions produces one digit to the right of the radix point, in addition to a possible *remainder* r_i from the set $\{0,1,\ldots,q-1\}$.

Division step	Current numeral	Current remainder
$b \cdot t = a_0 \cdot q + r_0$	$t/q = .a_0 \cdots$	$r_0 < q$
$b \cdot r_0 = a_1 \cdot q + r_1$	$t/q = .a_0 a_1 \cdots$	$r_1 < q$
$\vdots$	$\vdots$	$\vdots$
$b \cdot r_i = a_{i+1} \cdot q + r_{i+1}$	$t/q = .a_0 a_1 \cdots a_{i+1} \cdots$	$r_{i+1} < q$
$b \cdot r_{i+1} = a_{i+2} \cdot q + r_{i+2}$	$t/q = .a_0 a_1 \cdots a_{i+1} a_{i+2} \cdots$	$r_{i+2} < q$
$\vdots$	$\vdots$	$\vdots$

(4.16)

Because of the possible values the remainders r_j can assume, no more than q of the divisions in the (infinite) system (4.16) are distinct. (*This is an application of the pigeonhole principle (Section 2.2.6).*) Because of the way the system proceeds, once we have encountered two remainders, say, r_i and r_{i+k}, that are equal—i.e., $r_i = r_{i+k}$—we must thenceforth observe periodic behavior:

$$\begin{array}{ccccccc} r_i & = & r_{i+k} & = & r_{i+2k} & = & r_{i+3k} & = \cdots \\ r_{i+1} & = & r_{i+k+1} & = & r_{i+2k+1} & = & r_{i+3k+1} & = \cdots \\ \vdots & & \vdots & & \vdots & & \vdots & \\ r_{i+k-1} & = & r_{i+2k-1} & = & r_{i+3k-1} & = & r_{i+4k-1} & = \cdots \end{array}$$

This will engender periodicity in the digits of r's base-b numeral:

$$[\text{INITIAL SEGMENT}][a_i a_{i+1} \cdots a_{i+k-1}][a_i a_{i+1} \cdots a_{i+k-1}] \cdots [a_i a_{i+1} \cdots a_{i+k-1}] \cdots$$

We are, thus, observing the claimed ultimately periodic behavior in r's base-b numeral.

□

We end this section by illustrating the process of generating numerals for rationals via synthetic division. We employ the fraction $t/q = 4/7$ and base $b = 10$.

Division step	Current numeral	Current remainder
$10 \cdot 4 = 5 \cdot 7 + 5$	$4/7 = .5 \cdots$	5
$10 \cdot 5 = 7 \cdot 7 + 1$	$4/7 = .57 \cdots$	1
$10 \cdot 1 = 1 \cdot 7 + 3$	$4/7 = .571 \cdots$	3
$10 \cdot 3 = 4 \cdot 7 + 2$	$4/7 = .5714 \cdots$	2
$10 \cdot 2 = 2 \cdot 7 + 6$	$4/7 = .57142 \cdots$	6
$10 \cdot 6 = 8 \cdot 7 + 4$	$4/7 = .571428 \cdots$	4
$\vdots$	$\vdots$	$\vdots$

The remainder 4 in the last illustrated division step cycles us back to the initial division step, where the "4" came from the numerator of the target fraction. This repetition signals that the entire process cycles from this point on. In other words, we have determined that

$$\frac{4}{7} = .[571428]\,[571428]\,[571428]\,\cdots$$

Propositions 4.10 and 4.11 show us that the three sets of numbers we have defined are a nested progression of successively more inclusive sets, in the sense that *every integer is a rational number* and *every rational number is a real number*. Those interested in the (philosophical) foundations of mathematics might quibble about the verb "is" in the highlighted sentences, but for all practical purposes, we can accept the sentences as written.

4.7.3 Scientific Notation

There is a familiar game in which one is challenged to guess how many beans there are in a jar. The wild ranges of guesses that players make indicate eloquently what is one of the main starting points in the popular-science book *Innumeracy* [61]: While we "know" a lot about even *very* large numbers and *very* small numbers, we often lack *operational* command of the numbers. This fact can be illustrated in at least two ways.

1. The ability to compare the magnitudes of big numbers:

- Can you compare the probabilities of a person's being hit by lightning (say, in Mexico City) or by a car (say, crossing Fifth Avenue in Times Square)?
- Do you know whether you have more hairs on your body than there are grains of sand on the beach at Ipanema, Brazil?
- Do you know whether there were more Homo Sapiens alive on December 31, 1999, than had lived from the Big Bang to December 31, 1945?
- Can you compare the number of rhinoviruses that can populate a square of side-length 1mm with the number of stars visible on a clear night at the summit of Mount Everest?

2. The ability to delineate "how much information" a number tells us: Many of us know—or can calculate—that (in some sense) the distance between Earth and its closest star, the Sun, is, very roughly,

$$\begin{aligned} 93,000,000 &\text{ miles} \\ 148,800,000 &\text{ km} \\ 491,040,000,000 &\text{ feet} \\ 5,892,480,000,000 &\text{ inches} \end{aligned}$$

notice here that people in US need 3 different non-correlated metrics while from the french revolution and "systeme metrique" we need only one (the meter)...

All of these numbers are coarse approximations. In some sense, they all convey exactly the same information, since all are obtained from the first number (the number of miles to the Sun) by simple scaling. Yet, while the first number projects a

modest two (decimal) digits of accuracy, the others project, respectively, four digits, five digits, and six digits. Do all of these numbers convey the same (level of) truth?

Scientists and pedagogues and philosophers have grappled with the problems engendered by innumeracy throughout time; see, e.g., Footnote 8. One ingenious approach within the domain of astronomy has been to establish a new standard unit of distance to express the *very* large distances from Earth to stars beyond our solar system: A *light year* is the distance that light travels in an Earth-year, roughly 9.4607×10^{12} km. By using this measure, one can describe enormous (well, astronomical) numbers without unwarranted appearances of inflated accuracy. The notion of light year plays an important role for astronomy, but it does not port gracefully to other domains, for two reasons: (1) The use of the speed of light as a frame of reference has no meaning when one is, for instance counting grains of sand or numbers of viruses. (2) The scaling factor inherent in a light year is not appropriate for other domains. The widely accepted general alternative to a new scaling unit is *scientific notation*.

Within scientific notation, one specifies an arbitrary number, of arbitrary magnitude via a *rational approximation* of the form

$$.\beta_0\beta_1\beta_2\cdots\beta_{a-1} \times b^s$$

The interpretation is that

- $\beta_0\beta_1\beta_2\cdots\beta_{a-1}$ represents the a base-b *digits of accuracy* that are warranted by the accuracy of one's level of knowledge about the number being specified.
- b^s is the base-b *scaling factor* that adjust the digits of accuracy relative to the radix point.

Within this system of specification, we thus have

.93	$\times 10^8$	miles from Earth to the Sun
.94607	$\times 10^{13}$	kilometers traveled by light in an Earth-year
.31415	$\times 10$	value of π to 5 digits of accuracy
.166	$\times 10^{-23}$	grams of weight of a proton, to 3 digits of accuracy

I suggest to add a section on numbering systems (we already spoke about binary system, I would like to add a section on Fibonacci numbering system), I put it in the following, but may be we will have to put somewhere else, or as an exercice?

4.8 Fibonacci numbering system

We study how Fibonacci numbers can be used as a basis for representing any integers **??**.

Let us first introduce a useful notation: $j \gg k$ iff $j \geq k+2$.

We will first prove the *Zeckendorf's theorem* which states that every positive integer n has a unique representation of the form:

$n = F_{k_1} + F_{k_2} + ... + F_{k_r}$ where $k_1 \gg k_2 \gg ... \gg k_r$ and $k_r \geq 2$.

Here, we assume that the Fibonacci sequence starts at index 1 and not 0, moreover, the decompositions will never consider $F(1)$ (since $F(1) = F(2)$). For instance, the representation of 12345 turns out to be:

$12345 = 10946 + 987 + 377 + 34 + 1 = F(21) + F(16) + F(14) + F(9) + F(2)$

Figure 4.5 shows the decomposition of the first 26 integers written in this system.

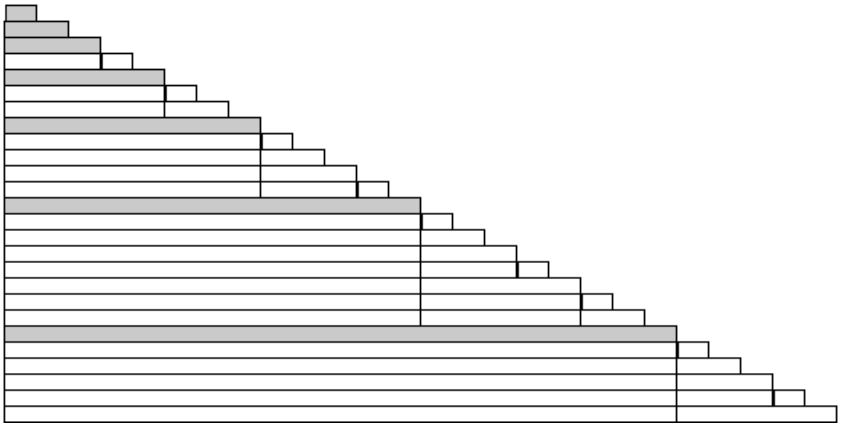


Fig. 4.5 The first integers (on the Y-axis) broken down into the Zeckendorf representation. The shaded rows corresponds to pure Fibonacci numbers.

Proof of Zeckendorf's Theorem

The proof is done by induction on n for proving simultaneously both construction and uniqueness.

- The basis is true since the decomposition is obviously unique for $n = 2$ (and also for $n = 3$). Notice that for $n = 4$, we have $4 = 3 + 1 = F(4) + F(2)$.
- Assume for the induction step that any integer strictly lower than $F(k)$ can be decomposed uniquely as the sum of non-consecutive Fibonacci numbers. We will prove as a consequence that an integer n in the next interval between two consecutive Fibonacci numbers $F(k) \leq n < F(k+1)$ may be decomposed.
  If $n = F(k)$ is a Fibonacci number, the decomposition is reduced to $F(k)$.
  Moreover, it is not difficult to check that it is unique.

  If $n \neq F(k)$ write $n = F(k) + N$.
  As N is strictly lower than $F(k)$, we apply the recurrence hypothesis to decompose it into non-consecutive Fibonacci numbers:
  $n = F(k) + F(k_1) + F(k_2) + \cdots + F(r)$ where $k_2 \gg ... \gg k_r \geq 2$.
  The last point to verify is that $F(k)$ and $F(k_1)$ are not consecutive ($F(k) \gg F(k_1)$), which is done by contradiction:

Assuming k and k_1 are consecutive ($k_1 = k-1$) leads to $n = F(k+1) + F(k_2) + \cdots + F(r)$ which contradicts $n < F(k+1)$.

Any unique system of representation is a numbering system.

The previous theorem ensures that any non-negative integer can be written as a sequence of bits b_i, in other words,

$n = (b_m b_{m-1} ... b_2)_F$ iff $n = \Sigma_{k=2}^{m} b_k F_k$.

Let us compare this system to the binary representation. For instance, the Fibonacci representation of 12345 is $(100001010000100000010)_F$ while $12345 = 2^{13} + 2^{12} + 2^5 + 2^4 + 2^3 + 2^0 = (1100000111001)_2$.

The binary representation is more compact.

The decomposition in the Fibonacci basis of the first integers (starting from $1 = (00001)_F$) is as follows:

$2 = (0010)_2 = F_3 = (00010)_F$
$3 = (0011)_2 = F_4 = (00100)_F$
$4 = (100)_2 = 3 + 1 = (00101)_F$
$5 = (101)_2 = F_5 = (01000)_F$
$6 = (110)_2 = 5 + 1 = (01001)_F$
$7 = (111)_2 = 5 + 2 = (01010)_F$
$8 = (1000)_2 = F_6 = (10000)_F$
$9 = (1001)_2 = (10001)_F$
$10 = (1010)_2 = (10010)_F$
$11 = (1011)_2 = (10100)_F$
$12 = (1100)_2 = (10101)_F$
$13 = (1101)_2 = F_7 = (100000)_F$
...

There is no consecutive digits equal to 1 in such representations.

Chapter 5
ARITHMETIC

**HERE – SOME INTRO

5.1 Arithmetic Operations and Their Laws

Numbers are *adjectives*—you have five apples and three oranges—but in contrast to adjectives that are purely descriptive—the red ball, the big dog—numbers can be *manipulated*, using the tools of *arithmetic*.

5.1.1 The Tools of Arithmetic

The basic tools of arithmetic reside in a small set of operations, together with two special integers that play important roles with respect to the operations. Since these entities are so tightly intertwined, we discuss them simultaneously.

Two special integers The integers zero (0) and one (1), play special roles within all four of the classes of numbers we have described.

The operations of arithmetic Arithmetic on the four classes of numbers that we have described is built upon a rather small repertoire of operations. When we say that an operation produces a number "of the same sort", we mean that it produces

- an integer result from integer arguments;
- a rational (number) result from rational (number) arguments;
- a real (number) result from real (number) arguments;
- a complex (number) result from complex (number) arguments;

The fundamental operations on numbers are, of course, familiar to the reader. Our goal in discussing them is to stress the laws that govern the operations. Along the way, we also introduce a few operations that are less familiar but no less important.

5.1.1.1 Unary (single-argument) operations

A. Negating and reciprocating numbers

i. The operation of negation:

- is a *total function* on the sets $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$. It replaces a number a by its *negative*, a number of the same sort, denoted $-a$.
- is a *partial function* on the nonnegative subsets of $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$. It replaces a number a by its negative, $-a$, whenever both a and $-a$ belong to the nonnegative subset being operated on.

Zero (0) is the unique *fixed point* of the operation, meaning that 0 is the unique number a such that $a = -a$.

ii. The operation of reciprocation:

- is a *total function* on the sets $\mathbb{Q}, \mathbb{R}, \mathbb{C}$, which replaces each number a by its *reciprocal*, a number of the same sort, denoted $1/a$ or $\frac{1}{a}$. We shall employ whichever notation enhances legibility.
- is *undefined* on every integer a except for 1.

B. Floors, ceilings, magnitudes

i. The operations of taking floors and ceilings are total operations on the sets $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$.

- The *floor* of a number a, also called *the integer part* of a, denoted $\lfloor a \rfloor$, is the largest integer that does not exceed a; i.e.,:

$$\lfloor a \rfloor \stackrel{\text{def}}{=} \max_{b \in \mathbb{N}} \Big[b \ \leq a \Big]$$

- The *ceiling* of a number a of a, denoted $\lceil a \rceil$, is the smallest integer that is not smaller than a:

$$\lceil a \rceil \stackrel{\text{def}}{=} \min_{b \in \mathbb{N}} \Big[b \ \geq a \Big]$$

Thus, the operations of taking floors and ceilings are two ways to *round* rationals and reals to their "closest" integers.

ii. The operations of taking absolute values/magnitudes: Let a be a real number. The *absolute value*, or, *magnitude*, of a, denoted $|a|$ equals either a or $-a$, whichever is positive. For a complex number a, the definition of $|a|$ is more complicated: it is a measure of a's "distance" from the "origin" complex number $0 + 0 \cdot i$. In detail:

$$|a| \ = \ \begin{cases} a & \text{if } [a \in \mathbb{R}] \text{ and } [a \geq 0] \\ -a & \text{if } [a \in \mathbb{R}] \text{ and } [a < 0] \\ \sqrt{b^2 + c^2} & \text{if } [a \in \mathbb{C}] \text{ and } [a = (b + ci)] \end{cases}$$

C. Factorials (of nonnegative integers)

The *factorial* of a nonnegative integer $n \in \mathbb{N}$, which is commonly denoted $n!$, is the function defined via the following recursion.

$$\text{FACT}(n) \;=\; \begin{cases} 1 & \text{if } n = 0 \\ n \cdot \text{FACT}(n-1) & \text{if } n > 0 \end{cases} \tag{5.1}$$

By "unwinding" the recursion in (5.1), one finds that, for all $n \in \mathbb{N}$,

$$n! \;=\; \text{FACT}(n) \;=\; n \cdot (n-1) \cdot (n-2) \cdot \cdots \cdot 2 \cdot 1 \tag{5.2}$$

A 3-step inductive argument validates this "unwinding":

1. If $n = 0$, then $\text{FACT}(n) = 1$, by definition (5.1).
2. Assume, for induction, that the expansion in (5.2) is valid for a given $k \in N$:

$$\text{FACT}(k) \;=\; k \cdot (k-1) \cdot (k-2) \cdot \cdots \cdot 2 \cdot 1$$

3. Then:

$$\begin{aligned} \text{FACT}(k+1) &= (k+1) \cdot \text{FACT}(k) && \text{by (5.1)} \\ &= (k+1) \cdot k \cdot (k-1) \cdot (k-2) \cdot \cdots \cdot 2 \cdot 1 && \text{by induction} \end{aligned}$$

5.1.1.2 Binary (two-argument) operations

A. Addition and Subtraction.

The operation of *addition* is a *total function* that replaces any two numbers a and b by a number of the same sort. The resulting number is the *sum of a and b* and is denoted $a+b$.
The operation of *subtraction* is a *total function* on the sets $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$, which replaces any two numbers a and b by a number of the same sort. The resulting number is the *difference of a and b* and is denoted $a-b$. On the nonnegative subsets of the sets $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$—such as $\mathbb{N}$, which is the largest nonnegative subset of $\mathbb{Z}$—subtraction is a *partial function*, which is defined only when $a \geq b$.

Subtraction can also be defined as follows. For any two numbers a and b, *the difference of a and b is the sum of a and the negation of b*; i.e.,

$$a - b \;=\; a + (-b)$$

The special role of 0 *under addition and subtraction.* The number 0 is the *identity* under addition and subtraction. This means that, for all numbers a,

$$a + 0 \;=\; a - 0 \;=\; a.$$

The special role of 1 *under addition and subtraction.* For any integer a, there is no integer between a and $a+1$ or between $a-1$ and a. For this reason, on the sets $\mathbb{Z}$ and $\mathbb{N}$, one often singles out the following special cases of addition and subtraction, especially in reasoning about situations that are indexed by integers. Strangely, these operations have no universally accepted notations.

- The *successor* operation is a *total function* on both $\mathbb{N}$ and $\mathbb{Z}$, which replaces an integer a by the integer $a+1$.
- The *predecessor* operation is a *total function* on $\mathbb{Z}$, which replaces an integer a by the integer $a-1$. It is a *partial function* on $\mathbb{N}$, which is defined only when the argument a is positive (so that $a-1 \in \mathbb{N}$).

The operations of addition and subtraction are said to be *mutually inverse operations* of each other because each can be used to "undo" the other:

$$a \;=\; (a+b)-b \;=\; (a-b)+b$$

B. Multiplication and Division

The operation of *multiplication* is a *total function* that replaces any two numbers a and b by a number of the same sort. The resulting number is the *product of a and b* and is denoted either $a \cdot b$ or $a \times b$. We shall usually favor the former notation, except when the latter enhances legibility.

The operation of *division* is a *partial function* on all of our sets of numbers. Given two numbers a and b, the result of dividing a by b—*when that result is defined*—is the *quotient of a by b* and is denoted by one of the following three notations: a/b, $a \div b$, $\dfrac{a}{b}$. The *quotient of a by b* is defined precisely when *both*

(1) $b \neq 0$: one can never divide by 0

and

(2) there exists a number c such that $a = b \cdot c$.

Assuming that condition (1) holds, *condition (2) always holds when a and b belong to* $\mathbb{Q}$ *or* $\mathbb{R}$ *or* $\mathbb{C}$.

Division can also be defined as follows. For any two numbers a and b, *the quotient of a and b is the product of a and the reciprocal of b* (assuming that the latter exists); i.e.,

$$a/b \;=\; a \cdot (1/b).$$

Computing reciprocals of nonzero numbers in $\mathbb{Q}$ and $\mathbb{R}$ is standard high-school level fare; computing reciprocals of nonzero numbers in $\mathbb{C}$ requires a bit of calculational algebra which we do not cover. For completeness, we note that the reciprocal of the *nonzero* complex number $a+bi \in \mathbb{C}$ is the complex number $c+di$ where

$$c \;=\; \frac{a}{a^2+b^2} \qquad \text{and} \qquad d \;=\; \frac{-b}{a^2+b^2}.$$

The special role of 1 *under multiplication and division.* The number 1 is the *identity* under the operations of multiplication and division. This means that, for all numbers a,

$$a \cdot 1 \;=\; a \cdot (1/1) \;=\; a.$$

The special role of 0 *under multiplication and division.* The number 0 is the *annihilator* under multiplication. This means that, for all numbers a

$$a \cdot 0 \;=\; 0.$$

The operations of multiplication and division are said to be *inverse operations* because, when both operations can be applied, each can be used to "undo" the other:

$$a = (a \cdot b) \div b \;=\; (a \div b) \cdot b.$$

C. Binomial coefficients and Pascal's triangle

We close our catalogue of arithmetic operations with a binary operation on[1] $\mathbb{N} \times \mathbb{N}$.

Let n and $k \leq n$ be nonnegative integers (i.e., elements of $\mathbb{N}$). The *binomial coefficient* denoted either as $\binom{n}{k}$ or as $\Delta_{n,k}$, is the number

$$\Delta_{n,k} \;=\; \binom{n}{k} \;\stackrel{\text{def}}{=}\; \frac{n!}{k!(n-k)!} \;=\; \frac{n(n-1)(n-2)\cdots(n-k+1)}{k(k-1)(k-2)\cdots 1} \tag{5.3}$$

Many of the secrets of these wonderful numbers—including the fact that they are *integers*—can be deduced from the following results.

Proposition 5.1 *For all $n,k \in \mathbb{N}$ with $k \leq n$:*

(a) *The symmetry rule:*

$$\binom{n}{k} \;=\; \binom{n}{n-k} \tag{5.4}$$

(b) *The addition rule:*

$$\binom{n}{k} \;+\; \binom{n}{k+1} \;=\; \binom{n+1}{k+1} \tag{5.5}$$

Proof. (*a*) We verify equation (5.4) by (5.3) plus the commutativity of multiplication (see Section 5.1.2),

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

[1] In advanced contexts, one encounters binomial coefficients with non-integer arguments.

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

(*b*) We verify equation (5.5) by explicitly adding the fractions exposed by (5.3):

$$\begin{aligned}\binom{n}{k} + \binom{n}{k+1} &= \frac{n!}{k!(n-k)!} + \frac{n!}{(k+1)!(n-k-1)!} \\ &= n! \cdot \frac{(k+1)+(n-k)}{(k+1)!(n-k)!} \\ &= \frac{(n+1)!}{(k+1)!(n-k)!} \\ &= \binom{n+1}{k+1}\end{aligned}$$

We have thus established both of the proposition's rules for binomial coefficients. □

Binomial coefficients are indispensable when studying myriad topics related to *counting*, such as:

- what are the relative likelihoods of various 5-card deals from a fair 52-card deck?
- What is the likelihood of observing 15 HEADs and 25 TAILs in 40 flips of a fair coin?
- What are the comparative operation-count costs of Merge-Sort and Quick-Sort when sorting n keys; cf. [28]?

We shall have a lot more to say about binomial coefficients throughout the text. Within the current chapter, we encounter binomial coefficients in Section 5.2.1; in Chapter 6, they play a prominent role in evaluating arithmetic summations (Section 6.2.1); in Chapter 8, they provide an important example of bilinear recurrences (Section 8.2); and in Chapter 9.3, they will prove indispensable in analyzing complex phenomena, by counting the way various situations (such as a royal flush in poker) can occur.

5.1.2 The Laws of Arithmetic, with Applications

Following are the basic laws of arithmetic on the reals, rationals, and reals—the ones that everyone should be able to employ cogently in rigorous argumentation.

5.1.2.1 The commutative, associative, and distributive laws

i. The commutative law. For all numbers x and y:

$$\begin{array}{ll} \textit{for addition:} & x+y \;=\; y+x \\ \textit{for multiplication:} & x\cdot y \;=\; y\cdot x \end{array}$$

ii. The associative law. For all numbers x, y, and z,

$$(x+y)+z \;=\; x+(y+z) \quad \textbf{and} \quad x\cdot(y\cdot z)(x\cdot y)\cdot z \;=\; x\cdot(y\cdot z).$$

This allows one, for instance, to write strings of additions or of multiplications without using parentheses for grouping.
iii. The distributive law. For all numbers x, y, and z,

$$x\cdot(y+z) \;=\; (x\cdot y)+(x\cdot z). \tag{5.6}$$

One commonly articulates this law as, "*Multiplication distributes over addition.*"

One of the most common uses of the distributive law reads equation (5.6) "backwards," thereby deriving a formula for *factoring* complex expressions that use both addition and multiplication.

Easily, addition does *not* distribute over multiplication; i.e., in general, $x+y\cdot z \neq (x+y)\cdot(x+z)$. Hence, when we see "$x+y\cdot z$", we know that the multiplication is performed before the addition. In other words, *Multiplication takes priority over addition.* This priority permits us to write the righthand side of (5.6) without parentheses, as in

$$x\cdot(y+z) \;=\; x\cdot y+x\cdot z.$$

Via multiple invocations of the preceding laws, we can derive a recipe for multiplying complicated expressions. We illustrate this via the "simplest" complicated expression, $(a+b)\cdot(c+d)$.

Proposition 5.2 *For all numbers a,b,c,d:*

$$(a+b)\cdot(c+d) \;=\; a\cdot c+a\cdot d+b\cdot c+b\cdot d \tag{5.7}$$

Proof. Note first that because multiplication takes priority over addition, the absence of parentheses in expressions such as (5.2) does not jeopardize unambiguity. Our proof of the proposition invokes the laws we have just enunciated multiple times.

$$\begin{array}{rll} (a+b)\cdot(c+d) &= (a+b)\cdot c \;+\; (a+b)\cdot d & \text{distributive law} \\ &= c\cdot(a+b) \;+\; d\cdot(a+b) & \text{commutativity of multiplication } (2\times) \\ &= c\cdot a+c\cdot b+d\cdot a+d\cdot b & \text{distributive law } (2\times) \\ &= a\cdot c+b\cdot c+a\cdot d+b\cdot d & \text{commutativity of multiplication } (4\times) \\ &= a\cdot c+a\cdot d+b\cdot c+b\cdot d & \text{commutativity of addition} \end{array}$$

□

We close our short survey of the laws of arithmetic with the following important two-part law.

- *The law of inverses*.
  - Every number x has an *additive inverse*, i.e., a number y such that $x+y=0$. This inverse is x's *negative* $-x$.
  - Every *nonzero* number $x \neq 0$ has a *multiplicative inverse*, i.e., a number y such that $x \cdot y = 1$. This inverse is x's *reciprocal*, $1/x$.

We close this section with another of our "fun" propositions.

5.1.2.2 A fun result: A "trick" for squaring some integers

Proposition 5.3 *Let n be any number that has a 2-digit decimal of the form $\delta 5$, where $\delta \in \{0,1,2,3,4,5,6,7,8,9\}$, so that*

$$n \;=\; 10 \cdot \delta + 5$$

Then

$$n^2 \;=\; 100 \cdot \delta \cdot (\delta + 1) + 25.$$

In other words, one obtains a base-10 numeral for n^2 by multiplying δ by $\delta + 1$ and appending 25 to the product.

Examples of Proposition 5.3 include $25^2 = 625$ (because $2 \cdot 3 = 6$) and $75^2 = 5625$ (because $7 \cdot 8 = 56$).

Proof. (for general δ). We invoke Proposition 5.2 and the distributive law.

$$\begin{array}{ll} n^2 = (10 \cdot \delta + 5)^2 & \text{Given} \\ \quad = 100 \cdot \delta^2 \;+\; 100 \cdot delta \;+\; 25 & \text{the proposition} \\ \quad = 100 \cdot (\delta^2 \;+\; \delta) \;+\; 25 & \text{factoring: distributive law} \\ \quad = 100 \cdot \delta \cdot (\delta + 1) \;+\; 25 & \text{factoring: distributive law} \end{array}$$

□

5.1.3 Rational Arithmetic: A Specialized Computational Exercise

In Section 4.4 we defined the rational numbers and reviewed why they were needed to compensate for the general lack of multiplicative inverses within the integers. But we did not review how to perform arithmetic on the elements of the set $\mathbb{Q}$. We correct this shortcoming now. Of course, the reader will have encountered rational arithmetic long ago—but we are now reviewing the topic in order to provide the reader with a set of worthwhile exercise to reinforce the mathematical thinking whose presentation is our main goal.

The rational numbers build their rules for arithmetic upon the corresponding rules for integers. For all p/q and r/s in $\mathbb{Q}$:

Addition:	$\frac{p}{q}+\frac{r}{s}=\frac{p\cdot s+r\cdot q}{q\cdot s}$
Subtraction:	$\frac{p}{q}+\frac{r}{s}=\frac{p}{q}+\frac{(-r)}{s}$
Multiplication:	$\frac{p}{q}\cdot\frac{r}{s}=\frac{p\cdot r}{r\cdot s}$
Division:	$\frac{p}{q}\div\frac{r}{s}=\frac{p}{q}\cdot\frac{s}{r}$

It is worth verifying that rational arithmetic as thus defined behaves in the required manner; in particular that rational arithmetic:

- works correctly when the argument rational numbers are, in fact, integers, i.e., when $q=s=1$ in the preceding table.
- treats the number 0 appropriately, i.e., as an additive identity and a multiplicative annihilator; cf., Sections 5.1.1 and 5.1.2.
- obeys the required laws; cf., Section 5.1.2.
  Verifying the distributivity of rational multiplication over rational addition will be a particularly valuable exercise because of the required amount of manipulation.

5.2 Basic Algebraic Concepts and Their Manipulations

5.2.1 Powers and polynomials

5.2.1.1 Raising a number to a power.

A conceptually powerful notational construct is the operation of *raising a number to a power:* For real numbers a and b, the *bth power* of a, denoted a^b is defined by the system of equations

$$\begin{aligned} &\text{for all numbers } a>0 \quad a^0=1 \\ &\text{for all numbers } a,b,c \quad a^b\cdot a^c=a^{b+c}. \end{aligned} \tag{5.8}$$

This deceptively simple definition has myriad consequences which we often take for granted.

- For all numbers $a>0$, the number $a^0=1$.

This follows (via cancellation) from (5.8) via the fact that

$$a^b \cdot a^0 \;=\; a^{b+0} \;=\; a^b \;=\; a^b \cdot 1.$$

- For all numbers $a > 0$, the number $a^{1/2}$ is the *square root* of a, i.e., $a^{1/2}$ is the (unique, via cancellation) number b such that $b^2 = a$. Another common notation for The number $a^{1/2}$ is $\sqrt{a}$.
  This follows from (5.8) via the fact that

$$a \;=\; a^1 \;=\; a^{(1/2)+(1/2)} \;=\; a^{1/2} \cdot a^{1/2} \;=\; \left(a^{1/2}\right)^2.$$

- For all numbers $a > 0$ and b, the number a^{-b} is the *multiplicative inverse* of a^b, meaning that $a^b \cdot a^{-b} = 1$
  This follows from (5.8) via the fact that

$$a^b \cdot a^{-b} \;=\; a^{(b+(-b))} \;=\; a^0 \;=\; 1$$

When the power b is a positive integer, then definition (5.8) can be cast in the following attractive inductive form:

$$\begin{array}{ll} \text{for all numbers } a > 0 & a^0 = 1 \\ \text{for all numbers } a \text{ and integers } b & a^{b+1} = a \cdot a^b. \end{array} \tag{5.9}$$

Summing up, we now know about powers that are integral or fractional, positive, zero, or negative

5.3 Polynomials and Their Roots

We want students to master the notions of polynomials and their associated notions, such as degrees and coefficients, and computations therewith, including polynomial summation and multiplication. While polynomial multiplication is often considered "non-elementary", it must be mastered in order to fully understand positional number systems; it is also essential, e.g., when discussing a range of topics relating to, say, fault tolerance and encryption).

**DEFINE "degree" and "root"

5.3.1 ⊕ The General Unsolvability of the Problem

**Briefly discuss Hilbert's Tenth Problem — Julia Robinson, Martin Davis, Yuri Matiyasevich, Hilary Putnam and Julia Robinson

This problem suggests that seeking roots of general polynomials will be exceedingly hard, and often impossible.

5.3.2 Univariate Polynomials and Their Roots

"finding the roots of polynomial $P(x)$" is often called "solving polynomial $P(x)$"

We focus throughout on univariate polynomials with real coefficients

5.3.2.1 Every degree-d polynomial has d roots

Finding roots of *univariate* polynomials is always possible.

Theorem 5.1 (The Fundamental Theorem of Algebra). *Every degree-d univariate polynolmial with complex coefficients has d complex roots.*

There are proofs of Theorem 5.1 that are constructive in the followig sense. Given a degree-d polynomial $P(x)$, the proof determines a disk D in space such all d roots of $P(x)$ reside in disk D.

Milestones in the quest for roots

Al-Khwarizmi, Descartes

5.3.2.2 Solving polynomials by radicals

The problem of *solving* arbitrary degree-d polynomials—i.e., of discovering their d roots, as promised by Theorem 5.1—is computationally very complex, even for moderately low degrees. (This assertion can be made mathematically precise, but the required notions are beyond the scope of this text.) For univariate polynomials of low degree, there do exist computationally feasible root-finding algorithms. Indeed, for polynomials of *very* low degree—specifically, degrees 2, 3 and 4—there actually exist "simple" *formulas* that specify the polynomial's roots. The word "simple" is used in a technical sense here: it refers to a formula that can be constructed using the following algebraic operations: adding/subtracting two quantities, multiplying/dividing two quantities, and raising a quantity to a rational power. Because the last of these operations is often expressed by using a *radical sign*, rather than an exponent—as when we write "$\sqrt{x}$" for "$x^{1/2}$"—these formulas are often referred to as *solution by radicals*.

A. Galois theory: the *unsolvability* of the quintic by radicals

B. Solving *quadratic* and *cubic* polynomials by radicals

This section is devoted to deriving the *quadratic* formula—the one that specifies the roots of any degree-2 *(quadratic)* polynomial—and the *cubic* formula—the one that specifies the roots of any degree-3 *(cubic)* polynomial. We shall observe that the cubic formula is so onerous calculationally that it is seldom actually written out; it is instead specified *algorithmically*. The *quartic* formula—the one that specifies the roots of any degree-4 *(quartic)* polynomial—is so complex that it is virtually never written out. The courageous reader can attack the quartic formula as an exercise, using the conceptual techniques we derive here.

i. Solving quadratic polynomials by radicals
We derive the *quadratic formula*, which solves an arbitrary quadratic polynomial with real coefficients:

$$P(x) \;=\; ax^2 \;+\; bx \;+\; c \quad \text{where } \; b,c \in \mathbb{R};\; a \in \mathbb{R} \setminus \{0\} \tag{5.10}$$

While the formula and its derivation are specialized to the structure of quadratic polynomials, several aspects of the derivation can be extrapolated to polynomials of higher degree. The formula that we derive is announced in the following proposition.

Proposition 5.4 *The two roots, x_1 and x_2, of the generic quadratic polynomial (5.10) are:*

$$\begin{aligned} x_1 &= \frac{-b \;+\; \sqrt{b^2 - 4ac}}{2a} \\ x_2 &= \frac{-b \;-\; \sqrt{b^2 - 4ac}}{2a}. \end{aligned} \tag{5.11}$$

Proof. We find the roots of $P(x)$ by solving the polynomial equation $P(x) = 0$. We simplify our task by dividing both sides of this equation by a; easily, this does not impact the two solutions we seek. We thereby reduce the root-finding problem to the solution of the equation

$$x^2 \;+\; \frac{b}{a}x \;=\; -\frac{c}{a}. \tag{5.12}$$

The technique of *completing the square* gives us an easy path to solving equation (5.12). This technique involves adding to both sides of the equation a variable-free expression E that turns the lefthand expression into a perfect square. In the current instance, the expression

$$E \;=\; \frac{b^2}{4a^2}$$

does the job, because

$$x^2 + \frac{b}{a}x + E = x^2 + \frac{b}{a}x + \frac{b^2}{4a^2} = \left(x + \frac{b}{2a}\right)^2$$

We have thereby converted equation (5.12) to the equation

$$\left(x + \frac{b}{2a}\right)^2 = \frac{b^2}{4a^2} - \frac{c}{a} = \frac{b^2 - 4ac}{4a^2}. \tag{5.13}$$

Elementary calculation on equation (5.13) identifies $P(x)$'s two roots as the values x_1 and x_2 specified in (5.11). □

Using a common shorthand, the expressions for x_1 and x_2 in (5.11) are often abbreviated by using the operator $\pm$, for "plus or minus". The quadratic formula is then written:

$$x = \frac{1}{2a}\left(-b \pm \sqrt{b^2 - 4ac}\right).$$

The reader can verify that our proof essentially proceeds by replacing $P(x)$'s variable x with the variable

$$u = x + \frac{b}{2a}.$$

This replacement streamlines the process of completing the square and finding the solutions x_1 and x_2. We presented the more elementary proof to let the reader see the solution process proceed step by step. As we turn now to the clerically more complex solution of the cubic polynomial, we get around some of the calculational complexity by employing the variable-substitution strategem.

ii. Solving cubic polynomials by radicals

We derive a formula that *solves* an arbitrary cubic polynomial with real coefficients:

$$P(x) = ax^3 + bx^2 + cx + d \quad \text{where } b,c,d \in \mathbb{R};\ a \in \mathbb{R} \setminus \{0\} \tag{5.14}$$

Although the so-called *cubic formula* that we derive, and its derivation, are specialized to the structure of degree-3 polynomials, the reader will recognize several aspects of the derivation that are akin to our derivation of the quadratic formula. Because the cubic formula is so complex in form, we present the formula's proof/derivation *before* presenting the formula.

Proof. (Derivation of the cubic formula)

Step 1. Convert the problem to solving a *monic* cubic polynomial.

We convert the generic cubic polynomial $P(x)$ of (5.14) to the monic cubic polynomial that has the same roots as $P(x)$.

$$P^{(1)}(x) = x^3 + Bx^2 + Cx + D \tag{5.15}$$
$$\text{where} \quad B = \frac{b}{a};\ C = \frac{c}{a};\ D = \frac{d}{a}$$

Monic polynomials lead to somewhat simpler calculation

It should be clear that the polynomials $P(x)$ and $P^{(1)}(x)$ have the same roots. We should include this as a simple exercise.

Step 2. Convert $P^{(1)}(x)$ to *reduced form*.
When we make the transformation of variable

$$y = x + \frac{B}{3} \tag{5.16}$$

in (5.15), we convert polynomial $P^{(1)}(x)$ in variable x into the polynomial following in variable y:

$$\begin{aligned} P^{(2)}(y) &= \left(y - \frac{B}{3}\right)^3 + B\left(y - \frac{B}{3}\right)^2 + C\left(y - \frac{B}{3}\right) + D \\ &= y^3 + \left(\frac{B^2}{9} - \frac{2B^2}{3} + C\right)y + \left(\frac{2B^3}{27} - \frac{BC}{3} + D\right) \end{aligned} \tag{5.17}$$

Cubic polynomials that lack a quadratic term—i.e., a term involving y^2—are said to be in *reduced form*. For simplicity, we rewrite $P^{(2)}(y)$, which clearly is in reduced form, as

$$P^{(2)}(y) = y^3 + Ey + F \tag{5.18}$$

where

$$\begin{aligned} E &= \frac{B^2}{9} - \frac{2B^2}{3} + C \\ F &= \frac{2B^3}{27} - \frac{BC}{3} + D \end{aligned}$$

Step 3. Convert $P^{(2)}(y)$ to its *associated* quadratic polynomial.
We next apply a transformation attributed to the 16th-century French mathematician François Viète (often referred to by his Latinized name, Franciscus Vieta) ; see [40]. Vieta's transformation converts $P^{(2)}(y)$ to a quadratic polynomial by means of the variable-substitution

$$y = z - \frac{E}{3z} \tag{5.19}$$

in (5.18). We thereby obtain (after calculations involving several cancellations) an expression

$$\begin{aligned} P^{(3)}(z) &= \left(z - \frac{E}{3z}\right)^3 + E\left(z - \frac{E}{3z}\right) + F \\ &= z^3 - \frac{E^3}{27z^3} + F \end{aligned} \tag{5.20}$$

Clearly, $P^{(3)}(z)$ is not a polynomial in z, but it is a valuable stepping stone because the function $P^{(3)}(z)$ vanishes—i.e., $P^{(3)}(z) = 0$—precisely when the following polynomial (in the "variable" z^3) vanishes.

$$P^{(4)}(z^3) \;=\; (z^3)^2 \;+\; (z^3)F \;-\; \frac{E^3}{27}.$$

We wrote both instances of z^3 in the expression for $P^{(4)}(z^3)$ within parentheses to facilitate the view of $P^{(4)}(z^3)$ as a (quadratic) polynomial in the "variable" z^3. The quadratic formula (5.11) provides us with two roots for $P^{(4)}$, which we express as the following two values for z^3 (abbreviated via the shorthand operator $\pm$).

$$(z^3) \;=\; -\frac{F}{2} \;\pm\; \sqrt{\frac{F^2}{4} + \frac{E^3}{27}} \tag{5.21}$$

We can now derive all solutions for the variable z in equation (5.21) via back-substitution in transformation (5.19). But completing this derivation requires a bit of background.

Theorem 5.1 assures us that the polynomial $P^{(2)}(z)$ has three roots. In order to compute these roots, we invoke a truly remarkable result that is known as *Euler's formula*, in honor of its discoverer, the much-traveled 18th-century mathematician Leonhard Euler. This result/formula exposes a fundamental relationship among:

- the imaginary unit i;
- the ratio of the circumference of a circle to its radius, $\pi \;=\; 3.141592653\cdots$;
- the base of natural logarithms, Euler's constant $e \;=\; 2.718281828\cdots$.

Theorem 5.2 (Euler's formula).

$$e^{i\pi} \;=\; -1.$$

Back to equation (5.21): Theorem 5.1 tells us that within the complex number system $\mathbb{C}$, the cubic polynomial

$$u^3 \;-\; 1$$

has three distinct roots. These numbers are known as the *primitive* 3*rd roots of unity* and are denoted ω^0, ω^1, and ω^2. Using Theorem 5.2, we can provide explicit values for these numbers, namely:

$$\omega^0 \;=\; 1; \quad \omega^1 \;=\; e^{2i\pi/3}; \quad \omega^2 \;=\; e^{4i\pi/3}.$$

When we unite the abbreviated double equation (5.21) for z^3 with Euler's formula (Theorem 5.2), we discover *six* solutions for the variable z, namely,

$$\begin{aligned}
z_1 &= \omega^0 \cdot \left(-\frac{F}{2} + \sqrt{\frac{F^2}{4} + \frac{E^3}{27}}\right)^{1/3} & z_2 &= \omega^0 \cdot \left(-\frac{F}{2} - \sqrt{\frac{F^2}{4} + \frac{E^3}{27}}\right)^{1/3} \\
z_3 &= \omega^1 \cdot \left(-\frac{F}{2} + \sqrt{\frac{F^2}{4} + \frac{E^3}{27}}\right)^{1/3} & z_4 &= \omega^1 \cdot \left(-\frac{F}{2} - \sqrt{\frac{F^2}{4} + \frac{E^3}{27}}\right)^{1/3} \\
z_5 &= \omega^2 \cdot \left(-\frac{F}{2} + \sqrt{\frac{F^2}{4} + \frac{E^3}{27}}\right)^{1/3} & z_6 &= \omega^2 \cdot \left(-\frac{F}{2} - \sqrt{\frac{F^2}{4} + \frac{E^3}{27}}\right)^{1/3}
\end{aligned} \tag{5.22}$$

The algorithmically interesting portion of the process of solving cubics by radicals is now complete. The remainder of the process consists of "reversing" the two transformations, (5.19) and (5.16), that have taken us from the problem of solving a polynomial in x to the problem of solving a polynomial in z. The calculations that embody this reverse transformation are onerous when solved symbolically, so we make do with some exercises in which the reader will solve numerical instances. The most interesting and noteworthy feature of these exercise will be the observation of "collapsing" of intermediate expressions, whose impact is to leave us with with only *three* solutions for x—which isthe number promised by Theorem 5.1—rather than the six solutions that the array (5.22) of z-values would lead one to expect.

While the promise of a visually appealing cubic analogue of the quadratic formula (5.11) is appealing, the actual cubic formula is so complex visually that it offers no important insights. The curious reader can find renditions of the formulas on the web. □

I think that solving specific examples of cubics via radicals, although specialized, does involve useful skills for manipulating polynomials. Therefore, I propose to stop the general case here and leave a few exercises. WHAT DO YOU THINK?

Because your geometric solution below involves just a single special quadratic — and one that seems not to have other special interest — I propose that we include it in some form of ENRICHMENT section.

Add here a small introduction about solving polynomial of degree 2 from El Kwharizmi, or remove it and put it as an exercice.

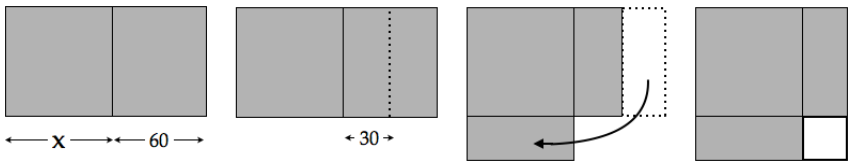


Fig. 5.1 Solving $x^2 + x = 45$. The idea of the proof is to represent the left hand side by the square x^2 beside a rectangle $60 \times x$. Then, split the right rectangle into two equal parts and move one part a the bottom of the left square. The final figure shows the whole square whose surface is equal to 45 plus the surface of the white square whose surface is equal to 30×30. In base 60, this is 15. $45 + 15 = 60$, thus, the big square is the unit square, its side is 60. Thus, the length of the initial square is equal to $60 - 30 = 30$.

5.3.3 Bivariate Polynomials

5.3.3.1 The Binomial Theorem.

Perhaps the simplest bivariate polynomials are the ones in the following family.

$$\text{For } n \in \mathbb{N}^+, \quad P_n(x,y) \stackrel{\text{def}}{=} (x+y)^n. \tag{5.23}$$

There are lessons to be learned from the structure of these polynomials, so let us begin to expand them using the arithmetic techniques we have learned earlier.

$$\begin{aligned}
P_1(x,y) &= (x+y)^1 = x+y \\
P_2(x,y) &= (x+y)^2 = (x+y)\cdot(x+y) \\
&= x^2+2xy+y^2 \\
P_3(x,y) &= (x+y)^3 = (x+y)\cdot(x^2+2xy+y^2) \\
&= (x^3+2x^2y+xy^2)+(x^2y+2xy^2+y^3) \\
&= x^3+3x^2y+3xy^2+y^3
\end{aligned}$$

Let us stop to review what we are seeing. We have remarked before that doing mathematics can sometimes involve a wonderfully exciting (quite sophisticated) pattern-matching game. So, let us pattern-match!

1. The coefficients of the expanded $P_1(x,y)$ are $\langle 1,1\rangle$.
2. The coefficients of the expanded $P_2(x,y)$ are $\langle 1,2,1\rangle$.
3. The coefficients of the expanded $P_3(x,y)$ are $\langle 1,3,3,1\rangle$.

There is a pattern emerging here. Can you spot it? Where have we seen a pattern of tuples that begins in the same manner? As a rather broad hint, look at Fig. 8.2! Could the coefficients of each P_n possibly be the successive binomial coefficients

$$\binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n-1}, \binom{n}{n}$$

Let us use induction to explore this possibility by expanding a generic P_n with symbolic "dummy" coefficients and see what this says about P_{n+1}. To this end, let $a_{n,n-r}$ denote the coefficient of $x^{n-r}y^r$ in the expansion of $P_n(x,y)$. Using our "dummy" coefficients, we have

$$\begin{aligned}
P_n(x,y) = x^n + \cdots + a_{n,n-r}x^{n-r}y^r + a_{n,n-r-1}x^{n-r-1}y^{r+1} \\
+ a_{n,n-r-2}x^{n-r-2}y^{r+2} + \cdots + y^n
\end{aligned}$$

Continuing with this symbolic evaluation, we have:

$$\begin{aligned}
x\cdot P_n(x,y) = x^{n+1} + \cdots + a_{n,n-r}x^{n-r+1}y^r + a_{n,n-r-1}x^{n-r}y^{r+1} \\
+ a_{n,n-r-2}x^{n-r-1}y^{r+2} + \cdots + xy^n
\end{aligned} \tag{5.24}$$

and

$$y \cdot P_n(x,y) \;=\; x^n y \;+\; \cdots \;+\; a_{n,n-r}x^{n-r}y^{r+1} \;+\; a_{n,n-r-1}x^{n-r-1}y^{r+2} \\ +\; a_{n,n-r-2}x^{n-r-2}y^{r+3} \;+\; \cdots \;+\; y^{n+1} \tag{5.25}$$

Because

$$P_{n+1}(x+y) \;=\; (x+y)\cdot P_n(x,y) \;=\; x \cdot P_n(x,y) \;+\; y \cdot P_n(x,y),$$

the "dummy" coefficient $a_{n-r+1,r}$ of $x^{n-r+1}y^r$ in $P_{n+1}(x+y)$ is the sum of the following coefficients in $P_n(x,y)$:

• the coefficient $a_{n,n-r}$ of $x^{n-r}y^r$ and • the coefficient $a_{n,n-r+1}$ of $x^{n-r+1}y^{r-1}$

By induction, then, for all $n, r \in \mathbb{N}$ with $r \leq n$,

$$a_{n,r} + a_{n,r+1} \;=\; a_{n+1,r+1}$$

Combining this equation with the observed initial conditions

$$a_{1,0} \;=\; a_{1,1} \;=\; 1$$

we see that each coefficient $a_{n,r}$ is actually the binomial coefficient $\binom{n}{r}$.

The preceding observation is attributed to the renowned English mathematician/physicist Isaac Newton and is enshrined in Newton's famous *Binomial Theorem*. In fact, the calculations preceding the observation constitute a proof of this seminal result.

Theorem 5.3 (The Binomial Theorem). *For all $n \in \mathbb{N}$,*

$$(x+y)^n \;=\; \sum_{i=0}^{n} \binom{n}{i} x^{n-i}y^i. \tag{5.26}$$

5.4 Exponential and Logarithmic Functions

This section introduces the fundamentals of two extremely important classes of functions which are functional inverses of each other, in the following sense. Functions f and g are *functional inverses* of each other if for all arguments x

$$f(g(x)) \;=\; x. \tag{5.27}$$

5.4.1 Basic definitions

5.4.1.1 Exponential functions

.

A function f is *exponential* if there is a positive number b such that, for all x,

$$f(x) \;=\; b^x. \tag{5.28}$$

The number b is the *base* of $f(x)$. The basic arithmetic properties of exponential functions are derivable from (5.8), so we leave these details to the reader and turn immediately to the functional inverses of exponential functions..

5.4.1.2 Logarithmic functions

.

Given an integer $b > 1$ (mnemonic for "base"), the *base-b logarithm* of a real number $a > 0$ is denoted $\log_b a$ and defined by the equation

$$a \;=\; b^{\log_b a}. \tag{5.29}$$

Logarithms are partial functions: $\log_b a$ is not defined for non-positive arguments.

The base $b = 2$ is so prominent in the contexts of computation theory and information theory that we commonly invoke one of two special notations for $\log_2 a$: (1) we often elide the base-2 subscript and write $\log a$; (2) we employ the specialized notation $\ln a$. Notationally:

$$\log_2 a \;\stackrel{\text{def}}{=}\; \log a \;\stackrel{\text{def}}{=}\; \ln a$$

We leave to the reader the easy verification, from (5.29), that the *base-b logarithmic function*, defined by

$$f(x) \;=\; \log_b x \tag{5.30}$$

is the functional inverse of the base-b exponential function.

5.4.2 Fun facts about exponentials and logarithms

Definition (5.29) exposes and—even more importantly—explains myriad facts about logarithms that we often take for granted.

Proposition 5.5 *For any base $b > 1$, for all numbers $x > 0$, $y > 0$,*

$$\log_b(x \cdot y) \;=\; \log_b x \;+\; \log_b y$$

Proof. Definition (5.29) tells us that $x = b^{\log_b x}$ and $y = b^{\log_b y}$. Therefore,

$$x \cdot y \;=\; b^{\log_b x} \cdot b^{\log_b y} \;=\; b^{\log_b x \,+\, \log_b y},$$

by the laws of powers. Taking base-b logarithms of the first and last terms in the chain yields the claimed equation. □

Many students believe that the following result is a *convention* rather than a consequence of the basic definitions. *The logarithm of* 1 *to any base is* 0.

Proposition 5.6 *For any base $b > 1$,*

$$\log_b 1 \;=\; 0$$

Proof. We note the following chain of equalities.

$$b^{\log_b x} \;=\; b^{\log_b (x \cdot 1)} \;=\; b^{(\log_b x) + (\log_b 1)} \;=\; b^{\log_b x} \cdot b^{\log_b 1}$$

Hence, $b^{\log_b 1} \;=\; 1$. If $\log_b 1$ did not equal 0, then $b^{\log_b 1}$ would exceed 1. □

Proposition 5.7 *For all bases $b > 1$ and all numbers x, y,*

$$x^{\log_b y} \;=\; y^{\log_b x}$$

Proof. We invoke (5.29) twice to remark that

$$\left[x^{\log_b y} \;=\; b^{(\log_b x)\cdot(\log_b y)}\right] \quad \text{and} \quad \left[y^{\log_b x} \;=\; b^{(\log_b y)\cdot(\log_b x)}\right]$$

The commutativity of addition completes the verification. □

Proposition 5.8 *For any base $b > 1$,*

$$\log_b (1/x) \;=\; -\log_b x$$

Proof. This follows from the fact that $\log_b 1 = 0$, coupled with the product law for logarithms.

$$\log_b x + \log_b (1/x) \;=\; \log_b (x \cdot (1/x)) \;=\; \log_b 1 \;=\; 0$$

□

Proposition 5.9 *For any bases $a, b > 1$,*

$$\log_b x \;=\; (\log_b a) \cdot (\log_a x)\,. \tag{5.31}$$

Proof. We begin by noting that, by definition, Note that

$$x \;=\; b^{\log_b x} \;=\; a^{\log_a x}. \tag{5.32}$$

Let us take the base-b logarithm of the second and third expressions in (5.32) and then invoke the product law for logarithms. From the second expression in (5.32), we find that

$$\log_b \left(b^{\log_b x} \right) \;=\; \log_b x. \tag{5.33}$$

From the third expression in (5.32), we find that

$$\log_b \left(a^{\log_a x} \right) \;=\; (\log_b a) \cdot (\log_a x)\,. \tag{5.34}$$

We know from (5.32) that the righthand expressions in (5.33) and (5.34) are equal, whence (5.31). □

If we set $x = b$ in (5.31), then we find the following marvelous equation.

Proposition 5.10 *For any integers $a, b > 1$,*

$$(\log_b a) \cdot (\log_a b) \;=\; 1 \qquad \textit{or, equivalently,} \qquad \log_b a \;=\; \frac{1}{\log_a b}. \tag{5.35}$$

5.4.3 Exponentials and logarithms within information theory

The student should recognize and be able to reason about the following facts. If one has an alphabet of a letters/symbols and must provide distinct string-label "names" for n items, then at least one string-name must have length no shorter than $\lceil \log_a n \rceil$.

Proposition 5.11 *Say that one must assign distinct labels to n items, via strings over an alphabet of a letters. Then at least one string-label must have length no shorter than $\lceil \log_a n \rceil$.*

Proof. Let *Sigma* be an alphabet of a letters/symbols. For each integer $k \geq 0$ (i.e., for each $k \in \mathbb{N}$), let $\Sigma^{(k)}$ denote the set of all length-k strings over Σ. The bound of Proposition 5.11 follows by counting the number of strings of various lengths over Σ, because each such string can label at most one item. Let us, therefore, inductively evaluate the cardinality $|\Sigma^{(k)}|$ of each set $\Sigma^{(k)}$.

- $|\Sigma^{(0)}| = 1$
  This is because the null-string ε is the unique string in $\Sigma^{(0)}$, i.e., $\Sigma^{(0)} = \{\varepsilon\}$.
- $|\Sigma^{(k+1)}| = |\mathit{Sigma}| \cdot |\Sigma^{(k)}|$.
  This reckoning follows from the following recipe for creating all strings over Σ of length $k+1$ from all strings of length k.

  $$\Sigma^{(k+1)} \;=\; \{\sigma x \mid \sigma \in \Sigma, x \in \Sigma^{(k)}\}$$

  This recipe is correct because

  – Each string in $\Sigma^{(k+1)}$, as constructed, has length $k+1$.
    This is because the recipe adds a single symbol to a length-k string.

- For each string $x \in \Sigma^{(k)}$, there are $|\Sigma|$ distinct strings in $\Sigma^{(k+1)}$, as constructed. This is because each string in $\Sigma^{(k+1)}$ begins with a distinct symbol from Σ.
- $\Sigma^{(k+1)}$, as constructed, contains all strings of length $k+1$ over Σ. This is because for each $\sigma \in \Sigma$ and each $x \in \Sigma^{(k)}$, the string σx is in $\Sigma^{(k+1)}$, as constructed.

We thus have the following recurrence.

$$|\Sigma^{(0)}| = 1$$
$$|\Sigma^{(k+1)}| = |\Sigma| \cdot |\Sigma^{(k)}| \quad \text{for } k \geq 0$$

Using the Master Theorem of Section 8.1.1, we thus find explicitly that:
For each $\ell \in \mathbb{N}$,

$$|\Sigma^{(\ell)}| = \frac{|\Sigma|^{\ell+1} - |\Sigma|}{|\Sigma| - 1} \leq c \cdot |\Sigma|^{\ell}$$

for some constant c. In order for this quantity to reach the value $n \in \mathbb{N}$, we must have

$$\ell > d \cdot \log_{|\Sigma|} n$$

for some small constant d. □

**HERE
Focus on Say, inductively, that there are ℓ_k

Proposition 5.12 *The number of distinct strings of length k over an alphabet of a letters is a^k.*

Proof. Focus on the generic a-letter alphabet $\mathscr{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_a\}$. We argue by induction on k.
Bases. The induction we develop can start either at $k = 0$ or at $k = 1$. Some people are willing to recognize the *null word*, which contains no letters, hence has length 0; others insist that the status "word" can be enjoyed only by non-null strings. This is purely a matter of taste.

At any rate, everyone agrees that, if you are willing to countenance the null word over the alphabet, there is only $a^0 = 1$ such word; this is the case $k = 0$ of the proposition. And, everyone agrees that, if you insist on non-null words, then there are $a^1 = a$ such words, one for each symbol in $\mathscr{A}$; this is the case $k = 1$ of the proposition.
The inductive hypothesis. Say that for all word-lengths k up through n, there are a^k distinct words of length k over $\mathscr{A}$.
Extending the induction. Take each length-n word w over $\mathscr{A}$, and append each of $\mathscr{A}$'s a symbols to w. One thereby creates a, obviously distinct, words from w, namely, $w\alpha_1, w\alpha_2, \ldots, w\alpha_a$. We have thus created a^{n+1} distinct length-$(n+1)$ words over $\mathscr{A}$ from $\mathscr{A}$'s a^n distinct length-n words. □

5.5 Useful Nonalgebraic Notions

5.5.1 Nonalgebraic Notions Involving Numbers

If the intended curriculum will approach more sophisticated application areas such as robotics or data science or information retrieval or data mining (of course, at levels consistent with the students' preparation), then one would do well to insist on familiarity with notions such as:

Chapter 6
SUMMATION

6.1 Introduction

The operation of *summation*—adding up aggregates of numbers—is of fundamental importance in the world of digital computing. While we humans are able to deal handily with abstractions such as "smoothness" and "continuity", we must employ sophisticated *discretizations* of these concepts in order to enlist the aid of digital computers in dealing with such abstractions. Summations provide a very useful discretization of "continuous" or "smooth" phenomena that are typically dealt with by humans with the aid of the (differential and integral) calculus that was invented by Newton and Leibniz for such dealings.

This chapter is dedicated to exploring how to employ summations as a computational tool. We deal throughout with *series*, *i.e.*, (possibly infinite) sequences of numbers

$$a_1, a_2, \ldots$$

whose sum

$$a_1 + a_2 + \cdots \tag{6.1}$$

is of interest.

$$\approx\approx\approx\approx\approx\approx\approx\approx\approx\approx$$

Of course, when we deal with *infinite* series, wherein there are infinitely many numbers a_i, we must address the question of whether the sum (6.1) exists as a finite number. For some infinite series the sum *does* exist as a finite number, as with the well-known sum

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots + \frac{1}{2^k} + \frac{1}{2^{k+1}} + \cdots = 2 \tag{6.2}$$

Such an infinite series is said to *converge*.

But sometimes an infinite series *does not* have a finite sum. This is true, for instance, with the well-known *harmonic* series

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{k} + \frac{1}{k+1} + \cdots \tag{6.3}$$

As more and more terms are added, the accumulated sum eventually exceeds every number. Such an infinite series is said to *diverge*.

≈≈≈≈≈≈≈≈≈≈

Sums such as (6.2) and (6.3) illustrate some of the complexity of dealing with infinite entities. Most obviously, as we have just remarked, when the sums are infinite, some of them have finite sums while others do not. Even more subtle, the series that *do* have (finite) sums illustrate the unintuitive fact that, sometimes finite objects or entities—such as the integer 2 in (6.2)—have infinite "names"—the infinite series in this example. Lots to think about!

≈≈≈≈≈≈≈≈≈≈

The complexity of the concept of convergence has been recognized in various forms for more than 25 centuries. Several charming and familiar examples appear in the paradoxes attributed to Zeno of Elea. In his *Paradox of Achilles and the Tortoise*, for instance, Zeno appears at first glance to prove that all motion is illusory. In this story, the slow-footed Tortoise (T) tries to convince the speedy Achilles (A) of the futility of trying to win any race in which A gives T even the smallest head start. As long as T is ahead of A, says T, every time A traverses half the distance between the competitors, T will respond by moving a bit further ahead. Thereby, T will always be a positive distance ahead of A, so that A can *never* catch T. A similar "argument" demonstrates that an arrow shot at you by an adversary can never reach you, as long as you continually move away from the archer. *DO NOT TRY THIS AT HOME!!*

The notion of *infinitesimals*, which explains the fallacy of assertions such as the Tortoise's, were not well understood until a few hundred years ago. This notion plays a huge role in modern mathematics, underlying such foundational concepts as *limits* and *continuity* (of functions).

The general topic of the convergence or divergence of infinite series is beyond the scope of this text. It is a fascinating subject for advanced study.

≈≈≈≈≈≈≈≈≈≈

Toward the end of guiding the reader through the forest of abstractions and operations and techniques associated with summations, we categorize the targets of our discussions in three ways.

1. We study a number of *fundamental sums* that have intrinsic interest. Examples of this topic category include *arithmetic summations*, *geometric summations*, and *mathematically "smooth" summations*, including sums of positive and negative powers of integers. Here is a sampler of summations that appear in this chapter.

$$
\begin{array}{l}
1 + 2 + 3 + 4 + 5 + \quad \cdots + k + (k+1) + \cdots + n \\
1 + 2 + 4 + 8 + 16 + \quad \cdots + 2^k + 2^{k+1} + \cdots + 2^n \\
1 + 4 + 9 + 16 + 25 + \cdots + k^2 + (k+1)^2 + \cdots + n^2 \\
1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{16} + \frac{1}{32} + \cdots + \frac{1}{2^k} + \frac{1}{2^{k+1}} + \cdots \\
1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \quad \cdots + \frac{1}{k} + \frac{1}{k+1} + \cdots \\
1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \cdots + \frac{1}{k^2} + \frac{1}{(k+1)^2} + \cdots
\end{array}
$$

2. We study a variety of *fundamental techniques* for performing summations.
   We include specialized techniques that work for specific classes of summations, as well as more general techniques that work in a broad range of situations.
   Examples of such techniques include, e.g.: estimating summations by integrating functions related to the summation; grouping/replication of terms within a summation; verifying "guessed" sums via induction.
3. We study a variety of *fundamental representations* of the elements being summed.
   We observe that being able to study the same phenomenon in a variety of seemingly unrelated ways often gives one unexpected mathematical understanding of and operational control over the phenomenon.
   Examples of such representations include, among others, representations of numbers by: numerals in a positional number system; slices of pie; tokens arranged in stylized ways; basic geometrical structures, including the unit-width rectangles of Riemann sums.

In summation, we treat each topic in multiple ways, as long as each new way teaches a new lesson.

To illustrate the power of summation methodology, consider the following modernized version of the *legend of Sissa ibn Dahir* who has invented a marvelous game, let's call it *chess*, that is played on an 8×8 array of unit-size squares—call it a *chessboard*. A benefactor offers you a one-time gift of *one million million (i.e.,* $10^{12} = 1,000,000,000,000$*) euros* in return for all rights to the game. As a counteroffer, you ask your benefactor instead for all of the money amassed in the following way. You ask your benefactor to proceed row by row along a chessboard, placing money in the board's squares, according to the following regimen. Your benefactor should place 1 euro in the first square, 2 euros in the second square, 4 $(= 2 \times 2)$ euros in the third, 8 $(= 4 \times 2)$ euros in the fourth, and so on, doubling the number of euros at each step of the procedure—so the last square would contain 2^{63} euros. Fig. 6.1 provides an illustration of the growing of the first steps of the filling process. *Have you made a good bargain?*

Fig. 6.1 First filled board's squares.

By the end of this chapter, you will be able to determine in minutes that under your procedure (the one that uses the chessboard), you would receive $2^{64}-1$ euros—which is more than 10^{20} euros, hence *much* more than the mere 10^{12} euros that your "benefactor" offered you!
A good bargain, indeed!

6.2 Summing Structured Series

6.2.1 Arithmetic Sums and Series

6.2.1.1 General development

We define arithmetic sequences and learn how to calculate their sums.

$$\begin{array}{l}\text{An } n\text{-term arithmetic sequence:}\\ \quad a,\ a+b,\ a+2b,\ a+3b,\ \ldots,a+(n-1)b\\ \\ \text{The corresponding arithmetic series:}\\ \quad a+(a+b)+(a+2b)+(a+3b)+\cdots+(a+(n-1)b)\\ \qquad = an+b\cdot(1+2+\cdots+n-1)\end{array} \tag{6.4}$$

We can, thus, sum the arithmetic series in (6.4) by determining the sum of the first m positive integers; $m=n-1$ in (6.4). We use this result as an opportunity to introduce important notation.

6.2.1.2 Special cases

A. Summing the first n integers: the case $a=b=1$

Our first goal is to sum the first n positive integers:

$$1\ +\ 2\ +\cdots+\ n,$$

that is, to find a *closed-form expression* for the sum. In somewhat informal terms, we say that an expression of the form

$$f(n)\ \stackrel{\text{def}}{=}\ \sum_{i=1}^{n}\ i$$

is in *closed form* if it exposes a prescription for evaluating the sum using a *fixed-length* sequence of arithmetic operations (e.g., addition/subtraction, multiplication/division, exponentiation/taking logarithms).

The desired sum $f(n)$ is commonly denoted Δ_n. We usually prefer the notation $S_1(n)$ for this sum because it exposes this summation as an instance of a related family of such summations that will occupy us through this chapter.

The remainder of subsection A is devoted to developing multiple ways to derive the following (closed-form) expression for $\Delta_n = S_1(n)$.

Proposition 6.1 *For all $n \in \mathbb{N}$,*

$$S_1(n) \;=\; \sum_{i=1}^{n} i \;=\; \frac{1}{2}n(n+1) \tag{6.5}$$

Proof. **A textual proof.** We begin with a *constructive* proof[1] of summation (6.5) that employs an approach known to the eminent German mathematician Karl Friedrich Gauss as a pre-teenager. This approach proceeds in two steps.

$$\begin{array}{ll} \text{Write } S_1(n) \text{ “forwards”:} & \sum_{i=1}^{n} i \;=\; 1 + \quad 2 \quad + \cdots + (n-1) + n \\ \text{Write } S_1(n) \text{ “in reverse”:} & \sum_{i=n}^{1} i \;=\; n + (n-1) + \cdots + \quad 2 \quad + 1 \end{array} \tag{6.6}$$

Now add the two representations of $S_1(n)$ in (6.6) *columnwise*. Because each of the n column-sums equals $n+1$, we find that $2S_1(n) = n(n+1)$, which we easily rewrite in the form (6.5) (after multiplying both sides of the equation by 2).

Remark. *Now is an opportune moment to step back from the specific result in Proposition 6.1 and concentrate on the textual proof. What Gauss noticed about the sum of the first n integers is that when the sum is doubly written as in (6.6), the column-sums are all the same. This phenomenon of* finding invariants *is a “pattern” of the form referred to in Section 2.1 as we discussed how mathematicians “do mathematics”. We see in the proof how the pattern can be exploited to determine sum of any arithmetic series.* What seemed to be a “trick” turns out to be an insightful instance of pattern-matching. *We shall soon see that the pattern can be exploited to other, related, ends.*

Not everyone thinks the same way—even within the context of mathematics. It is, therefore, very important for the reader to recognize that even the simplest mathematical facts can be proved and analyzed in a broad variety of ways. We illustrate this assertion by developing more proofs of Proposition 6.1.

Proof. **A “pictorial”, graphic proof.** The idea now is to look at the problem of summing the first n integers as a problem of estimating the area of a simple (in the *good sense* of the word) surface. In this worldview, integers are represented by concatenating basic *unit-side* (i.e., 1×1 *squares*), as in Fig. 6.2.

Our summation process can be obtained in three steps, in the manner illustrated by the three figures Figs. 6.2, 6.3, and 6.4.

[1] The proof is *constructive* in that it actually derives an answer. This is in contrast to, say, the inductive validation of the sum in Section 6.3.2.2.C, which just verifies a “guessed” answer.

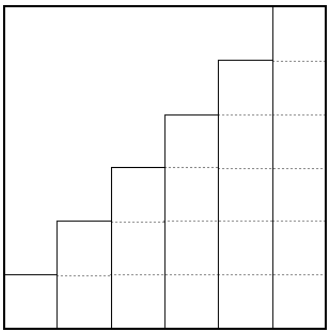


Fig. 6.2 Representing the first n integers using basic unit squares; $n = 6$ in this example.

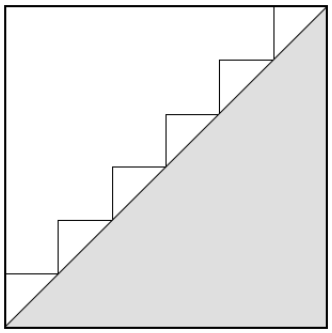


Fig. 6.3 The area of the lower-right triangle (light grey) is one-half that of the entire $n \times n$ square.

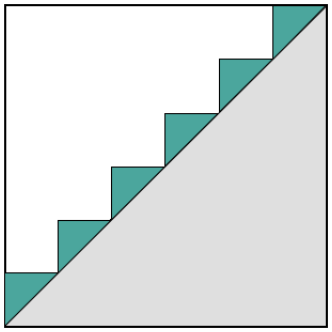


Fig. 6.4 The area of the (dark) triangles sitting on the upper diagonal of the $n \times n$ square is $\frac{1}{2}n$.

1. We begin, in Fig. 6.2, by depicting the problem calculating $S_1(n)$ as the problem of determining the area of a surface constructed from unit-side squares.

2. Next, we illustrate in Fig. 6.3 that the area of the lower-right triangle of the $n \times n$ square—depicted in light grey in the figure—is one-half that of the entire $n \times n$ square.
3. Finally, we indicate in Fig. 6.4 that the area of the small (dark grey in the figure) triangles that cover the upper diagonal of the $n \times n$ square is $\frac{1}{2}n$. This reckoning notes that there are n triangles, and each has an area that is one-half that of a unit-side square.

We thereby reckon the area of the surface depicting $S_1(n)$ as

One-half the area of the $n \times n$ square, i.e., $\frac{1}{2}n^2$

plus

n times the area of one-half a unit-side square, i.e., $\frac{1}{2}n$

We have, thus, derived the value of $S_1(n)$.

We present one final proof of Proposition 6.1.

Proof. **A combinatorial proof.** The following argument is *combinatorial* in that it achieves its goal by *counting* instances of the first n integers, laid out in a line.

Place (tokens that represent) the integers 0 to n along a line. For each integer i, count how many integers $j > i$ lie to its right. We see that in general, there is a *block* of $n - i$ integers that lie to the right of integer i. In detail: the block of integers lying to the right of $i = 0$ contains n values of j; the block to the right of $i = 1$ contains $n - 1$ values of j, and so on, as suggested in Fig. 6.5.

All integers ≤ 4:	0 1 2 3 4
integers to the right of 0:	1 2 3 4
integers to the right of 1:	2 3 4
integers to the right of 2:	3 4
integers to the right of 3:	4

Fig. 6.5 A two-dimensional (triangular) depiction of the right-lying integer-instances.

On the one hand, we see that the total number of right-lying integers j equals $n + (n-1) + ... + 1 \ = \ S_1(n)$.

On the other hand, every instance of a right-lying integer can be identified uniquely by the pair of nonnegative integers, i (the instance's block) and $j > i$ (the instance's position-within-block). The total number of right-lying integer-instances corresponds to the number of ways to select two integers from among $n + 1$.[2] This number is the binomial coefficient whose definition we specialize from equation (5.3) in Section 5.1.1.2.C:[3]

[2] We study such counting techniques in depth in Section 9.2.

[3] The sums Δ_n are, thus, special binomial coefficients, namely, $\binom{n+1}{2}$. The many ways of viewing the underlying summation in terms of triangles—as in Figs. 6.2–6.5—have therefore led to the naming of these special binomial coefficients as *triangular numbers*.

$$\Delta_n \;=\; \binom{n+1}{2} \;\stackrel{\text{def}}{=}\; \frac{1}{2}n(n+1).$$

We have thus derived two distinct—but, of course, equal—expressions for $S_1(n)$.

≈≈≈≈≈≈≈≈≈≈

Our combinatorial derivation of the sum (6.5) illustrates one of the most important roles of mathematical abstraction. There is no obvious intuition to explain the relationship between the activity of summing n consecutive integers and the activity of extracting two items out of a set of n items. Yet, our combinatorial derivation exposes an intimate connection between the two.

≈≈≈≈≈≈≈≈≈≈

Now that we know—*and understand*—how to derive the value of $S_1(n)$, we can finally evaluate our original series in (6.4).

Proposition 6.2 *The arithmetic series in (6.4) has the sum*

$$a+(a+b)+(a+2b)+(a+3b)+\cdots+(a+(n-1)b) \;=\; an+b\cdot\Delta_n. \qquad (6.7)$$

B. Perfect squares are sums of odd integers

In this section, we build on Proposition 6.1 to craft multiple constructive proofs of the fact that each perfect square, say, n^2, is the sum of the first n odd integers, $1,3,5,\ldots,2n-1$. All of these proofs complement the "guess-and-verify" inductive proof of this result in Section 6.3.2.2.C.

Proposition 6.3 *For all $n \in \mathbb{N}^+$,*

$$\sum_{k=1}^{n}(2k-1) \;=\; 1+3+5+\cdots+(2n-1) \;=\; n^2. \qquad (6.8)$$

That, is, the nth perfect square is the sum of the first n odd integers.

Before we present several proofs of this result, we note that the notation for odd integers in (6.8) is perfectly general: every positive odd integer n can be written in the form $2k-1$ for some positive integer k.

Our first two proofs of Proposition 6.3 note that the result is a corollary of both Proposition 6.1 and Proposition 6.2.

The first of these proofs builds on the stratagem of *finding invariants* that we exploited in the textual proof of Proposition 6.1.

Proof. **A proof using algebra.** By direct calculation, we find that

$$\begin{aligned}\sum_{k=1}^{n}(2k-1) &= 2\sum_{k=1}^{n}k \;-\; n \\ &= 2\Delta_n \;-\; n \qquad \text{(by Proposition 6.1)} \\ &= (n^2+n)-n \\ &= n^2.\end{aligned}$$

Proof. **A proof by calculation.** Because summation (6.8) is an arithmetic series with $a = 1$ and $b = 2$, we know from Proposition 6.2 that the summation evaluates to

$$(1\cdot n)+2\Delta_{n-1} \;=\; n+n^2-n \;=\; n^2.$$

Proof. **A textual proof.** We adapt Gauss's "trick", wherein one adds the series written forwards to the series written backwards, to this summation. Let us denote the target sum $\sum_{k=1}^{n}(2k-1)$ by $S(n)$. We record $S(n)$ in two ways:

$$\begin{aligned}&\text{"Forwards":}\quad S(n) \;=\; 1 \;+\; 3 \;+\cdots+\; (2n-3) \;+\; (2n-1) \\ &\text{"Backwards":}\; S(n) \;=\; (2n-1) \;+\; (2n-3) \;+\cdots+\; 3 \;+\; 1\end{aligned} \tag{6.9}$$

Now add these two representations of $S(n)$ *columnwise*. Because each of the n column-sums equals $2n$, we find that

$$2S(n) \;=\; 2\sum_{k=1}^{n}(2k-1) \;=\; 2n^2. \tag{6.10}$$

We thus derive the sum (6.8) when we halve the three equated quantities in equation (6.10), i.e., divide each quantity by 2.

Proof. **A proof "by pictures".** We now build up to a proof that is almost purely pictorial, with just a bit of reasoning mixed in. The only "sophisticated" knowledge required is that

$$(n+1)^2 \;=\; n^2 \;+\; 2n \;+\; 1. \tag{6.11}$$

≈≈≈≈≈≈≈≈≈≈

We remark that equation (6.11) is the simplest instance of the restricted Binomial Theorem*, which appears later in this chapter as Theorem 6.1.*

≈≈≈≈≈≈≈≈≈≈

The well-known equation (6.11) can be verified by explicitly symbolically squaring $n+1$:

$$(n+1)\cdot(n+1) \;=\; n\cdot(n+1) \;+\; (n+1) \;=\; n^2 \;+\; n \;+\; n \;+\; 1.$$

The equation can also be verified using the highly perspicuous Fig. 6.6. The figure

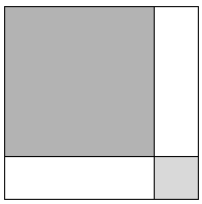


Fig. 6.6 A geometrical proof of the identity $(n+1)^2 = n^2 + 2n + 1$.

tells its tale by exhibiting four rectangles that make up an $(n+1)\times(n+1)$ square; the area of this square is, of course, $(n+1)^2$. This large square is made up of four rectangles.

- Reading across the top of the figure, we encounter a darkly shaded $n \times n$ square (area $= n^2$) and an unshaded $n \times 1$ rectangle (area $= n$).
- Reading across the bottom of the figure, we encounter an unshaded $1 \times n$ rectangle (area $= n$) and a lightly shaded 1×1 square (area = 1).

The overall message is that $(n+1)^2$ (the area of the large, composite, square)
is the sum of
n^2 (the area of the darkly shaded square)
plus
$2n$ (the combined areas of the unshaded rectangles)
plus
1 (the area of the lightly shaded square)

Back to our proof of Proposition 6.3. Our pictorial proof begins by representing each integer n as a horizontal sequence of n "bullets", i.e., darkened circles. The problem of summing the first n odd integers then begins with a picture such as appears in Fig. 6.7, for the illustrative case $n = 5$.

Fig. 6.7 Representing the first n odd integers using bullets. In this illustration, $n = 5$.

Starting with such a picture, we take each row of $2k-1$ bullets and fold it at its midpoint so that it becomes a reversed letter "L". The row of $2k-1$ bullets becomes an "L" whose horizontal portion (at the bottom of the reversed "L") is a row of k bullets and whose vertical portion (at the right of the reversed "L") is a column of k bullets (one bullet is in common). See Fig. 6.8 wherein the depicted values of k are $k = 1,2,3,4,5$.

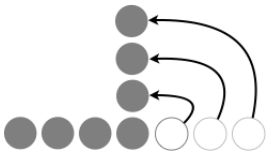


Fig. 6.8 Folding a single row into a reversed letter "L".

Once we have folded every row of bullets into a reversed "L", we nest the occurrences of "L" in the manner depicted in Fig 6.9. Clearly, this nesting produces an

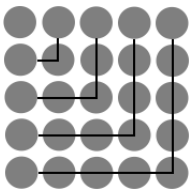


Fig. 6.9 The final picture organized as an $n \times n$ square array of bullets.

$n \times n$ square array of (perforce, n^2) bullets.

Proof. **Another proof "by pictures".** The reader who enjoyed our "proof 'by pictures"' may be amused by the challenge of completing the kindred proof that is illustrated by Fig. 6.10. The figure arranges four copies of the triangle of bullets that illustrates summation $S(n)$ in such a way that the triangles combine to produce the $2n \times 2n$ square of bullets. The underlying arithmetic exposes the fact that the side of the square consists of $1 + 2n - 1 \;=\; 2n$ bullets. The conclusion is that $4S(n) \;=\; (2n)^2 n \;=\; 4n^2$.

Proof. **A proof by rearranging terms.** We describe in Section 2.2.5.2 how the Italian mathematician Guido Fubini was able to make notable mathematical progress by rearranging representations [35]. Within the context of the current chapter, such rearrangements work on the terms of a summation of interest. Indeed, using this

Fig. 6.10 Four copies of $S(n)$ represented as a trianle of bullets. The triangles are arranged to yield a $2n \times 2n$ square of bullets.

strategy, we obtain a surprising, delightful proof of Proposition 6.3. Let us take the odd integers in order, in groups of sizes 1, then 2, then 3, and so on. We begin with the first n odd integers in order:

$$1,\ 3,\ 5,\ 7,\ 9,\ 11,\ 13,\ 15,\ 17,\ 19,\ \ldots$$

Now we start peeling off prefixes of successive numbers of odd integers and arranging them in an array, as depicted in the following table.

$$\begin{array}{lll}
\text{group of size 1:} & 1, & \to 1 = 1^3 \\
\text{group of size 2:} & 3,\ 5, & \to 2 \times 4 = 2^3 \\
\text{group of size 3:} & 7,\ 9,\ 11, & \to 3 \times 9 = 3^3 \\
\text{group of size 4:} & 13,\ 15,\ 17,\ 19 & \to 4 \times 16 = 4^3
\end{array}$$

What we note is that—at least with the illustrated portion of the procedure—the kth group, of size k, adds up to k^3.

Before we proceed further, let us verify—by induction—that this pattern persists indefinitely.

Base for the induction. The trivial one-term entry in row 1 of the preceding table provides the base for our induction.

Inductive hypothesis. We know from Proposition 6.1 that the kth group consists of k consecutive odd numbers beginning with

$$2\Delta_{k-1}+1 \quad \text{which is the} \quad (\Delta_{k-1}+1)\,\text{th odd number}$$

Hence, this group begins with $2\Delta_{k-1}+1$ and ends with $2\Delta_k - 1$.

This proof can be represented graphically as follows. We begin with the unit square (the leftmost item in Fig. 6.11) as the basis of our induction. We next represent the number 2^3 (the *cube* of 2) by two 2×2 squares: the figures following

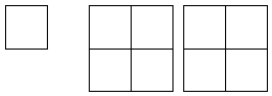


Fig. 6.11 The basis for the inductive pictorial proof

the unit square in Fig. 6.11. Fig. 6.12 illustrates graphically that $1+2^3$ is a per-

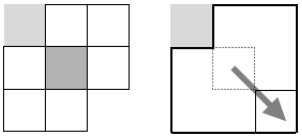


Fig. 6.12 Showing graphically that 2^3+1 is a perfect square

fect square. Fig. 6.13 illustrates that iterating the process also produces a perfect

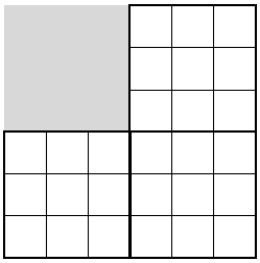


Fig. 6.13 The next step in the construction also produces a perfect square

square. Comparing Figs. 6.12 and 6.13 indicates a parity constraint on the process: at even-numbered steps,the subsquares that get "merged" are overlapping; at odd-numbered steps, they are not.

Inductive extension. Since successive odd numbers differ by 2, we know that the kth group consists of the following odd integers ($k > 1$):

$$2\Delta_{k-1}+1,\ 2\Delta_{k-1}+3,\ 2\Delta_{k-1}+5,\ldots,\ 2\Delta_{k-1}+(2k-1)$$

Once one verifies (say, by induction) that $2\Delta_{k-1}+2k-1 \ = \ 2\Delta_k+1$, one discovers that this group has the sum

$$2k\Delta_{k-1} \ - \ 2\Delta_{k-1}+k \ = \ (2k-1)\Delta_k+k \ = \ k^3.$$

Because each row contains one more integer than its predecessor, row k contains the sum of the odd integers from $\Delta_{k-1}+1$ to Δ_k. By direct calculation, we can therefore compute the row's sum as

$$\Delta_k^2 \; - \; \Delta_{k-1}^2 \;\; = \;\; \frac{k^2}{4}\left((k+1)^2-(k-1)^2\right) \;\; = \;\; k^3$$

We close our treatment of Proposition 6.3 with another, really basic proof. As stated earlier, our goal is really to encourage readers to utilize every possible mathematical concept as they hone their mathematical and computational skills: *Simple proofs often lend as much insight as sophisticated ones.*

Proof. **A final proof, from elementary school.** Consider the following reasoning which emerges from the way multipication tables are developed in elementary school. We illustrate the idea using the case $n = 5$.

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 6 & 8 & 10 \\ 3 & 6 & 9 & 12 & 15 \\ 4 & 8 & 12 & 16 & 20 \\ 5 & 10 & 15 & 20 & 25 \end{array}$$

Write the integers $1,2,\ldots,n$ in a row. Below this row, write the doubles of these integers. Below the "double" row, write the triples of the integers. Below the "triple" row, write the quadruples of the integers, then the quintuples, and so on. Note that the resulting table is symmetric: its rows are identical to its columns.

Using Fubini's rearrangement stratagem, we count all the integers in the table in two different ways.

1. We sum the entries of our table by peeling away successively larger reversed instances of the letter "L" (as in our earlier "pictorial" proof of Proposition 6.3). We find that the integers in each "L" sum to a perfect cube:

$$\begin{array}{ccccccccc|l} 1 & & & & & & & & & 1 = 1^3 \\ 2 & 4 & 2 & & & & & & & 8 = 2^3 \\ 3 & 6 & 9 & 6 & 3 & & & & & 27 = 3^3 \\ 4 & 8 & 12 & 16 & 12 & 8 & 4 & & & 64 = 4^3 \\ 5 & 10 & 15 & 20 & 25 & 20 & 15 & 10 & 5 & 125 = 5^3 \end{array}$$

2. We sum the successive rows of our previous n by n table. The first row of the table sums to Δ_n; the second sums sums to $2\Delta_n$; the third sums to $3\Delta_n$, $\ldots$; the last row sums to $n\Delta_n$. Thus, the total sum is $(1+2+\cdots+n)\Delta_n \;=\; (\Delta_n)^2$.

We conclude that

$$\sum_{i=1}^{n} i^3 \;=\; (\Delta_n)^2$$

A promise, as we close the section. In Section 6.3.2.1, we develop the underpinnings of techniques that incrementally compute the sums of the first n consecutive integers (the summations $S_1(n)$), the squares of these integers (the summations $S_2(n)$), the cubes of these integers (the summations $S_3(n)$), and so on. Most interesting are the techniques that are incremental, i.e., that compute the summations $S_c(n)$ for cth powers of integers from the summations for smaller powers: $S_1(n)$, $S_2(n)$, $\ldots$, $S_{c-1}(n)$.

C. A nonobvious identity for arithmetic sums

We close this subsection by using a "picture" to verify an identity for arithmetic sums that one would be unlikely to come upon by purely textual thinking.

Proposition 6.4 *For any positive integer n,*

$$\Delta_{2n-1} \;=\; n + 4\Delta_{n-1}.$$

Proof. Consider the arithmetic series in (6.4) for the case $a = 1$ and $b = 4$. By Proposition 6.2, this series, call it $S^{(1,4)}(n)$, has the sum

$$S^{(1,4)}(n) \;=\; n + 4\Delta_{n-1}. \tag{6.12}$$

Let us represent the sum Δ_{n-1} in the natural way as a triangle of bullets. This triangle has a base of $n-1$ bullets, upon which sits a row of $n-2$ bullets, upon which sits a row of $n-3$ bullets, $\ldots$, all the way to the apex, which has a single bullet.

Now, let us view equation (6.12) as giving us access to four copies of the preceding triangle of bullets. Let us arrange these triangles in the manner depicted in Fig. 6.14. Now, "complete the picture" by adding an "extra" row of n bullets at row

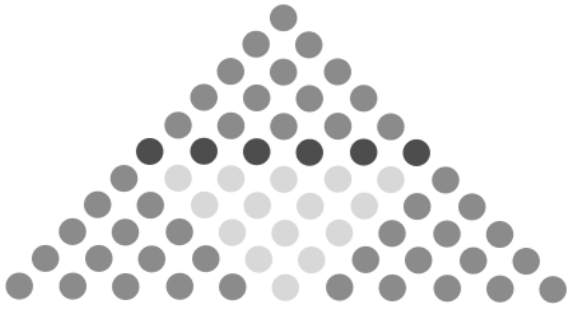


Fig. 6.14 Arranging the four triangles plus a row to obtain a new (bigger) triangle.

n of the figure (these are depicted in dark gray in the figure). The four small trian-

gles, augmented by the "extra" row of n bullets has clearly become a representation of Δ_{2n-1} by bullets.

We now have a purely pictorial proof of the proposition.

≈≈≈≈≈≈≈≈≈≈

Full disclosure: *Our proof of Proposition 6.4 is not* purely *pictorial, because we must somehow verify that the construction is completely general, i.e., that the depicted emergence of the Δ_{2n-1} triangle of bullets from four copies of the Δ_n-triangle plus the extra row of n bullets is not an artifact of the depicted case $n = 6$.*

Even with this caveat, one must admit that the discovery *of the proposition is really pictorial, even if the verification requires additional modalities of reasoning.*

≈≈≈≈≈≈≈≈≈≈

6.2.2 Geometric Sums and Series

6.2.2.1 Overview and main results

We define geometric sequences and learn how to calculate their sums via the following generic examples.

An n-term geometric sequence:

$$a,\ ab,\ ab^2,\ \ldots, ab^{n-1} \tag{6.13}$$

The corresponding geometric summation:

$$\begin{aligned} S_{a,b}(n) &\stackrel{\text{def}}{=} \sum_{i=0}^{n-1} ab^i \\ &= a + ab + ab^2 + \cdots + ab^{n-1} \\ &= a \cdot (1 + b + b^2 + \cdots + b^{n-1}) \end{aligned} \tag{6.14}$$

The associated geometric *(infinite) series* (used only when $b < 1$):

$$S_{a,b}^{(\infty)} \stackrel{\text{def}}{=} \sum_{i=0}^{\infty} ab^i = a + ab + ab^2 + \cdots \tag{6.15}$$

It is clear from these definitions that we can evaluate the summation (6.14) by evaluating just the sub-summation

$$S_b(n) \stackrel{\text{def}}{=} \sum_{i=0}^{n-1} b^i = 1 + b + b^2 + \cdots + b^{n-1}, \tag{6.16}$$

and we can evaluate the series (6.15) by evaluating just the sub-series

$$S_b^{(\infty)} \stackrel{\text{def}}{=} \sum_{i=0}^{\infty} b_i = 1 + b + b^2 + \cdots \tag{6.17}$$

The major results that we develop in this section are:

Proposition 6.5 *Let $S_b(n)$ be a geometric summation, as defined in (6.16).*
(a) *When $b > 1$, $S_b(n)$ evaluates to the following sum.*

$$S_b^{(b>1)}(n) = \frac{b^n - 1}{b - 1}. \tag{6.18}$$

(b) *When $b < 1$, $S_b(n)$ evaluates to the following sum.*

$$S_b^{(b<1)}(n) = \frac{1 - b^n}{1 - b}. \tag{6.19}$$

Of course, in the uninteresting degenerate case $b = 1$

$$S_b^{(b=1)}(n) = 1 + 1 + \cdots + 1 \quad (n \text{ times}) \quad = n.$$

The infinite case (6.17) can be dealt with as a corollary to Proposition 6.5(b), by letting n grow without bound and observing that the resulting sequence of values converges.

Proposition 6.6 *When $b < 1$, the* infinite *series $S_b^{(\infty)}$* converges *to the following sum.*

$$S_b^{(\infty)} = \sum_{i=0}^{\infty} b^i = 1 + b + b^2 + \cdots = \frac{1}{1 - b}.$$

6.2.2.2 Techniques for summing geometric series

We turn now to a sequence of proofs of Propositions 6.5 and 6.6.

Proof. **A proof by textual replication.** Toward the end of developing our first method for summing $S_b(n)$, we note that we can rewrite the sum in two ways that are *(textually) recurrent*.

This phenomenon of finding recurrent subexpressions *is a "pattern" of the form described in Section 2.1 as we discussed how mathematicians "do mathematics". We now exemplify how this pattern can be exploited to find explicit sums for geometric summations and series.*

Both of the recurrent expressions for $S_b(n)$ have the following form.

$$S_b(n) \;=\; \alpha \cdot S_b(n) \;+\; \beta(n) \tag{6.20}$$

where α is a constant and $\beta(n)$ is a function of n; both α and $\beta(n)$ may depend on the parameter b. We provide two recurrent expressions for $S_b(n)$, one of which is more interesting when $b > 1$, the other when $b < 1$.

$$\begin{aligned} S_b(n) &\stackrel{\text{def}}{=} 1 + b + b^2 + \cdots + b^{n-1} \\ &= b \cdot S_b(n) \;+\; (1 - b^n) \qquad (6.21) \\ &= \frac{1}{b} \cdot S_b(n) \;+\; \frac{b^n - 1}{b} \qquad (6.22) \end{aligned}$$

The significance of a recurrent expression of the form (6.20) is that it exposes an explicit value for $S_b(n)$:

$$S_b(n) \;=\; \frac{\beta(n)}{1 - \alpha} \tag{6.23}$$

We now combine the generic value (6.23) of $S_b(n)$ with the specialized recurrent expressions in (6.21) and (6.22) to derive two explicit solutions for $S_b(n)$.

1. The first solution is most useful and perspicuous when $b > 1$. In this case, we find that
$$\left(1 - \frac{1}{b}\right) S_b^{(b>1)}(n) \;=\; b^{n-1} - \frac{1}{b},$$
which can easily be rearranged to the equivalent and more perspicuous form (6.18).
2. The second solution is most useful and perspicuous when $b < 1$. In this case, we find that
$$(1 - b) S_b^{(b<1)}(n) \;=\; 1 \;-\; b^n$$
which can easily be rearranged to the equivalent and more perspicuous form (6.19).

Note that both $S_b^{(b>1)}(n)$ and $S_b^{(b<1)}(n)$ have simple *approximate* values which are useful in "back-of-the-envelope" calculations: For very large values of n, we have

$$S_b^{(b>1)}(n) \;\approx\; \frac{b^n}{b-1} \quad \text{and} \quad S_b^{(b<1)}(n) \;\approx\; \frac{1}{1-b}. \tag{6.24}$$

The expression for $S_b^{(b<1)}(n)$ in (6.24) is actually a rewording of Proposition 6.6.

Proof. **A pictorial representation for summing $S_{1/2}^{(\infty)}$.** Fig. 6.15 depicts a pictorial process whose analysis provides a rigorous proof of Proposition 6.6 for the case $b = 1/2$, *i.e.*, a rigorous argument that the series $S_{1/2}^{(\infty)} = \sum_{i=0}^{\infty} 2^{-i}$ sums to 2. In this evaluation of $S_{1/2}^{(\infty)}$, we measure fractional quantities by the portion of a unit-side

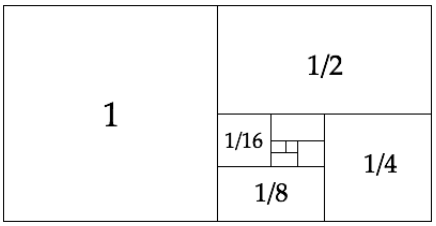


Fig. 6.15 Arranging successive rectangles to evaluate $S_{1/2}^{(\infty)}$.

rectangle that they fill. Thus (follow in the figure): the initial term of $S_{1/2}^{(\infty)}$, namely 1, is represented by the unit square that is labeled "1" in the figure. The next term of the series, namely $1/2$, is represented by the rectangle labeled "$1/2$" in the figure, and so on, with successively smaller rectangles. By designing each rectangle to have half the area of its predecessor, the sequence of rectangles thus represents successively smaller inverse powers of 2. As the process proceeds, we observe increasingly more of the righthand unit-side square being filled. In fact, one can argue that *every* point in the righthand unit-side square eventually gets covered by some small rectangle (as n tends to ∞), thereby establishing that the infinite series $S_{1/2}^{(\infty)}$ does, indeed, sum to 2.

This procedure is difficult, but not impossible, to adapt to values of $b < 1$ other than $1/2$. The sequence Fig. 6.16, Fig. 6.17, Fig. 6.18 suggests how to achieve such

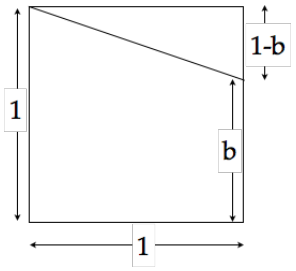


Fig. 6.16 Initial state: the unit square and the base b.

an adaptation for any value of b with $0 \leq b < 1$, by an appropriate cascade of shrinking squares. The unit-side square in Fig. 6.16 begins the construction of the cascade. The two squares in Fig. 6.17 illustrate the second step in constructing the cascade; the suggestive cascade depicted in Fig. 6.18 illustrates what the final cascade looks like: the cumulative length of the bases of its abutting rectangles is the value of the infinite series $= \sum_{i=0}^{\infty} b^i$ (where $0 \leq b < 1$).

We should add a final remark here. The final touch is that the infinite sum is given by the base of the big right rectangle. It is similar (we call this property semblable

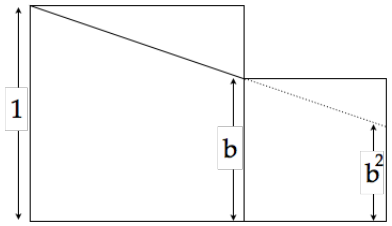


Fig. 6.17 Beginning to craft the geometric series by cascading shrinking squares.

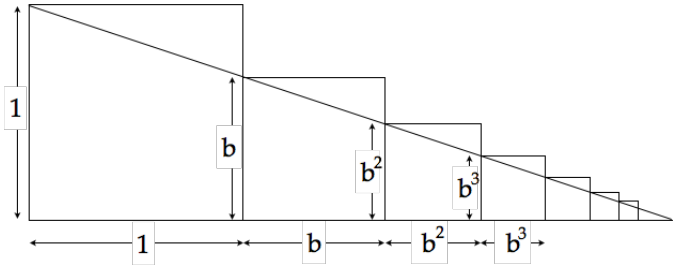


Fig. 6.18 The complete process for computing a geometric series using a cascade of shrinking squares.

in french) to the little right rectangle top left in the figure. By Thales theorem in geometry, the ratio of the sides are proportional: $\frac{S_b^{(\infty)}}{1} = \frac{1}{1-b}$.

Proof. **Another pictorial representation for summing $S_{1/2}^{(\infty)}$.** The pictorial derivation of the sum $S_{1/2}^{(\infty)}$ can be accomplished using geometric shapes other than squares. We now present a natural derives the sum $S_{1/2}^{(\infty)}$ by vigorously slicing a pie.

The process of pie-slicing works most naturally with the modified series

$$\overline{S}_{1/2}^{(\infty)} \;=\; \sum_{i=1}^{\infty} 2^{-i} \;=\; S_{1/2}^{(\infty)} \;-\; 1.$$

which omits the initial summand 1 from $S_{1/2}^{(\infty)}$. Of course, $\overline{S}_{1/2}^{(\infty)}$ sums to 1, because $S_{1/2}^{(\infty)}$ sums to 2.

The pie-slicing evaluation of $\overline{S}_{1/2}^{(\infty)}$ is depicted in Fig 6.19. In the figure, the inverse powers of 2 are represented by appropriate fractions of a unit-diameter disk (the pie). The evaluation begins with this disk before it is sliced: this represents the number 1, which we eventually show to be the sum of $\overline{S}_{1/2}^{(\infty)}$. We slice the disk in half by means of the depicted diameter; we label one of the resulting half-disks "$1/2$". Next, we

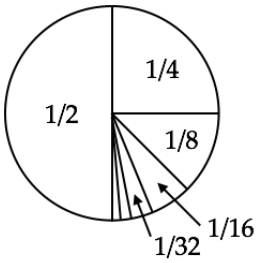


Fig. 6.19 Computing the sum of $1/2^i$ using a unit disk.

slice one of the half-disks in half by means of a radius of the unit disk; we label one of the quarter-disks "$1/4$". We continue in this manner *ad infinitum*. The analysis that yields the sum of $\overline{S}^{(\infty)}_{1/2}$ amounts to a proof that every point in the unit-diameter disk eventually resides in a slice that is not further sliced. Details are left to the interested reader.

6.2.2.3 A fun result via geometric sums: When is integer n divisible by 9?

We now exploit our ability to evaluate geometric summations to illustrate a somewhat surprising, nontrivial fact. One can deduce information about the divisibility of an integer n from n's positional numerals. We hope that this "fun" result will inspire the reader to seek kindred numeral-encoded properties of numbers.

Proposition 6.7 *An integer n is divisible by an integer m if, and only if, m divides the sum of the digits in the base-$(m+1)$ numeral for n.*

The most familiar instance of this result is phrased in terms of our traditional use of base-10 (decimal) numerals.
An integer n is divisible by 9 if, and only if, the sum of the digits of n's base-10 numeral is divisible by 9.

Proof. (*Argument for general number-base b*). Of course, we lose no generality by focusing on numerals without leading 0's, because leading 0's do not alter a numeral's sum of digits.

Let us focus on the base-b numeral for a number n (so $b = m+1$ in the statement of the proposition). There therefore exist base-b digits—i.e., integers from the set $\{0,1,\ldots,b-1\}$—call them $\delta_k \neq 0$, $\delta_{k-1}, \ldots \delta_1, \delta_0$, such that

$$n \;=\; \delta_k \cdot b^k + \delta_{k-1} \cdot b_{k-1} + \cdots + \delta_1 \cdot b + \delta_0.$$

The sum of the digits of n's base-b numeral is, then

$$s_b(n) \stackrel{\text{def}}{=} \delta_k + \delta_{k-1} + \cdots + \delta_1 + \delta_0.$$

Let us calculate the difference $n - s_b(n)$ in the following manner, digit by digit.

$$\begin{array}{rcccccccc} n & = & \delta_k \cdot b^k & + & \delta_{k-1} \cdot b^{k-1} & + \cdots + & \delta_1 \cdot b & + \delta_0 \\ s_b(n) & = & \delta_k & + & \delta_{k-1} & + \cdots + & \delta_1 & + \delta_0 \\ \hline n - s_b(n) & = & \delta_k \cdot (b^k - 1) & + & \delta_{k-1} \cdot (b^{k-1} - 1) & + \cdots + & \delta_1 \cdot (b-1) & & \end{array} \tag{6.25}$$

We now revisit summation (6.18). Because b is a positive integer, so that $1 + b + \cdots + b^{a-2} + b^{a-1}$ is also a positive integer, we infer that *the integer $b^a - 1$ is divisible by $b - 1$*.

We are almost home. Look at the equation for $n - s_b(n)$ in the system (6.25). As we have just seen, every term on the righthand side of that equation is divisible by $b - 1$. It follows therefore, that the lefthand expression, $n - s_b(n)$, is also divisible by $b - 1$. An easy calculation, which we leave to the reader, now shows that this final fact means that n is divisible by $b - 1$ if, and only if, $s_b(n)$ is.

6.2.2.4 Extended geometric series and their sums

We now build on our ability to evaluate geometric summations of the forms (6.18, 6.19) to evaluate summations that we shall term *extended* geometric summations (not a standard term), *i.e.*, summations of the form

$$S_b^{(c)}(n) \stackrel{\text{def}}{=} \sum_{i=1}^{n} i^c b^i,$$

where c is an arbitrary fixed positive integer, and b is an arbitrary fixed real number.

We restrict attention here to the situation defined by the joint non-equalities $c \neq 0$ and $b \neq 1$.

- We have already adequately studied the case $c = 0$, which characterizes "ordinary" geometric summations.
- The case $b = 1$ removes the "geometric growth" of the sequence underlying the summation. We study various aspects of this *summation-of-fixed-powers* case in Section 6.3, with special treatment of summations of fixed powers of consecutive integers in Section 6.3.2.

The method we now develop for evaluating all other summations $S_b^{(c)}(n)$, *i.e.*, those with $b \neq 1$ and $c \neq 0$, has two major characteristics.

1. The method is *inductive in parameter c*, in the sense that we will be able to express our sum for $S_b^{(c)}(n)$ in terms of sums for summations $S_b^{(c-1)}(n)$, $S_b^{(c-2)}(n)$, $\ldots$, $S_b^{(1)}(n)$, $S_b^{(0)}(n) = S_b(n)$. And, for each fixed value of c, the method is *inductive in the argument n*.
2. The method will rely on the recurrent-subexpression strategy which was so effective in Section 6.2.2.

A. The extended geometric sum $S_b^{(1)}(n) = \sum_{i=1}^{n} ib^i$

We illustrate our strategy in detail for the case $c = 1$ and sketch only briefly how it deals with larger values of c. Elementary algebraic manipulations which are suggested by the analysis in the case $c = 1$ should thereby allow the reader to deal with any value $c > 1$.

Proposition 6.8 *For all bases $b > 1$,*

$$S_b^{(1)}(n) \;=\; \sum_{i=1}^{n} ib^i \;=\; \frac{(b-1)n-1}{(b-1)^2}\cdot b^{n+1} \;+\; \frac{b}{(b-1)^2} \tag{6.26}$$

Proof. **Deriving a sum via algebraic manipulation.** We begin to develop our strategy by writing the natural expression for

$$S_b^{(1)}(n) \;=\; b \;+\; 2b^2 \;+\; 3b^3 \;+\cdots+\; nb^n$$

in two different ways. First, we isolate the summation's last term:

$$S_b^{(1)}(n+1) \;=\; S_b^{(1)}(n) \;+\; (n+1)b^{n+1}. \tag{6.27}$$

Then we isolate the summation's first term:

$$\begin{aligned}
S_b^{(1)}(n+1) &= b + \sum_{i=2}^{n+1} ib^i \\
&= b + \sum_{i=1}^{n} (i+1)b^{i+1} \\
&= b + b\cdot\sum_{i=1}^{n} (i+1)b^i \\
&= b + b\cdot\left(\sum_{i=1}^{n} ib^i \;+\; \sum_{i=1}^{n} b^i\right) \\
&= b\cdot\left(S_b^{(1)}(n) \;+\; S_b^{(0)}(n)\right) \;+\; b \\
&= b\cdot\left(S_b^{(1)}(n) \;+\frac{b^{n+1}-1}{b-1} - 1\right) \;+\; b \\
&= b\cdot S_b^{(1)}(n) \;+\; b\cdot\frac{b^{n+1}-1}{b-1}
\end{aligned} \tag{6.28}$$

Combining expressions (6.27) and (6.28) for $S_b^{(1)}(n+1)$, we finally find that

$$(b-1)\cdot S_b^{(1)}(n) = (n+1)\cdot b^{n+1} \;-\; b\cdot\frac{b^{n+1}-1}{b-1}$$

$$= \left(n - \frac{1}{b-1}\right) \cdot b^{n+1} + \frac{b}{b-1} \tag{6.29}$$

One now uses standard algebraic manipulations to derive expression (6.26) from equation (6.29).

Proof. **Solving the case $b = 2$ using subsum rearrangement.** We can evaluate the sum

$$S_2^{(1)}(n) = \sum_{i=1}^{n} i2^i$$

in an especially interesting way, by rearranging the sub-summations of the target summation.

≈≈≈≈≈≈≈≈≈≈

The reader should pay careful attention to this technique. It can sometimes decompose a cumbersome expression for a summation into a readily manipulated one.

≈≈≈≈≈≈≈≈≈≈

Note that we can rewrite summation $S_2^{(1)}(n)$ as a *double summation:*

$$S_2^{(1)}(n) = \sum_{i=1}^{n} \sum_{k=1}^{i} 2^i \tag{6.30}$$

By suitable applications of the laws of arithmetic Section 5.1.2—specifically, the distributive, associative, and commutative laws—we can perform the required double summation in a different order than that specified in (6.30). In fact, we can exchange the indices of summation, to compute $S_2^{(1)}(n)$ as specified via the following expression:

$$S_2^{(1)}(n) = \sum_{k=1}^{n} \sum_{i=k}^{n} 2^i.$$

This process is illustrated in Fig. 6.20. The indicated summation is much easier to perform in this order, because its core consists of instances of the "ordinary" geometric summation $\sum_{i=k}^{n} 2^i$ (see Proposition 6.5). Expanding these instances, we find finally that

$$S_2^{(1)}(n) = \sum_{k=1}^{n} (2^{n+1} - 1 - \sum_{i=0}^{k-1} 2^i).$$

$$S_2^{(1)}(n) = \sum_{k=1}^{n} (2^{n+1} - 2^k).$$

$$S_2^{(1)}(n) = n \cdot 2^{n+1} - (2^{n+1} - 1) + 1 = (n-1) \cdot 2^{n+1} + 2.$$

I added an interesting comment to finish this section: Let remark that the process of obtaining the single summation can be also seen in the figure by scanning the

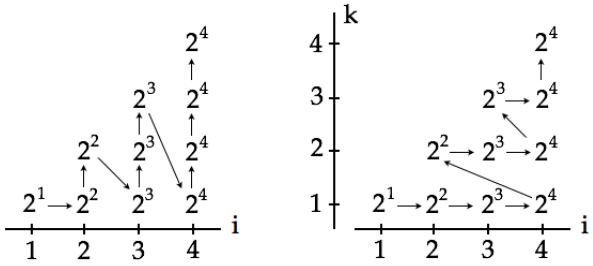


Fig. 6.20 Illustration of the exchange of indices in the summation. The original sum is on the left, each $i \cdot 2^i$ is distributed i times on column i. The second sum is obtained by introducing a new index k and to scan each term of the sum row by row (on the right). The arrows represent the order of execution. The drawing was done for $n = 4$

elements of the sum by diagonals (see Fig. 6.21). Each of the n diagonal is exactly the different between the complete geometric series minus the partial geometric series truncated at rank k.

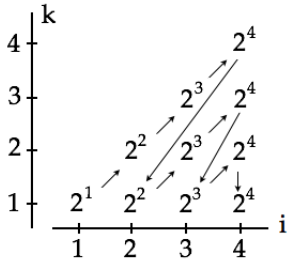


Fig. 6.21 The successive diagonal patterns correspond to the single summation obtained after exchanging the two initial sums.

B. The general extended geometric sum $S_b^{(c)}(n) = \sum_{i=1}^{n} i^c b^i$

We now develop a strategy that adapts the evaluation of summation $S_b^{(1)}(n)$ in the proof of Proposition 6.8 to an evaluation of the general extended geometric summation

$$S_b^{(c)}(n) \;=\; \sum_{i=1}^{n} i^c b^i \;=\; b \,+\, 2^c b^2 \,+\, 3^c b^3 \,+\, \cdots \,+\, n^c b^n$$

The strategy is *recursive*, in that it computes a value for $S_b^{(c)}(n)$ from values for $S_b^{(c-1)}(n), S_b^{(c-2)}(n), \ldots, S_b^{(0)}(n)$. It proceeds in three steps.

Step 1. As in the case $c = 1$, we write summation $S_b^{(c)}(n)$ in two ways. The expression that embodies the first way isolates the summation's first term:

$$S_b^{(c)}(n+1) \;=\; b + \sum_{i=1}^{n} (i+1)^c b^{i+1}$$

The expression that embodies the second way isolates the summation's last term:

$$S_b^{(c)}(n+1) \;=\; S_b^{(c)}(n) \;+\; (n+1)^c b^{n+1}.$$

By combining these expressions, we find that

$$S_b^{(c)}(n) \;=\; b \cdot \left(1 \;-\; (n+1)^c b^n \;+\; \sum_{i=1}^{n} (i+1)^c b^i \right) \tag{6.31}$$

Step 2. We next invoke the Restricted Binomial Theorem (Theorem 6.1) to see that

$$(i+1)^c \;=\; i^c \;+\; c \cdot i^{c-1} \;+\; \binom{c}{2} \cdot i^{c-2} \;+\cdots+\; \binom{c}{k} \cdot i^{c-k} \;+\cdots+\; 1$$

We use this expansion of $(i+1)^c$, together with multiple applications of the laws of arithmetic from Section 5.1.2 to verify that

$$\begin{aligned} \sum_{i=1}^{n} (i+1)^c b^i = S_b^{(c)}(n) \;+\; c \cdot S_b^{(c-1)}(n) \;+\; \binom{c}{2} \cdot S_b^{(c-2)}(n) \;+\cdots \\ \cdots+\; \binom{c}{k} \cdot S_b^{(c-k)}(n) \;+\cdots+\; S_b^{(0)}(n) \end{aligned} \tag{6.32}$$

Step 3. We finally combine equations (6.31) and (6.32) to discover that

$$\begin{aligned} (b-1) \cdot S_b^{(c)}(n) = (n+1)^c b^n \;-\; c \cdot S_b^{(c-1)}(n) \;-\; \binom{c}{2} \cdot S_b^{(c-2)}(n) \;-\cdots \\ \cdots-\; \binom{c}{k} \cdot S_b^{(c-k)}(n) \;-\cdots-\; S_b^{(0)}(n) \;-\; 1 \end{aligned} \tag{6.33}$$

We thus have the promised method of evaluating the extended geometric summation $S_b^{(c)}(n)$ associated with the fixed power c in terms of the sums of extended geometric summations associated with smaller fixed powers.

6.3 On Summing "Smooth" Series

6.3.1 Approximate Sums via Integration

This section illustrates a powerful strategy for obtaining nontrivial upper and lower bounds on the values of sum, by finding continuous *envelopes* that bound the discrete summations both above and below. The areas under the enveloping continuous functions—which we can calculate via integration—provide the desired bounds on the summations.

The strategem operates via the following three steps. Say that we have a sum

$$a_1 + a_2 + \cdots + a_n$$

For convenience we use a finite sum for illustration; the stratagem often works with infinite sums also, as our specific examples illustrate.

Step 1. Represent the summands seriatim as abutting unit-width rectangles.

Our generic example has n unit-width rectangles, of respective heights a_1, a_2, ..., a_n. We describe two special cases, to help the reader understand how the strategem is applied.

1. Figs. 6.22 and 6.23 illustrate our strategem applied to the summation $S_2(n) = \sum_{i=1}^{n} i^2$. In both figures, we represent $S_2(n)$ by the aggregate area of a sequence of unit-width rectangles. The rectangles that represent the respective addends in this example have respective heights 1, 4, 9, 16, 25, 36 and 49. If we were to extend the figures rightward (to extend the summation by encompassing more addends thereby increasing n), then the next rectangle would have height 64.

   The rectangles in Fig. 6.22 are accompanied by a continuous curve (labeled (a) in the figure) which connects their upper lefthand corners. Because this curve completely "covers" the rectangles (which we emphasize by shading the rectangles), the area under the curve is an *upper bound* on the aggregate area of the rectangles; this area is

   $$\int_1^n x^2 dx$$

   The rectangles in Fig. 6.23 are accompanied by a continuous curve (labeled (b) in the figure) which connects the upper righthand corners of the rectangles. Because this curve lies completely within the area formed by the rectangles, the area under the curve is a *lower bound* on the aggregate area of the rectangles; this area is

   $$\int_0^{n-1} \frac{1}{x+1} dx$$

2. In Figs. 6.24 and 6.25, we represent the *harmonic* sum

   $$S^{(H)}(n) = \sum_{i=1}^{n} i^{-1} = \sum_{i=1}^{n} 1/i.$$

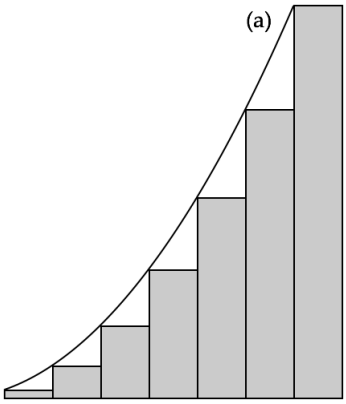


Fig. 6.22 The summation $S_2(n) = \sum_{i=1}^{n} i^2$ represented by the aggregate area of a sequence of unit-width shaded rectangles. The summation is bounded above by the area under the continuous curve (a) that connects the upper lefthand corners of the rectangles. The area under curve (a) is $\int_0^n (x+1)^2 dx$.

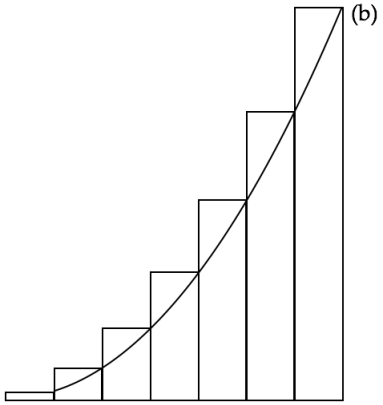


Fig. 6.23 The summation $S_2(n)$ represented by the aggregate area of a sequence of unit-width unshaded rectangles. The summation is bounded from below by the area under the continous curve (b) that connects the upper righthand corners of the rectangles. The area under curve (b) is $\int_1^n x^2 dx$.

In both figures, $S^{(H)}(n)$ is represented by the aggregate area of a sequence of abutting unit-width rectangles, of respective heights 1, $1/2$, $1/3$, ..., $1/10$ (so the figures represent the case $n = 10$). It would be only a clerical task to add more rectangles, of heights $1/11$, $1/12$, etc., to represent larger values of n.

In Fig. 6.24, the rectangles are accompanied by a continuous curve (marked (a) in the figure) that passes through their upper righthand corners. Because this curve completely "covers" the rectangles (which we emphasize by shading the rectangles), the area under the curve is an *upper bound* on the aggregate area of the rectangles; the area under curve (a) is

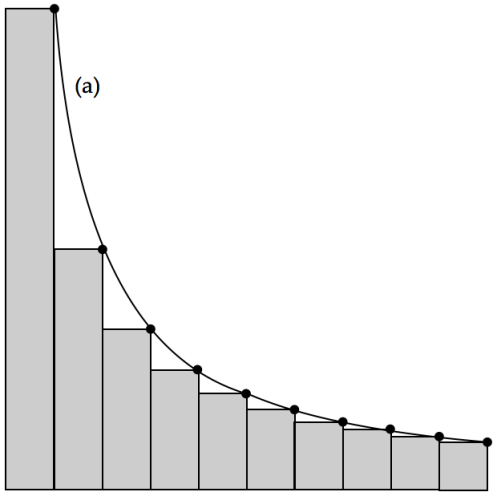


Fig. 6.24 The summation $S^{(H)}(n) = \sum_{i=1}^{n} 1/i$ represented by the aggregate area of a sequence of unit-width rectangles, $S^{(H)}(n)$ is bounded from above by the area under a continuous curve (a) that passes through the upper righthand corners of the rectangles. This area is $\int_1^n \frac{1}{x} dx$.

$$\int_1^n \frac{1}{x} dx$$

In Fig. 6.25, the rectangles are accompanied by a continuous curve (marked (b) in the figure) that passes through their upper lefthand corners. Because curve (b) lies completely within the aggregate area of the rectangles, the area under the curve is a *lower bound* on the aggregate area of the rectangles; the area under curve (b) is

$$\int_0^{n-1} \frac{1}{x+1} dx$$

Step 2. Construct a continuous curve $\overline{C}(x)$ that passes through the corners of the unit-width rectangles specified by the summation, in such a way that the aggregate areas of the rectangles lies completely within the area under $\overline{C}(x)$.

≈≈≈≈≈≈≈≈≈≈

The curves labeled (a) in Figs. 6.22 and 6.24 are instances of the mandated continuous curve $\overline{C}(x)$.

≈≈≈≈≈≈≈≈≈≈

Because the aggregate areas of the abutting rectangles lies completely under curve $\overline{C}(x)$, the area under the curve—which we obtain by integrating $\overline{C}$ between limits appropriate for the summation—affords *an upper bound* on the value of the summation of interest.

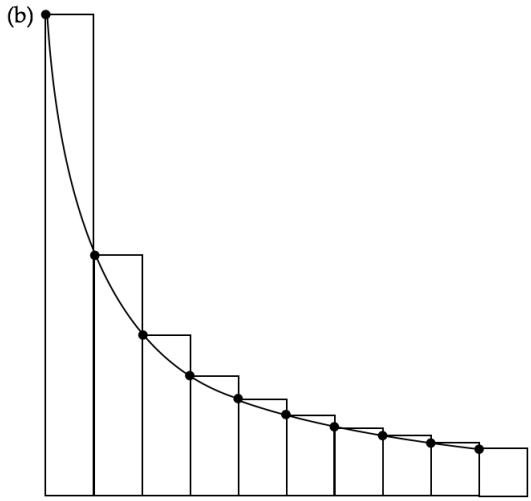


Fig. 6.25 Representing the summation $S^{(H)}(n)$ as in Fig. 6.24, the summation is bounded from below by the area under a continuous curve (b) that passes through the upper lefthand corners of the rectangles. The area under curve (b) is $\int_0^{n-1} \frac{1}{x+1} dx$.

Step 3. Construct a continuous curve $\underline{C}(x)$ that passes through the corners of the unit-width rectangles specified by the summation, in such a way that the area under $\underline{C}(x)$ lies completely within the aggregate areas of the rectangles.

≈≈≈≈≈≈≈≈≈≈

The curves labeled (b) in Figs. 6.23 and 6.25 are instances of the mandated continuous curve $\underline{C}(x)$.

≈≈≈≈≈≈≈≈≈≈

Because the area under the curve $\underline{C}(x)$ lies completely within the aggregate area of the abutting rectangles, the area under the curve—which we obtain by integrating $\underline{C}$—affords *a lower bound* on the summation of interest.

In the next subsection, we apply this strategem to summations of fixed powers of successive integers—i.e., summations of the form $S_c(n) \stackrel{\text{def}}{=} \sum_{i=1}^{n} i^c$—for various (classes of) values of the fixed power c.

6.3.2 Sums of Fixed Powers of Consecutive Integers: $\sum i^c$

We obtain bounds on the summations $S_c(n)$ that are rather good for large values of n. In special cases, our bounds are good, sometimes even exact, for all values of n.

6.3.2.1 $S_c(n)$ for general *nonnegative* real cth powers

We begin to illustrate the technique of bounding summations via integrals by focusing on summations of the form

$$S_c(n) \stackrel{\text{def}}{=} \sum_{i=1}^{n} i^c,$$

for arbitrary positive numbers c. The reader can garner intuition for the upcoming bounds from the general shape of the rectangles and continuous curves in Figs. 6.22 and 6.23 We obtain our upper bound on $S_c(n)$ by evaluating the integral that yields the area $\overline{C}_c(n)$ under the lefthand continuous curve (a) in Fig. 6.22, namely,

$$\begin{aligned} \overline{C}_c(n) \;=\; \int_0^n (x+1)^c dx &= \frac{1}{c+1}(n+1)^{c+1} \;+\; O(1) \\ &= \frac{1}{c+1}n^{c+1} \;+\; O(n^c). \end{aligned} \tag{6.34}$$

We obtain our lower bound on $S_c(n)$ by evaluating the integral that yields the area $\underline{C}_c(n)$ under the righthand continuous curve (b) in Fig. 6.23, namely,

$$\underline{C}_c(n) \;=\; \int_1^n x^c dx \;=\; \frac{1}{c+1}n^{c+1} \;+\; O(1). \tag{6.35}$$

Combining these bounds, we finally have the following two-sided bound on $S_c(n)$.

$$\frac{1}{c+1}n^{c+1} \;+\; O(1) \;\;\leq\;\; S_c(n) \;\;\leq\;\; \frac{1}{c+1}n^{c+1} \;+\; O(n^c). \tag{6.36}$$

The main message here is:

The behavior of $S_c(n)$ as a function of n is dominated by $\frac{1}{c+1}n^{c+1}$ as n grows without bound.

6.3.2.2 Nonnegative integer cth powers

A. A better bound via the Binomial Theorem

When c is a positive integer, the following special case of Newton's *Binomial Theorem*.[4] affords us a much more detailed version of the upper bound (6.34) on the sum $S_c(n)$.

Theorem 6.1 (The Restricted Binomial Theorem). *For all positive integers k,*[5]

[4] The general form of the Binomial Theorem expands the polynomial $(x+y)^k$ rather than $(x+1)^k$. See Section 5.3.3.1.

[5] See (5.3) for the definition of, and notation for, the binomial coefficient $\binom{k}{i}$.

$$(x+1)^k \;=\; \sum_{i=0}^{k} \binom{k}{i} x^{k-i}. \tag{6.37}$$

We obtain our improved upper bound on $S_c(n)$ by parallelling the reasoning that led us to the relation (6.34). Our improved upper bound emerges also by evaluating the integral that yields the area $\overline{C}_c(n)$ under the continuous curve that passes through the upper lefthand corners of the unit-width rectangles specified by summation $S_c(n)$.

$$\begin{aligned} \overline{C}_c(n) \;=\; \int_0^n (x+1)^c dx &= \int_0^n \left(\sum_{i=0}^{c} \binom{c}{i} x^{c-i} \right) dx \\ &= \sum_{i=0}^{c} \left(\int_0^n \binom{c}{i} x^{c-i} \right) dx \\ &= \sum_{i=0}^{c} \frac{1}{c-i+1} \binom{c}{i} n^{c-i+1} \;+\; O(1) \end{aligned} \tag{6.38}$$

This is a proper upper bound because the region defined by this curve totally contains the region subtended by the rectangles.
Using this strategy, we find that for any positive integer c, summation $S_c(n)$ enjoys the following two-sided bound:

$$\frac{1}{c+1} n^{c+1} \;+\; O(1) \;\leq\; \sum_{i=1}^{n} i^c \;\leq\; \sum_{i=0}^{c} \frac{1}{c-i+1} \binom{c}{i} n^{c-i+1} \;+\; O(1) \tag{6.39}$$

We have, of course, not changed the dominant behavior of $S_c(n)$ as n grows without bound, but we have taken a substantial step toward developing explicit expressions for the summations $S_c(n)$ when c is a positive integer.

B. Using *undetermined coefficients* to refine sums

We now introduce the *Method of Undetermined Coefficients* and illustrate how to use it to derive explicit expressions for the sums $S_c(n)$ when c is a positive integer. Our development builds on the intuition garnered from the bounds (6.39) that

$$S_c(n) \;=\; \frac{1}{c+1} n^{c+1} \;+\; a_c^{(c)} n^c \;+\; a_{c-1}^{(c)} n^{c-1} \;+\; \cdots \;+\; a_2^{(c)} n^2 \;+\; a_1^{(c)} n + a_0^{(c)}$$

for some nonnegative numbers $a_c^{(c)}, \ldots, a_0^{(c)}$. To begin, we know that $a_0^{(c)} = 0$, because $S_c(0) = 0$.

$$\approx\approx\approx\approx\approx\approx\approx\approx\approx$$

Because we are beginning with a conjecture based on intuition, we will have to verify the explicit expressions that we derive. We do this after deriving our expressions.

$$\approx\approx\approx\approx\approx\approx\approx\approx\approx\approx$$

Because the Method becomes computationally cumbersome for large values of c, we introduce the reader to it via the first few integer values of c.

The case $c = 1$. We begin with the sum $S_1(n)$, whose value we already know. Reasoning from the case $c = 1$ of (6.39), we intuit that

$$S_1(n) \;=\; \frac{1}{2}n^2 \;+\; a_1^{(1)}n$$

for some positive *undetermined coefficient* $a_1^{(1)}$. We can discover the value of the single unknown, $a_1^{(1)}$ by evaluating $S_1(n)$ at any single value for the variable n. Any value of n will work; using the *smallest* one, $n = 1$, simplifies our calculation.

Because $S_1(1) = 1$, we have

$$S_1(1) \;=\; 1 \;=\; \frac{1}{2} \;+\; a_1^{(1)}.$$

Therefore, $a_1^{(1)} = 1/2$, which gives us yet one more derivation of the value

$$S_1(n) \;=\; \frac{1}{2}\left(n^2+n\right) \;=\; \frac{n(n+1)}{2}.$$

The case $c = 2$. We derive an explicit expression for $S_2(n) \stackrel{\text{def}}{=} 1+4+\cdots+n^2$.

Proposition 6.9 For all $n \in \mathbb{N}$,

$$S_2(n) \stackrel{def}{=} \sum_{i=1}^{n} i^2 \;=\; \frac{1}{3}n^3 \;+\; \frac{1}{2}n^2 \;+\; \frac{1}{6}n \tag{6.40}$$

$S_2(n)$ is often expressed in a more aesthetic form:

$$S_2(n) \;=\; \frac{1}{6}n(n+1)(2n+1) \;=\; \frac{2n+1}{3}\cdot\binom{n}{2}.$$

Proof. Reasoning from the case $c = 2$ of (6.39), we propose the conjecture that

$$S_2(n) \;=\; \sum_{i=0}^{n} i^2 \;=\; \frac{1}{3}n^3 + a_2^{(2)}n^2 + a_1^{(2)}n. \tag{6.41}$$

for some positive *undetermined coefficients* $a_2^{(2)}$ and $a_1^{(2)}$. We thereby express $S_2(n)$ as a polynomial in two unknowns, $a_2^{(2)}$ and $a_1^{(2)}$. We can determine values for the unknowns by instantiating the polynomial with (any) two values of n; to simplify calculations, we select the smallest two values of n, namely, $n = 1, 2$. These instantiations of the polynomial leave us with the following pair of linear equations.

$$n = 1 : \sum_{i=0}^{1} i^2 = 1 = 1/3 + a_2^{(2)} + a_1^{(2)}$$

$$n = 2 : \sum_{i=0}^{2} i^2 = 5 = 8/3 + 4a_2^{(2)} + 2a_1^{(2)}$$

By elementary arithmetic, these equations simplify to yield the pair

$$a_2^{(2)} + a_1^{(2)} = 2/3$$

$$2a_2^{(2)} + a_1^{(2)} = 7/6$$

These equations reveal that

$$2/3 - a_2^{(2)} = 7/6 - 2a_2^{(2)}$$

so that

$$a_2^{(2)} = 1/2$$

which means that

$$a_1^{(2)} = 1/6.$$

We have, thus, derived equation (6.40).

We verify our expressions for $S_1(n)$ and $S_2(n)$ by induction in subsection C.

With more (calculational) work, but no new (mathematical) ideas, one can derive explicit expressions for the sum of the first n cth powers, i.e., the sum $S_c(n)$, for any positive integer c.

C. Validating approximate summations via induction

We employ the proof technique of (Finite) Induction by proving the correctness of three summation formulas that have occupied our attention in this chapter:

1. the sum $S_1(n)$ of the first n positive integers; cf., equation (6.5)
2. the sum $S_2(n)$ of the squares of the first n positive integers; cf., equation (6.40)
3. the sum of the first n odd positive integers; cf., Proposition 6.3.

We deal with these formulas in turn.
1. Verifying equation (6.5) for $S_1(n)$. For every positive integer m, let $\mathbf{P}_1(m)$ be the proposition

$$1 + 2 + \cdots + m = \binom{m+1}{2}.$$

We proceed according to the standard format of an inductive argument.

The base case $\mathsf{P}_1(1)$. Because $\binom{2}{2} = 1$, proposition $\mathbf{P}_1(1)$ is true.

The inductive hypothesis. Assume, for the sake of induction, that proposition $\mathbf{P}_1(m)$ is true for every positive integer $m < n$. In particular, then, proposition $\mathbf{P}_1(n-1)$ is true.

Extending the induction. Because proposition $\mathbf{P}_1(n-1)$ is true, we know that

$$S_1(n-1) \;=\; 1+2+\cdots+(n-1) \;=\; \binom{n}{2}.$$

By direct calculation, then,

$$\begin{aligned} S_1(n) &= \binom{n}{2} + n \\ &= \frac{n(n-1)}{2} \;+\; n \\ &= \frac{n^2+n}{2} \\ &= \binom{n+1}{2}, \end{aligned}$$

as was verified in equation (6.5).

Because n is an arbitrary positive integer, we conclude that $\mathbf{P}_1(n)$ is true whenever

- $\mathbf{P}_1(1)$ is true
- *and* $\mathbf{P}_1(m)$ is true for all $m < n$.

By the Principle of (Finite) Induction, then, we conclude that proposition $\mathbf{P}_1(n)$ is true for all positive integers n. □

2. Verifying equation (6.40) for $S_2(n)$. For every positive integer m, let $\mathbf{P}_2(m)$ be the proposition

$$1 \;+\; 2^2 \;+\; 3^2 \;+\cdots+\; m^2 \;=\; \frac{1}{6}m(m+1)(2m+1).$$

We proceed according to the standard format of an inductive argument.

The base case $\mathsf{P}_2(1)$. Because $\frac{1}{6}(2\cdot 3) = 1$, proposition $\mathbf{P}_2(1)$ is true.

The inductive hypothesis. Assume, for the sake of induction, that proposition $\mathbf{P}_2(m)$ is true for every positive integer $m < n$. In particular, then, proposition $\mathbf{P}_2(n-1)$ is true.

Extending the induction. Because proposition $\mathbf{P}_2(n-1)$ is true, we know that

$$S_2(n-1) \;=\; \frac{1}{6}(n-1)\cdot n\cdot(2n-1).$$

By direct calculation, then,

$$\begin{aligned} S_2(n) &= \frac{1}{6}(n-1)\cdot n\cdot(2n-1) \;+\; n^2 \\ &= \frac{n}{6}\left((n-1)\cdot(2n-1)+6n\right) \\ &= \frac{n}{6}\left(2n^2+3n+1\right) \\ &= \frac{n}{6}(n+1)(2n+1), \end{aligned}$$

as was verified in equation (6.40).

Because n is an arbitrary positive integer, we conclude that $\mathbf{P}_2(n)$ is true whenever

- $\mathbf{P}_2(1)$ is true
- *and* $\mathbf{P}_2(m)$ is true for all $m < n$.

By the Principle of (Finite) Induction, then, we conclude that proposition $\mathbf{P}_2(n)$ is true for all positive integers n. □

3. Verifying that each perfect square n^2 is the sum of the first n odd integers. We turn finally to the assertion that, for every positive integer n,

$$n^2 \;=\; 1 \;+\; 3 \;+\; 5 \;+\cdots+\; 2n-1.$$

For each positive integer n, let $\mathbf{P}(n)$ denote the proposition that the preceding equation holds.

The following inductive proof complements the constructive proofs of the same result in Proposition 6.3. We proceed according to the standard format of an inductive argument.

The base case $\mathsf{P}(1)$. Because 1 is a perfect square, proposition $\mathbf{P}(1)$ is true.

The inductive hypothesis. Assume, for the sake of induction, that proposition $\mathbf{P}(m)$ is true for every positive integer $m < n$. In particular, then, proposition $\mathbf{P}(n-1)$ is true.

Extending the induction. Because proposition $\mathbf{P}(n-1)$ is true, we know that

$$1+3+5+\cdots+2n-3+2n-1 \;=\; (n-1)^2+2n-1.$$

By direct calculation, we see that

$$(n-1)^2+2n-1 \;=\; (n^2-2n+1)+(2n-1) \;=\; n^2.$$

Because n is an arbitrary positive integer, we conclude that $\mathbf{P}(n)$ is true whenever

- $\mathbf{P}(1)$ is true
- *and* $\mathbf{P}(m)$ is true for all $m < n$.

By the Principle of (Finite) Induction, then, we conclude that $\mathbf{P}(n)$ is true for all $n \in \mathbb{N}^+$. □

6.3.2.3 $S_c(n)$ for general *negative* cth powers

We focus finally on summations of the form

$$S_c(n) \stackrel{\text{def}}{=} \sum_{i=1}^{n} i^c,$$

for arbitrary *negative* numbers c. The reader can garner intuition for the upcoming bounds from the general shape of the rectangles and continuous curves in Figs. 6.24 and 6.25.

Proposition 6.10 *For summations $S_c(n)$ with fixed negative powers $c < 0$,*

$$\left[\underline{C}_c(n) \;=\; \int_0^{n-1} (x+1)^c dx\right] \;\leq\; S_c(n) \;\leq\; \left[\overline{C}_c(n) \;=\; \int_1^{n} x^c dx\right]. \tag{6.42}$$

Proof. We obtain our upper bound on the sum of $S_c(n)$ by evaluating the integral that yields the area $\overline{C}_c(n)$ under the righthand continuous curve (a) in the analogue of Fig. 6.24 for $S_c(n)$. We obtain our lower bound on $S_c(n)$ by evaluating the integral that yields the area $\underline{C}_c(n)$ under the lefthand continuous curve (b) in the analogue of Fig. 6.25 for $S_c(n)$.

When $c \neq -1$,[6] we can provide more detail, using reasoning similar to that underlying the bounds (6.36) that hold for positive values of c.

A. Negative powers c with $-1 < c < 0$

In this case, we obtain essentially the same bounds as in the case of nonnegative c. To wit,

Proposition 6.11 *For sums $S_c(n)$ with fixed negative powers in the range $-1 < c < 0$,*

For $-1 < c < 0$:

$$\frac{1}{c+1}n^{c+1} \;-\; O(n^c) \;\leq\; S_c(n) \;\leq\; \frac{1}{c+1}n^{c+1} \;+\; O(1). \tag{6.43}$$

The infinite version of summation $S_c(n)$, namely, the series

$$S_c^{(\infty)} \stackrel{\text{def}}{=} \sum_{i=0}^{\infty} i^c$$

diverges.

We thus observe that $S_c(n)$ has the same growth *pattern* as n grows as it does when c is positive, but that $S_c(n)$'s growth *rate* is slower because of the damping

[6] We need to avoid the case $c = -1$ so that we do not attempt to divide by 0.

effect the negative c in the exponent. This damped growth notwithstanding, the infinite series $S_c^{(\infty)}$ diverges because n^{c+1}, which is the variable portion of the lower bound on $S_c(n)$, grows without bound as n grows without bound.

B. Negative powers c with $c < -1$

When c is "very negative", specifically, when $c < -1$, then the infinite version of $S_c(n)$, call it $S_c^{(\infty)}$, is a *convergent* infinite series. Because n^c *shrinks* in this case as n grows, an analysis mirroring the one that leads to (6.43) provides the following sum for $S_c^{(\infty)}$.

Proposition 6.12 *When the fixed negative power c is smaller than -1, then the infinite version, $S_c^{(\infty)}$, of $S_c(n)$, converges, with the following sum.*

$$S_c^{(\infty)} \;=\; \frac{1}{c+1} \tag{6.44}$$

Proof. We see as in (6.43) that, for $c < -1$, as n grows without bound, $S_c(n)$ tends to the value $\dfrac{1}{c+1}$.

C. Negative powers c with $c = -1$: the *harmonic* summation

The singular case defined by the value $c = -1$ defines the important *harmonic series*,

$$S^{(H)} \;=\; \sum_{i=1}^{\infty} \frac{1}{i}$$

and its finite prefixes that comprise the *harmonic summation*

$$S^{(H)}(n) \;=\; \sum_{i=1}^{n} \frac{1}{i}$$

(i) The asymptotic behavior of $S^{(H)}(n)$. It has been known since the time of the well-traveled Swiss mathematician Leonhard Euler that $S^{(H)}$ and $S^{(H)}(n)$ are closely related to the *natural*, or, *Napierian*,[7] logarithm $\ln n$, *i.e.*, the logarithm whose base is Euler's constant $e = 2.718281828\ldots$.

Proposition 6.13 *The behavior of the harmonic summation $S^{(H)}(n)$ as a function of n is given by*

$$S^{(H)}(n) \;\approx\; \ln n.$$

It follows, in particular, that the harmonic series $S^{(H)}$ diverges.

[7] The natural logarithm, i.e., the logarithm to the base e, is commonly referred to as the *Napierian logarithm*, in honor of the Scottish polymath John Napier.

≈≈≈≈≈≈≈≈≈≈

The adjective "harmonic" calls to mind a number of concepts associated with music*, such as "harmonics" and "harmony". The association between our series and these musical concepts is not a coincidence. The name of the harmonic series derives from the concept of* overtones*, or* harmonics*, in music. When one observes a vibrating string, one finds that the wavelengths of its overtones, as fractions of the string's fundamental wavelength, are the terms of the* harmonic sequence*, namely,* $\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$.

≈≈≈≈≈≈≈≈≈≈

(ii) Bounds on the asymptotic behavior of $S^{(H)}(n)$. Fig 6.26 depicts the harmonic summation $S^{(H)}(n)$ as the area of abutting unit-width rectangles of respective heights (from left to right) of $1, 1/2, 1/3, \ldots, 1/n$.

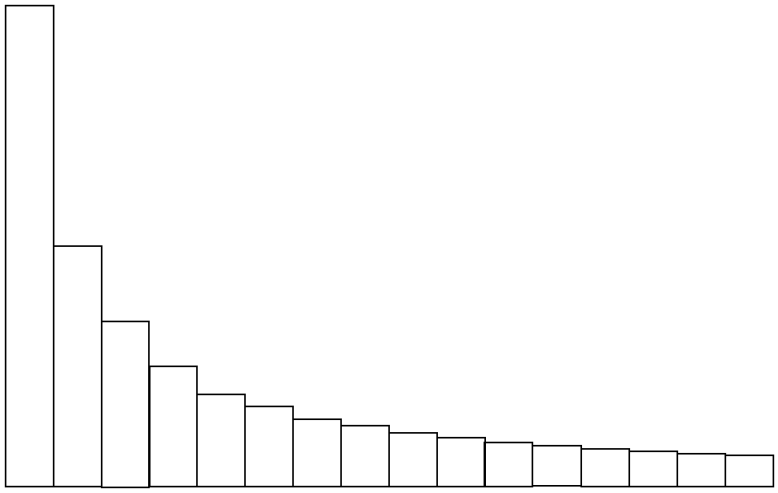


Fig. 6.26 The harmonic summation $S^{(H)}(n)$ represented by the area of abutting unit-width rectangles of decreasing heights.

In order to better understand the behavior of $S^{(H)}(n)$ as a function of n, let us group the summation's consecutive terms into subsums composed from groups of summands whose sizes are consecutive powers of 2:

$$(1) + \left(\frac{1}{2}+\frac{1}{3}\right) + \left(\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\frac{1}{7}\right) + \left(\frac{1}{8}+\frac{1}{9}+\frac{1}{10}+\frac{1}{11}+\frac{1}{12}+\frac{1}{13}+\frac{1}{14}+\frac{1}{15}\right) + \cdots$$

Next, we isolate each grouped subsum and list the subsums in order of size, measured as number of inverse-integer summands.

$$\begin{array}{ll}
\text{Sum of } (2^0 = 1) \text{ consecutive inverses:} & A_0 = 1 \\
\text{Sum of } (2^1 = 2) \text{ consecutive inverses:} & A_1 = \frac{1}{2} + \frac{1}{3} \\
\text{Sum of } (2^2 = 4) \text{ consecutive inverses:} & A_2 = \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} \\
& \vdots \quad \vdots \\
\text{Sum of } 2^i \text{ consecutive inverses:} & A_i = \frac{1}{2^i} + \frac{1}{2^i + 1} + \cdots + \frac{1}{2^{i+1} - 1} \\
& \vdots \quad \vdots
\end{array}$$

Finally, we derive absolute-constant upper and lower bounds that hold for all of the subsums. To derive these bounds, we focus on the generic subsum A_i, which consists of 2^i summands. When we focus on the largest and smallest inverse-integers in A_i—which are, respectively, $1/2^i$ and $1/(2^{i+1}-1)$—we note the following absolute bounds.

$$\frac{1}{2} \;<\; \frac{2^i}{2^{i+1}-1} \;<\; 2^i \cdot A_i \;<\; \frac{2^i}{2^i} \;=\; 1$$

We thereby have absolute constant upper and lower bounds on every subsum A_i.

Referring back to Fig. 6.26, what the just-derived bounds mean is the following. Let us proceed left to right along the abutting rectangles in the figure, and let us recall, from Section 5.4.B, the definition of "logarithm to the base b". As we double the number of rectangles we have traversed:

1. We increase the aggregate area of the thus-far traversed rectangles by at most 1.
   This means that $S^{(H)}(n)$ *grows no faster than* $\log_2 n$.
2. We increase the aggregate area of the thus-far traversed rectangles by more than $1/2$.
   This means that $S^{(H)}(n)$ *grows faster than* $\log_4 n$.

Of course, these observations are consistent with the verified actual natural-logarithmic growth rate of $S^{(H)}(n)$, because $2 < e < 4$.

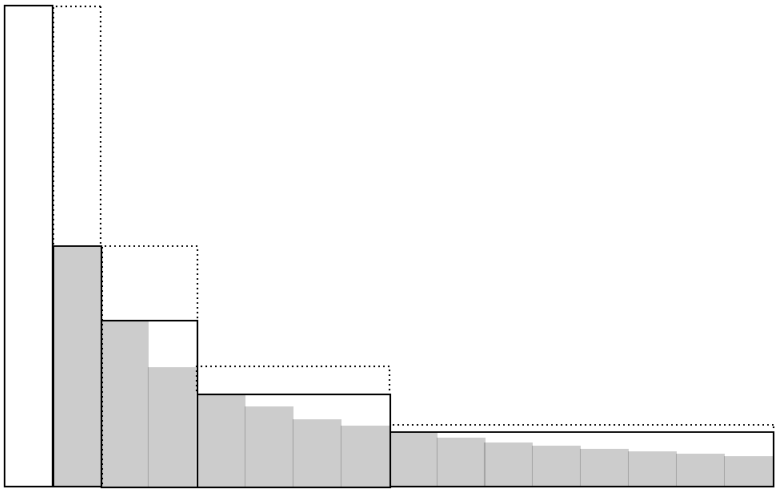


Fig. 6.27 Upper bound of A_i by larger unit-size rectangles in the harmonic sum.

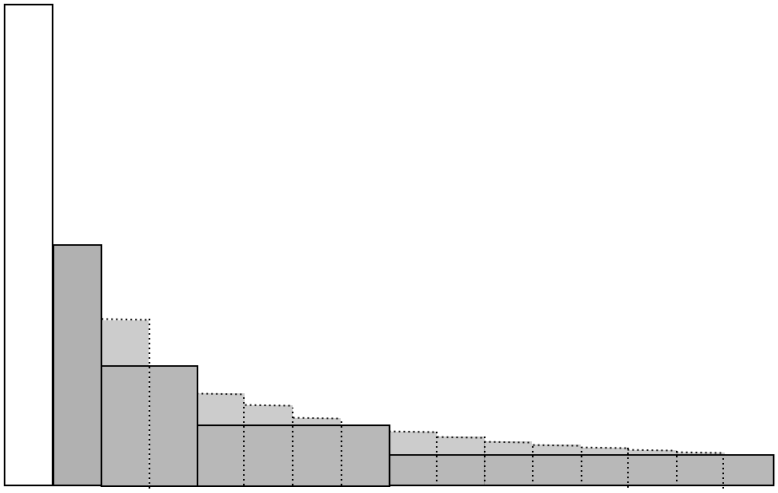


Fig. 6.28 A pictorial lower bound for the harmonic summation.

Chapter 7
NUMBERS II: BEYOND THE BASICS

7.1 Introduction

Chapter 4 was devoted to establishing the mathematical basics of the most familiar objects of mathematical discourse, numbers and the numerals that we use to manipulate them. The current chapter builds on those basics with the help of the material in the intervening chapters, which have given us advanced tools for discussing and manipulating numbers and aggregations of numbers. We focus on three advanced subjects. In Section 7.2, we develop a number of important topics concerning the *prime numbers*, a set of integers that can aptly be termed the *building blocks of the integers*. In Section 7.3, we focus on the important topic of *pairing functions*. These functions allow us, in both theory and practice, to mathematically and computationally treat tuples of numbers—as well as many other aggregates—with the same ease as we treat ordinary numbers. One particularly important contribution of pairing functions is their endowing tuples and other aggregates of numbers with a natural *total order*. Finally, in Section 7.4, we establish the elements of *finite number systems*. We use such systems every day, as we tell time and measure angles: It is important to understand the ways in which such systems mirror our ore familiar infinite number systems, and in which ways they do not.

7.2 Prime Numbers: Building Blocks of the Integers

We single out a subclass of the positive integers whose mathematical importance has been recognized for millennia but which have found important new applications (e.g., within the domain of computer security) as recently as within the past several decades. This subclass is defined by its divisibility characteristics.

Note that every positive integer n is divisible by 1 and by n. The subclass of interest consist of those n that have no other divisors.

An integer $p > 1$ is *prime* if its *only* positive integer divisors are 1 (which divides every integer) and itself (which is always a divisor).

≈≈≈≈≈≈≈≈≈≈

We often use the shorthand assertion, "p is a prime" (or even the simpler "p is prime") instead of the longer, but equivalent, "p is a prime integer."

≈≈≈≈≈≈≈≈≈≈

7.2.1 The Fundamental Theorem of Arithmetic

7.2.1.1 Statement and proof

A very consequential way to classify a positive integer n is to list the primes that divide it, coupling each such prime p with its *multiplicity*, i.e., the number of times that p divides n. Let $p_1, p_2, \ldots, p_r$ be all of the distinct primes that divide n, and let each p_i divide n with multiplicity m_i. The *prime factorization* of n is the product $p_1^{m_1} \cdot p_2^{m_2} \cdot \cdots \cdot p_r^{m_r}$; note that this product satisfies the equation

$$n = p_1^{m_1} \cdot p_2^{m_2} \cdot \cdots \cdot p_r^{m_r} \tag{7.1}$$

When writing an integer n's prime factorization, it is traditional to write the factorization in *canonical form*, i.e., with the primes $p_1, p_2, \ldots, p_r$ that divide n listed in increasing order, i.e., so that $p_1 < p_2 < \cdots < p_r$.

A positive integer n is totally characterized by its canonical prime factorization, as attested to by the following classical theorem, which has been known for millennia and has been honored with the title *The Fundamental Theorem of Arithmetic*. We state the Theorem in two equivalent ways which suggest somewhat different ways of thinking about the result.

Theorem 7.1 (The Fundamental Theorem of Arithmetic).
(Traditional formulation.) *The canonical prime factorization of every positive integer is unique.*
(Alternative formulation.) *Let $n \in \mathbb{N}^+$ be a positive integer, and let $\widehat{P}_n$ denote the ordered sequence of prime numbers that are no larger than n:*

$\widehat{P}_n = \langle P_1, P_3, \ldots, P_{r-1}, P_r \rangle$
where: $P_1 = 2$
each $P_i < P_{i+1}$
$P_r \leq n$.

There exists a unique sequence of nonnegative *integers, $\langle a_1, a_2, \ldots, a_r \rangle$ such that*

$$n = \prod_{i=1}^{r} P_i^{a_i} = P_1^{a_1} \cdot P_2^{a_2} \cdot \cdots \cdot P_{r-1}^{a_{r-1}} \cdot P_r^{a_r}$$

A simple, yet important, corollary of Theorem 7.1 is the following result, whose proof we leave to the reader.

Proposition 7.1 *Every integer $n > 1$ is divisible by at least one prime number.*

Proving the Fundamental Theorem of Arithmetic. The proof of Theorem 7.1 is actually rather elementary, providing that one approaches it gradually. It employs a lot of important techniques and concepts involved in "doing mathematics", as discussed in the eponymous Chapter 2.

We begin with a purely technical result.

Proposition 7.2 *Let p be a prime, and let m be any positive integer that is* not *divisible by p. There exist integers a, b, not necessarily positive, such that*

$$ap + bm \;=\; 1.$$

Proof. This result is a special case of Proposition 4.4 because for any prime p and integer m that is not divisible by p, $\text{GCD}(p,m) = 1$. □

Proposition 7.3 *If the prime p divides a composite number $m \cdot n$, then either p divides m, or p divides n, or both.*[1]

Proof. Let p, m, and n be as asserted, and say that p does not divide m. By Proposition 7.2, then, there exist integers a, b, not necessarily positive, such that

$$ap + bm \;=\; 1.$$

Let us multiply both sides of this equation by n. After some manipulation—specfically, applying the distributive law—we find that

$$apn + bmn \;=\; n.$$

Now, p divides the expression to the left of the equal sign: p divides p by definition, and p divides mn by assumption. It follows that p must divide the expression to the right of the equal sign—namely, the integer n. □

We are finally ready to develop the proof of the Fundamental Theorem.

Proof. The Fundamental Theorem of Arithmetic. Our dominant tool for proving Theorem 7.1 will be *proof by contradiction* (see Chapter 2.2.4). We assume, for the sake of contradiction, that there is a positive integer n that has two distinct canonical prime factorizations.

Our argument will be a trifle simpler if we employ the *alternative* form of the Theorem. To this end, let

$$P_1 \;<\; P_2 \;<\cdots<\; P_{r-1} \;<\; P_r$$

[1] The closing phease "or both" signals our use of the *inclusive* or.

denote, in increasing order, the set of all primes that do not exceed n; i.e., every $P_i \leq n$.

The fact that n has two distinct canonical prime factorizations manifests itself, in this formulation, by the assumption that there exist *two* distinct sequences of *nonnegative* integers,

$$\langle a_1, a_2, \ldots, a_r \rangle \quad \text{and} \quad \langle b_1, b_2, \ldots, b_r \rangle$$

such that n is expressible by—i.e., is equal to—both of the following products of the primes $P_1, P_2, \ldots, P_{r-1}, P_r$.

$$P_1^{a_1} \cdot P_2^{a_2} \cdot \cdots \cdot P_{r-1}^{a_{r-1}} \cdot P_r^{a_r} \tag{7.2}$$

$$P_1^{b_1} \cdot P_2^{b_2} \cdot \cdots \cdot P_{r-1}^{b_{r-1}} \cdot P_r^{b_r} \tag{7.3}$$

Let us now "cancel" from the products (7.2) and (7.3) the longest common prefix. Because the two products are, by hypothesis, distinct, at least one of them will not be reduced to 1 by this cancellation. We are, therefore, left with residual products of the forms

$$P_i^{a_i} \cdot X \tag{7.4}$$

$$P_i^{b_i} \cdot Y \tag{7.5}$$

where:

- Precisely one of a_i and b_i equals 0.
  Say, with no loss of generality (because we have no intrinsic way to distinguish the products), that $b_i = 0$ while $a_i \neq 0$.
- Products X and Y are composed only of primes that are strictly bigger than P_i.

Note that

$$P_i^{a_i} \cdot X \;=\; P_i^{b_i} \cdot Y \;=\; Y,$$

because these products result from cancelling the same prefix from the equal products (7.2) and (7.3), and because $b_i = 0$ so that $P_i^{b_i} = 1$.

We have finally reached the point of contradiction.

On the one hand, P_i *must* divide the product Y, because it divides the product $P_i^{a_i} \cdot X$ which equals Y.

On the other hand, P_i *cannot* divide the product Y, because every prime factor of Y is bigger than P_i (and a prime cannot divide a bigger prime).

We conclude that one of the products (7.2) and (7.3) cannot exist, so the theorem must hold. □

7.2.1.2 A "prime" corollary: There are infinitely many primes

The main result of this section, which is traditionally attributed to (our friend, by now) Euclid, , invokes Theorem 7.1 in a crucial way.

Proposition 7.4 *There are infinitely many prime numbers.*

Proof. We know that the first several primes are

$$(P_1 = 2),\ (P_2 = 3),\ (P_3 = 5),\ (P_4 = 7),\ (P_5 = 11), \ldots$$

How far does this sequence extend? Does it ever end?

Let us assume, for the sake of contradiction, that there are only finitely many primes (so that our sequence ends). Say, in particular, that the following r-element sequence of integers enumerates all (and only) primes, in order of magnitude:

Prime-Numbers $= \langle P_1,\ P_2,\ \ldots,\ P_r \rangle$
where $P_1 < P_2 < \cdots < P_{r-1} < P_r$

We verify the *falseness* of the alleged completeness of the sequence **Prime-Numbers** by analyzing the positive integer

$$n^\star \ = \ 1 + \prod_{i=1}^{r} P_i \ = \ 1 \ + \ (P_1 \cdot P_2 \cdot \cdots \cdot P_r).$$

In fact, we claim that $n^\star$ is a prime that is not in the sequence **Prime-Numbers**. We begin to verify our claim by making three crucial observations.

1. *The number $n^\star$ is not divisible by any prime in the sequence* **Prime-Numbers**.
   To see this, note that for each P_k in the sequence,

   $$n^\star / P_k \ = \ \frac{1}{P_k} + \prod_{i \neq k} P_i.$$

   Because $P_k \geq 2$, we see that $n^\star / P_k$ obeys the inequalities

   $$\prod_{i \neq k} P_i \ < \ n^\star / P_k \ < \ 1 + \prod_{i \neq k} P_i.$$

   The discreteness of the set $\mathbb{Z}$—see Section 4.3.1—implies that $n^\star / P_k$ is not an integer, because it lies strictly between two adjacent integers.
2. Because of observation 1, if the sequence **Prime-Numbers** actually did contain *all* of the prime numbers, then we would have to conclude that *the number $n^\star$ is not divisible by any prime number.*
3. Finally, we remark that the Fundamental Theorem of Arithmetic (Theorem 7.1) implies that *every integer $m > 1$ is divisible by (at least one) prime number*.

The preceding chain of assertions leads to a mutual inconsistency. On the one hand, the integer $n^\star > 1$ has no prime-integer divisor. On the other hand, no such integer can fail to have a prime-integer divisor!

Let us analyze how we arrived at this uncomfortable place.

- At the front end of this string of assertions we have the assumption that there are only finitely many prime numbers. We have (as yet) no substantiation for this assertion.

- At the back end of this string of assertions, we have the (*rock solid*) Fundamental Theorem of Arithmetic (Theorem 7.1).
- In between these two assertions we have a sequence of assertions, each of which follows from its predecessors via irrefutable rules of inference.

It follows that the *only* brick in this edifice that could be faulty—i.e., the only assertion that could be false—is the initial assumption, which states that there are only finitely many prime numbers. *We must, therefore, conclude that this vulnerable assumption is false!* In other words, we conclude from this classical proof by contradiction that there are infinitely many prime numbers. □

7.2.1.3 Applying the Theorem in *encryption*

One of the most important applications of Theorem 7.1 is as a mechanism for facilitating *encryption*. While the details of both encryption and the use of prime numbers to that end are beyond the scope of this text, we will provide a peek into that area by means of the following result concerning *encodings* of sequences of positive integers as single integers!

≈≈≈≈≈≈≈≈≈≈

There is a crucial difference between *encoding* and *encryption*, despite the words' often being confused in the vernacular.

Encodings seek representations of objects which achieve some benefit, such as efficient computation or compactness. An example might be the conversion of Roman numerals to positional numerals to enhance the arithmetic operations.

Encryption usually has some notion of secrecy attached. An example might be some key-based cipher which is intended to limit access to some information.

≈≈≈≈≈≈≈≈≈≈

We illustrate (and achieve) the sought encodings as follows. Consider the (infinite) ordered sequence of *all primes:*

$$(P_1 = 2), (P_2 = 3), (P_3 = 5), \ldots$$

Let

$$\bar{s} \;=\; \langle m_1, m_2, \ldots, m_k \rangle \tag{7.6}$$

be an arbitrary sequence of positive integers. Then Theorem 7.1 assures us that the (single) positive integer

$$\iota(\bar{s}) \;=\; P_1^{m_1} \cdot P_2^{m_2} \cdot \;\cdots\; \cdot P_k^{m_k}$$

is a (uniquely decodable) integer-representation of sequence $\bar{s}$.

We return to this idea of encoding-via-integers in a later chapter.

7.2.1.4 ⊕ The "density" of the prime numbers

There are two advanced topics that we may want to mention/discuss: (1) The prime-number theorem ($n/\log n$ primes $\leq n$); (2) the polynomials that generate lots of primes. We should discuss this

7.2.2 Fermat's Little Theorem

A measure of the greatness of the 17th-century French mathematician Pierre de Fermat is that the following fundamental result is called his "little theorem." Aside from its exposing an important and basic property of prime numbers, the theorem provides the basis for a valuable algorithm for testing the primality of integers.

Theorem 7.2 (Fermat's Little Theorem).
Let a be any integer, and let p be any prime.
(Formulation 1): *The number a^p a is divisible by p.*

(Formulation 2): $a^p \equiv a \bmod p$.

We provide two proofs for this fundamental result, each providing rather different insights on the result.

7.2.2.1 A proof using "necklaces"

Proof. The idea underlying this proof is to design a framework in which the result can be reduced to counting a special set of strings.

Letting a and p be as in the theorem, consider the set $S(\mathscr{A}, p)$ of all words/strings of length p over an alphabet/set $\mathscr{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_a\}$ of a symbols. For instance, when $\mathscr{A} = \{0, 1\}$ (so that $a = 2$) and $p = 3$, the set $S(\mathscr{A}, p)$ consists of the words:

$$000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111$$

We begin with some basic definitions and observations.

- The number of words in $S(\mathscr{A}, p)$ is a^p; see Proposition 5.12.
- A *(one-place) circular shift* c of a word in $S(\mathscr{A}, p)$ is accomplished by placing the last symbol of this word into the first position and shifting all other symbols one position rightward. For illustration:

$$c(\alpha_1 \alpha_2 \cdots \alpha_p) \ = \ \alpha_p \alpha_1 \cdots \alpha_{p-1}$$

- By iterating the shift c on a length-p word w at most $p-1$ times, we obtain the *necklace* $\mathscr{N}(w)$, which is the sequence

$$\mathscr{N}(w) \ = \ w,\ c(w),\ c(c(w)), \ldots,\ c(\cdots c(w)) \cdots))$$

in which one further shift would replicate word w.
- The *period* of the necklase $\mathscr{N}(w)$ is the number of words in the sequence.
  Note that *the period of* $\mathscr{N}(w)$ *can never exceed* $p-1$—because an earlier-seen word must recur by the time the length-p word w has been shifted p times.

Consider now a word w that is a *replicate* of another word u, in the sense that $w = uu\cdots u$. Say that u is the shortest word of which w is a replicate and that u has length m. Then:

- *u's length m divides p.*
  This is obvious from our ability to write w in the indicated form.
- *The period of* $\mathscr{N}(w)$ *is* $m-1$.
  This is because, by the time one has shifted w m times, one has transferred a copy of u from the end of w to the beginning. Hence, one has recreated w.
- In our special situation—where w has prime length—*The only candidates for the shortest replicated word u have length* 1 *or p.*
  This is because m divides the prime p.

Summing up, one of the following two situations must hold.
Possibility #1: *The word w has the form* $w = \alpha\alpha\cdots\alpha$ *for some symbol* $\alpha \in \mathscr{A}$.
This can occur in a distinct ways because of $\mathscr{A}$'s cardinality..
Possibility #2: *The word w is not a replicate of any shorter word.*

Because p is a prime, this possibility must hold for every one of the $a^p - a$ words over $\mathscr{A}$ that contains at least 2 distinct symbols. Because the period of any necklace $\mathscr{N}(w)$ for a word that contains at least 2 distinct symbols is exactly $p-1$, the lengths of such necklaces must be exactly p.

This means that the $a^p - a$ words that each contain at least 2 distinct symbols partition $S(\mathscr{A}, p)$ into disjoint sets of size p each. This, in turn, means that p divides $a^p - a$. □

The reader's comprehension of this multi-step proof might be enhanced by Figs. 7.1 and 7.2. These figures jointly depict all necklaces $\mathscr{N}(w)$ for $(p = 3)$-

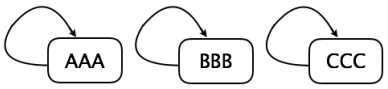


Fig. 7.1 The 3 necklaces composed of the the same symbol ($a = 3$)

letter words over the alphabet $\mathscr{A} = \{A, B, C\}$. Fig. 7.1 depicts the necklaces for words that use only a single letter; Fig. 7.2 depicts the necklaces for words that use at least two distinct letters.

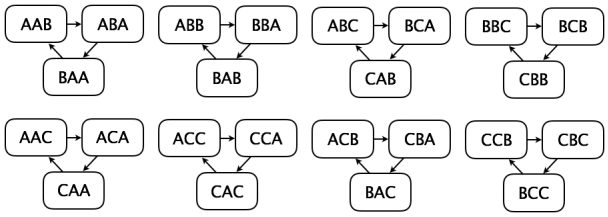


Fig. 7.2 Eight groups of necklaces of size $p = 3$ (for $a = 3$).

7.2.2.2 A proof using the Binomial Theorem

Proof. Our next proof employs Formulation 2 of the Proposition. We focus on a fixed prime p and argue by induction on the alphabet size a, that $a^p \equiv a \bmod p$.
Base of the induction. The base case $a = 1$ is straightforward because $1^p = 1$
Inductive hypothesis. We assume for induction that $a^p \equiv a \bmod p$ for all alphabet sizes not exceeding the integer b.
Extending the induction. Invoking the restricted form of the Binomial Theorem, we know—see (6.37)—that

$$(b+1)^p \;=\; (b^p+1) \;+\; \sum_{i=1}^{p-1} \binom{p}{i} b^{p-i} \tag{7.7}$$

Pondering this equation, we make two important observations.
1. We learn from the development in Section 8.2.1.C that p divides all "internal" binomial coefficients, i.e., all coefficients $\binom{p}{i}$ with $0 < i < p$. This means that there exists an integer n_1 such that

$$\sum_{i=1}^{p-1} \binom{p}{i} b^{p-i} \;=\; p \cdot n_1. \tag{7.8}$$

≈≈≈≈≈≈≈≈≈≈

As an aside, one can observe the just-exposed divisibility property of "internal" binomial coefficients by looking at Pascal's triangle; see the rows corresponding to primes—i.e., rows $n = 2$, $n = 3$, and $n = 5$—in the triangle of Fig. 7.3. (Remember that rows are indexed beginning with $n = 0$.)

≈≈≈≈≈≈≈≈≈≈

2. By the inductive hypothesis, p divides $b^p - b$, which means that there exists an integer n_2 such that

$$b^p \;=\; b \;+\; p \cdot n_2. \tag{7.9}$$

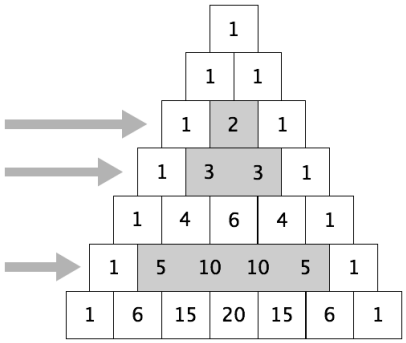


Fig. 7.3 The rows of Pascal's triangle that correspond to $n = 0$, $n = 1$, ..., $n = 6$. The "internal" entries of the rows that correspond to prime numbers—in this case, $n = 2$, $n = 3$, and $n = 5$—are divisible by that number.

When we combine relations (7.8) and (7.9), and we use them to rewrite equation (7.7), we find that

$$(b+1)^p \;=\; (b+1) \;+\; p \cdot (n_1 + n_2).$$

This means that p divides $(b+1)^p \;-\; (b+1)$, which extends the induction and completes the proof. □

7.2.3 ⊕ Mersenne Primes and Perfect Numbers

This short section is dedicated to two related topics whose intrinsic charm has garnered attention from mathematicians who study numbers and their properties for more than three millennia. The section is placed here because of the central role of a particular class of prime numbers in the story we relate here.

7.2.3.1 Perfect numbers

Hearkening back to the ancient Greeks' mystical affinity for special classes of integers, we term a positive integer $n \in \mathbb{N}^+$ *perfect* if n equals the sum of its proper divisors.[2] It is not intuitive that perfect numbers even exist, but only a short search is needed until one realizes that

$$6 \;=\; 1 \cdot 2 \cdot 3 \;=\; 1 + 2 + 3$$

A few more minutes will lead one to the double equation

[2] Euclid refers to perfect numbers in his *Elements*, using adjectives such as "perfect" and "ideal."

$$28 \;=\; 1\cdot 2\cdot 4\cdot 7\cdot 14 \;=\; 1 + 2 + 4 + 7 + 14$$

It may take a bit longer, but the curious reader will eventually discover the double equation

$$\begin{aligned} 496 &= 1\cdot 2\cdot 4\cdot 8\cdot 16\cdot 31\cdot 62\cdot 124\cdot 248 \\ &= 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248 \end{aligned}$$

You may have absorbed enough of our "How to be a mathematician" lore by this point to ask the following "natural" (to a mathematician, at least) questions.

Question #1. *Are there* infinitely many *perfect numbers?*
Answer. We imminently derive a positive response.

Question #2. *Are there any* odd *infinitely many perfect numbers?*
Answer. No one knows—as of 2018.

7.2.3.2 Mersenne primes

A prime number p is a *Mersenne prime*—so named for the 17th-century French monk Marin Mersenne—if it has the form $p = 2^q - 1$ for some integer q. It is obvious that Mersenne primes exist: to wit,

$$\begin{array}{l} \text{The number } \; p = 3 \; \text{ is a Mersenne prime, because } \; 3 \;=\; 2^2 - 1 \\ \text{The number } \; p = 7 \; \text{ is a Mersenne prime, because } \; 7 \;=\; 2^3 - 1 \\ \text{The number } \; p = 31 \; \text{ is a Mersenne prime, because } \; 31 \;=\; 2^5 - 1 \end{array} \tag{7.10}$$

(There are obviously primes—such as 5 and 13—that are not Mersenne primes.) Despite the promising beginning to our list, we cannot extend it too far: To wit, only about 50 Mersenne primes are known! Indeed, it is not known—as of 2018—whether there exist infinitely many Mersenne primes! It is true, though, that *largest known prime*—as of 2018—*is a Mersenne prime:* $2^{77,232,917} - 1$. We close our discussion of Mersenne primes as an isolated topic—i.e., unrelated to perfect numbers—with the following result, which limits one's search for Mersenne primes to expressions with prime powers of 2.

Proposition 7.5 *The integer $2^q - 1$ is prime only if the integer q is prime.*

Proof. Say, for contradiction, that there is a composite number $q = m\cdot n$, with both $m, n > 1$, such that $2^q - 1$ is a prime. We invoke identity (6.18) from Proposition 6.5 to show that $2^q - 1$ is, in fact, *not* prime. We note, by direct calculation, that

$$\begin{aligned} 2^{m\cdot n} - 1 &= (2^m - 1)\cdot \frac{2^{m\cdot n} - 1}{2^m - 1} \\ &= (2^m - 1)\cdot \sum_{i=0}^{n-1} (2^m)^i \end{aligned}$$

$$= (2^m - 1) \cdot \left(1 + (2^m) + (2^m)^2 + \cdots + (2^m)^{n-1}\right) \qquad (7.11)$$

The number $r = 2^q - 1$ is, thus the product of two numbers, both greater than 1 and less than r, hence is not a prime. □

7.2.3.3 Using Mersenne prime to generate perfect numbers

We unite the subjects of the preceding two subsections to derive the basis of a procedure that employs Mersenne primes to generate perfect numbers.

Let us revisit the three sample perfect numbers presented at the beginning of Section 7.2.3.1. As the following table illustrates, all three numbers share the form $2^{p-1} \cdot (2^p - 1)$, where both p and $2^p - 1$ are primes.

$$\begin{array}{rrrrrrrrr} n = & 6 = & 2 \cdot 3 = 2^1 \cdot (2^2 - 1) & \text{so that} & p = 2 & \text{and} & 2^p - 1 = 3 \\ n = & 28 = & 4 \cdot 7 = 2^2 \cdot (2^3 - 1) & \text{so that} & p = 3 & \text{and} & 2^p - 1 = 7 \\ n = 496 = & 16 \cdot 31 = 2^4 \cdot (2^5 - 1) & \text{so that} & p = 5 & \text{and} & 2^p - 1 = 31 \end{array}$$

It was no accident that the three perfect numbers we found have intimate formal relationships with the first three primes. In fact, the result we present now establishes that every Mersenne prime has such a formal relationship with a perfect number—thereby providing us the promised path toward generating perfect numbers.[3]

Proposition 7.6 *For every Mersenne prime $2^p - 1$, the number*

$$2^{p-1} \cdot (2^p - 1) = \binom{2^p}{2} \qquad (7.12)$$

is perfect.

Proof. Focus on the instance of expression (7.12) associated with the Mersenne prime $2^p - 1$. With the aid of Theorem 7.1 (the Fundamental Theorem of Arithmetic), let us enumerate the factors of $2^{p-1} \cdot (2^p - 1)$. Our list consists of two groups:

1. all powers of 2, from $2^0 = 1, \ldots, 2^{p-1}$;

2. all products: $(2^p - 1) \times$(a power of 2 from $2^0 = 1, \ldots, 2^{p-1}$).

Summing all of these factors leads to the expression

$$\sum_{i=0}^{p-1} 2^i + (2^p - 1) \cdot \sum_{i=0}^{p-1} 2^i = 2^p \cdot \sum_{i=0}^{p-1} 2^i = 2^p \cdot (2^p - 1). \qquad (7.13)$$

We derive the ultimate expression in (7.13) from the penultimate expression by invoking Proposition 6.5 for the case $b = 2$.

We thus see that the factors of $n = 2^p \cdot (2^p - 1)$ sum to n, so that n is perfect. □

I am not sure what to do with the binary representation. It is an interesting curiosity ... but is ti more than that?

[3] This result (of course with no mention of Mersenne) appears in Euclid's *Principae IX-36*.

I incorporated the triangular numbers into the statement of the proposition. It is just an interesting aside — no new lessons

7.3 Pairing Functions: Bringing Linear Order to Tuple Spaces

Paraphrasing an oft-used quip by the late stand-up comic Rodney Dangerfield, integers "don't get no respect"! Superficially, it appears that integers are useful for counting things but for little else. The Fundamental Theorem of Arithmetic (Theorem 7.1) hints at the potential importance of the prime numbers, but it does little to inspire respect for the non-prime integers. Once one augments the integers with a bit of structure, *then* one can begin to represent interesting situations.

As we shall see imminently, we can accomplish our goals by focusing on very simple integer-based structures, namely, *ordered pairs of integers*—i.e., the sets $\mathbb{Z} \times \mathbb{Z}$, $\mathbb{N} \times \mathbb{N}$, and $\mathbb{N}^+ \times \mathbb{N}^+$. Using $\mathbb{Z} \times \mathbb{Z}$ as the exemplar of these three sets of ordered pairs of integers:

- Each *element* of $\mathbb{Z} \times \mathbb{Z}$ has the form $\langle a_1, a_2 \rangle$, where a_1 and a_2 belong to $\mathbb{Z}$.
- The *semantics* of this set allow one to
  - *aggregate* any two elements of $\mathbb{Z}$, call them b_1, $\mathscr{B}_2$, and create the ordered pair $\langle b_1, b_2 \rangle \in \mathbb{Z} \times \mathbb{Z}$;
  - given any pair $\langle a_1, a_2 \rangle \in \mathbb{Z} \times \mathbb{Z}$, *select* $a_1 \stackrel{\text{def}}{=} \text{first}(\langle a_1, a_2 \rangle)$ and $a_2 \stackrel{\text{def}}{=} \text{second}(\langle a_1, a_2 \rangle)$.

7.3.1 Encoding Complex Structures via Ordered Pairs

Among the myriad other structures that "contain" integers that one can represent—or, *encode*—via some sort of iterated formation of ordered pairs are the following.
(i) *(Ordered) tuples of integers.* We focus on the set of k-tuples of integers, for any integer $k > 1$. One way to accomplish this is by recursion, with *ordered pair* (the just-described case $k = 2$) as the base case. For any $k > 1$, we represent the k-tuple $\langle a_1, a_2, \ldots, a_k \rangle$ as the ordered pair whose *first* is the ordered $(k-1)$-tuple $\langle a_1, a_2, \ldots, a_{k-1} \rangle$ and whose *second* is the integer a_k:

$$\langle a_1, a_2, \ldots, a_k \rangle \;=\; \langle \langle a_1, a_2, \ldots, a_{k-1} \rangle, a_k \rangle$$

(ii) *Strings of integers.* One way to represent the string of integers $a_1 a_2 \cdots a_n$ using ordered pairs is as follows.

$$a_1 a_2 \cdots a_n \;=\; \langle a_1, \langle a_2, \langle a_3, \ldots, \langle a_{n-2}, a_{n-1} \rangle, a_n \rangle \cdots \rangle \rangle$$

The following small example should make the aggregation perfectly clear (without any possibly confusing dots):

$$a_1a_2a_3a_4 \;=\; \langle a_1, \langle a_2, \langle a_3, a_4\rangle\rangle\rangle$$

(iii) *Binary trees.* For illustration, the *complete binary tree* with leaves a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8 would be represented via the following aggregation of ordered pairs,

$$\langle\langle\langle a_1, a_2\rangle, \langle a_3, a_4\rangle\rangle, \langle\langle a_5, a_6\rangle, \langle a_7, a_8\rangle\rangle\rangle$$

while the *comb-structured binary tree* with the same leaves would be represented via the following aggregation of ordered pairs,

$$\langle a_1, \langle a_2, \langle a_3, \langle a_4, \langle a_5, \langle a_6, \langle a_7, a_8\rangle\rangle\rangle\rangle\rangle\rangle\rangle$$

$$\approx\approx\approx\approx\approx\approx\approx\approx\approx$$

It should be no surprise that a single character string, such as $\langle a_1, \langle a_2, \langle a_3, a_4\rangle\rangle\rangle$, can be used to represent many distinct, but isomorphic *(literally, "same-shaped"), objects—for instance a length-*4 *string and a* 4*-leaf comb-structured binary tree. Indeed, one of the biggest strengths of mathematics is its ability to expose often-unexpected structural similarities.*

$$\approx\approx\approx\approx\approx\approx\approx\approx\approx$$

The rather long preamble to this section has been our attempt to enhance the reputation of the integers—all three sets, $\mathbb{Z}$, $\mathbb{N}$, and $\mathbb{N}^+$—in the eyes of the reader. With that process hopefully begun, we turn now to a demonstration via multiple examples that what we have accomplished has largely been an exercise in form rather than essence. Specifically, we show that one can easily and efficiently "encode" structures exemplified by the ones we have been describing as positive integers!

What does it mean to *encode* one class of entities, A as another class B? Our definition is a rather strict mathematical one. We insist that there exist a *bijection* $F_{A,B}$ that maps A *one-to-one onto* B (cf., Chapter 3.2.4). In other words, when presented with an element $a \in A$, the function $F_{A,B}$ "produces" a unique element $b \in B$, and conversely, when presented with an element $b \in B$, the function $F_{A,B}^{-1}$ "produces" a unique element $a \in A$.

7.3.2 Pairing Functions as Encodings of $\mathbb{N}^+ \times \mathbb{N}^+$ *as* $\mathbb{N}^+$

The remainder of this section is devoted to developing *easily computed* bijections between the set $\mathbb{N}^+ \times \mathbb{N}^+$ and the set $\mathbb{N}^+$ of positive integers. We thereby exhibit easily computed mechanisms for encoding ordered pairs of integers—hence, also, tuples, strings, and binary trees of integers—as single integers. Because of the special role of ordered pairs of integers in our study of encodings of structured sets of integers—they form the fundamental puzzle from whose solution all else will

follow—a special name has been associated with bijections between $\mathbb{N}^+ \times \mathbb{N}^+$ and $\mathbb{N}^+$. These special bijections are called *pairing functions*.

One of the most valuable by-products of encodings provided by pairing functions is that such encodings provide *linear orderings* of the set being encoded. We noted in Section 4.3.A that "order within a number system is among one's biggest friends when reasoning about the numbers within the system." The orderings provided by these encodings are particularly valuable when the structured sets being encoded as integers do not have their own "intrinsic" or "natural" orderings. Included in this category are structures such as tuples, strings, and trees.

≈≈≈≈≈≈≈≈≈≈

Of course some structured sets do have natural, native linear orders: consider, as one such, strings under lexicographic ordering. Even for such sets, we often benefit from having alternative orderings as we design and analyze algorithms on the sets.

≈≈≈≈≈≈≈≈≈≈

We focus on $\mathbb{N}^+$ as the avatar of *integer*, rather than on $\mathbb{Z}$ or $\mathbb{N}$, primarily for definiteness and a bit of clerical simplification. We could easily rewrite this section with a focus on bijections between $\mathbb{Z} \times \mathbb{Z}$ and $\mathbb{Z}$ or on bijections between $\mathbb{N} \times \mathbb{N}$ and $\mathbb{N}$.

We now embark on a very short guided tour that will introduce the reader to three interesting pairing functions.

7.3.2.1 The Diagonal pairing function $\mathscr{D}$

Pairing functions first appeared in the literature early in the 19th century. Perhaps the simplest and "prettiest" such function (since it is a *polynomial*) appears, pictorially, in an 1821 work by the great French mathematician Augustin Cauchy [22]. This *diagonal* pairing function was formally specified a half-century later by the German logician Georg Cantor, whose studies [19, 20] revolutionized how we think about *infinite* sets.

$$\mathscr{D}(x,y) \;=\; \binom{x+y-1}{2} + (1-y) \tag{7.14}$$

($\mathscr{D}$ of course has a twin that exchanges x and y). $\mathscr{D}$'s mapping of $\mathbb{N}^+ \times \mathbb{N}^+$ onto $\mathbb{N}^+$, as depicted in Fig. 7.4, exposes that we can view $\mathscr{D}$'s mapping of $\mathbb{N}^+ \times \mathbb{N}^+$ as a two-step conceptual process:

1. partitioning $\mathbb{N}^+ \times \mathbb{N}^+$ into "diagonal shells" defined as

$$\{\langle x,y\rangle \mid x+y=2\},\ \{\langle x,y\rangle \mid x+y=3\},\ \{\langle x,y\rangle \mid x+y=4\},\ \ldots$$

This is accomplished by the following subexpression in (7.14).

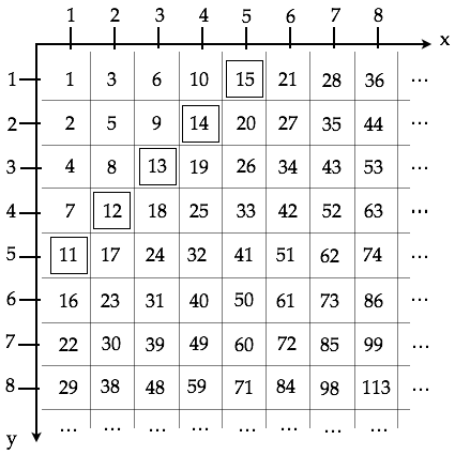


Fig. 7.4 *The diagonal pairing function $\mathscr{D}$. The shell $x+y=6$ is highlighted.*

$$\frac{1}{2}(x+y)\cdot(x+y-1) \;=\; \binom{x+y-1}{2}$$

2. "climbing up" these shells in order.
   This is accomplished by the additional subexpression "$+1-y$" in (7.14).

≈≈≈≈≈≈≈≈≈≈

Keep in mind that we have just described a conceptual *process. $\mathscr{D}$ is a function,* not *an algorithm. "Running" this infinite process would take forever.*

≈≈≈≈≈≈≈≈≈≈

Understanding $\mathscr{D}$'s structure leads to a broadly applicable strategy for inductively constructing a broad range of pairing functions. We develop this strategy in Subsection B, along with an accompanying rather simple inductive verification of bijectiveness. One finds a computationally more satisfying proof of $\mathscr{D}$'s bijectiveness in [31], along with an explicit recipe for computing $\mathscr{D}$'s inverse. The material in [31] builds specifically on $\mathscr{D}$'s structure.

≈≈≈≈≈≈≈≈≈≈

$\oplus$ Esoterica for enrichment: *The fact that $\mathscr{D}$ is a* polynomial *in x and y raises the natural (to a mathematician!) question of whether there exist any other polynomial pairing functions. This is a quite advanced topic that is beyond the scope of this text. Indeed, even as a research problem, the question remains largely open. But, a few nontrivial pieces of an answer are known.*

1. There is no *quadratic* polynomial pairing function other than $\mathscr{D}$ (and its twin) [36, 52].
2. No *cubic* or *quartic* polynomial is a pairing function [53].
3. The development in [53] excludes large families of higher-degree polynomials from being pairing functions; e.g., a *super-quadratic* polynomial whose coefficients are all positive cannot be a pairing function.

$\approx\approx\approx\approx\approx\approx\approx\approx\approx$

7.3.2.2 A methodology for constructing pairing functions

The shell-oriented strategy that underlies the diagonal pairing function $\mathscr{D}$ can be adapted to incorporate shell-"shapes" that are inspired by a variety of computational situations—and can be applied to great computational advantage in such situations. We describe how such adaptation can be effected, and we describe a few explicit shapes and situations. We invite the reader to craft others.

Procedure PF-Constructor($\mathscr{A}$)
/*Construct a shape-inspired pairing function (PF) $\mathscr{A}$*/

Step 1. Partition the set $\mathbb{N}^+ \times \mathbb{N}^+$ into finite sets called *shells*. Order the shells linearly in some way: many natural shell-partitions carry a natural order.

As noted above, Shell c of the diagonal pairing function $\mathscr{D}$ is the following subset of $\mathbb{N}^+ \times \mathbb{N}^+$: $\{\langle x,y\rangle \mid x+y=c\}$. The parameter c orders $\mathscr{D}$'s shells.

Step 2. Construct a pairing function from the shells as follows.

Step 2a. Enumerate $\mathbb{N}^+ \times \mathbb{N}^+$ shell by shell, honoring the ordering of the shells; i.e., list the pairs in shell #1, then shell #2, then shell #3, etc.

Step 2b. Enumerate each shell in some systematic way, e.g., "by columns": Enumerate the pairs $\langle x,y\rangle$ in each shell in increasing order of y and, for pairs having equal y values, in decreasing order of x.

Proposition 7.7 *Any function $\mathscr{A} : \mathbb{N}^+ \times \mathbb{N}^+ \leftrightarrow \mathbb{N}^+$ that is designed via Procedure* PF-Constructor *is a bijection.*

Proof. (Sketch) Step 1 of Procedure PF-Constructor constructs a partial order on $\mathbb{N}^+ \times \mathbb{N}^+$, in which: (*a*) each shell is finite; (*b*) there is a linear order on the shells. Step 2 extends this partial order to a linear order, by honoring the inherent ordering of shells and imposing a linear order within each shell. The function constructed via the Procedure is: *injective* because the disjoint shells are enumerated sequentially; *surjective* because the enumeration within each shell begins immediately after the enumeration within the preceding shell, with no gap. □

We have noted how to use Procedure PF-Constructor to construct pairing function $\mathscr{D}$. We now use the Procedure to design two other useful pairing functions.

7.3.2.3 The Square-shell pairing function $\mathscr{S}$

One computational situation where pairing functions can be useful involves storage-mappings for arrays/tables that can expand and/or contract dynamically. In conventional systems, when one expands an $n \times n$ table into an $(n+1) \times (n+1)$ table, one allocates a new region of $(n+1)^2$ storage locations and copies the current table from its n^2-location region to the new region. Of course, this is very wasteful: one is moving $\Omega(n^2)$ items to make room for the anticipated $2n+1$ new items. On any given day, the practical impact of this waste depends on current technology. But, this is a mathematics text, not an engineering one, so we are exploring whether *in principle* we can avoid the waste. The answer is "YES". If we employ a pairing function $\mathcal{E} : \mathbb{N}^+ \times \mathbb{N}^+ \leftrightarrow \mathbb{N}^+$ to allocate storage for tables, then to expand a table from dimensions $n \times n$ to $(n+1) \times (n+1)$, we need move only $O(n)$ items to accommodate the *new* table entries; the current entries need not be moved. For square tables, the following *Square-shell* pairing function $\mathscr{S}$ manages the described scenario perfectly. After describing $\mathscr{S}$, we comment on managing tables of other shapes.

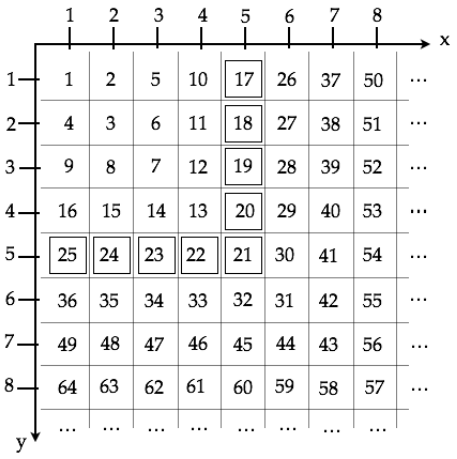


Fig. 7.5 *The square-shell pairing function $\mathscr{S}$. The shell* $\max(x,y) = 5$ *is highlighted.*

$$\begin{aligned} \mathscr{S}(x,y) &= m^2 + m + y - x + 1 \\ &\text{where} \quad m \stackrel{\text{def}}{=} \max(x-1, y-1). \end{aligned} \tag{7.15}$$

One sees in Fig. 7.5 that $\mathscr{S}$ follows the prescription of Procedure PF-Constructor: (1) it maps integers into the "square shells" defined by: $m = 0,\ m = 1, \ldots$. (2) it enumerates the entries in each shell in a counterclockwise direction. (Of course, $\mathscr{S}$ has a twin that enumerates the shells in a clockwise direction.)

≈≈≈≈≈≈≈≈≈≈

Using somewhat more complicated instantiations of Procedure PF-Constructor, the study in [64] adapts the square-shell pairing function $\mathscr{S}$ to: (a) accommodate, with no wastage, arrays/tables of any fixed aspect ratio $an \times bn$ ($a, b \in \mathbb{N}$); (b) accommodate, with only $O(n)$ wastage, arrays/tables whose aspect ratios come from a fixed finite set of candidates—i.e., $(a_1 n \times b_1 n)$ or $(a_2 n \times b_2 n)$ or ... or $(a_k n \times b_k n)$.

$\approx\approx\approx\approx\approx\approx\approx\approx\approx\approx$

7.3.2.4 The Hyperbolic-shell pairing function $\mathscr{H}$

We have just seen, in subsections B and C, that when the growth patterns of one's arrays/tables is very constrained, one can use pairing functions as storage mappings with very little wastage. In contrast, if one employs a pairing function such as $\mathscr{D}$ without consideration of its wastage, then a storage map would show some $O(n)$-entry tables being "spread" over $\Omega(n^2)$ storage locations. In the worst-case, $\mathscr{D}$ spreads the n-position $1 \times n$ array/table over $> \frac{1}{2}n^2$ addresses: $\mathscr{D}(1,1) = 1$ and $\mathscr{D}(1,n) = \frac{1}{2}(n^2+n)$. This degree of wastefulness can be avoided via careful analysis, coupled with the use of Procedure PF-Constructor. The target commodity to be minimized is the *spread* of a PF-based storage map, which we define as follows.

Note that an ordered pair of integers $\langle x, y\rangle$ appears as a position-index within an n-position table if, and only if, $xy \le n$. Therefore, we define the spread of a PF-based storage map $\mathscr{M}$ via the function

$$\mathbf{S}_{\mathscr{M}}(n) \stackrel{\text{def}}{=} \max\{\mathscr{M}(x,y) \mid xy \le n\}. \tag{7.16}$$

$\mathbf{S}_{\mathscr{M}}(n)$ is the largest "address" that PF $\mathscr{M}$ assigns to any position of a table that has n or fewer positions.

Happily, the tools we have developed enable us to design a pairing function that (to within constant factors) has minimum worst-case spread. This is the *Hyperbolic-shell pairing function* $\mathscr{H}$ of (7.17) and Fig. 7.6.[4]

$$\begin{aligned} &\text{Let } \delta(k) \text{ be the number of divisors of the integer } k. \\ &\mathscr{H}(x,y) = \sum_{k=1}^{xy-1} \delta(k) \; + \text{ the position of } \langle x,y\rangle \text{ among 2-part} \\ &\qquad\qquad\qquad\qquad \text{factorizations of the number } xy, \text{ in} \\ &\qquad\qquad\qquad\qquad \text{reverse lexicographic order} \end{aligned} \tag{7.17}$$

Proposition 7.8 ([64]) **(a)** *The hyperbolic function $\mathscr{H}$ is a pairing function.*
(b) *The spread of $\mathscr{H}$ is given by $\mathbf{S}_{\mathscr{H}}(n) = O(n \log n)$.*[5]

---

[4] Details appear in [63, 64].

[5] A detailed analysis reveals that the spread of $\mathscr{H}$ is closely related to the *natural* logarithm, whose base is Euler's constant e.

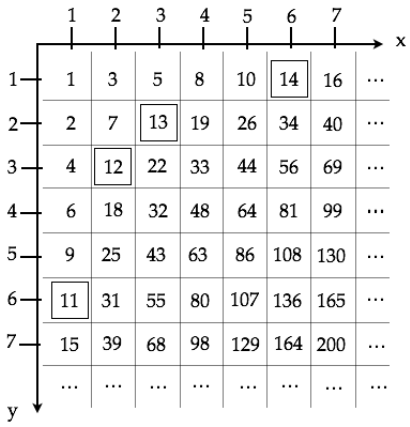


Fig. 7.6 *The hyperbolic pairing function $\mathscr{H}$. The shell $xy = 6$ is highlighted.*

(c) *No pairing function has better compactness than $\mathscr{H}$ (in the worst case) by more than a constant factor.*

Proof. **(a)** The fact that $\mathscr{H}$ is a pairing function follows from Proposition 7.7.
(b) The pairing function $\mathscr{H}$ maps integers along the "hyperbolic shells" defined by $xy = 1$, $xy = 2$, $xy = 3, \ldots$. Hence, when an integer n is "placed" into the table of values of $\mathscr{H}$, the number of occupied slots is within n of

$$\sum_{i=1}^{n-1} |\{\langle x, y\rangle \mid xy < i\}|$$

Elementary calculations show that this sum is $O(n \log n)$.
(c) The optimality of $\mathscr{H}$ in compactness (up to constant factors) is seen via the following argument. The set of tables that have n or fewer positions are those of aspect ratios $a_i \times b_i$, where $a_i b_i \leq n$. As one sees from Fig. 7.7 (generalized to arbitrary n), the union of the positions of all these arrays is the set of integer lattice points under the hyperbola $xy = n$. It is well-known—cf. [60]—that this set of points has cardinality $\Theta(n \log n)$.[6] Since every table contains position $\langle 1, 1\rangle$, it follows that, for every n, some table containing n or fewer positions is spread over $\Omega(n \log n)$ "addresses." □

[6] Recall from Chapter 3.1.1 that the *cardinality* of a finite set S is the number of elements in S.

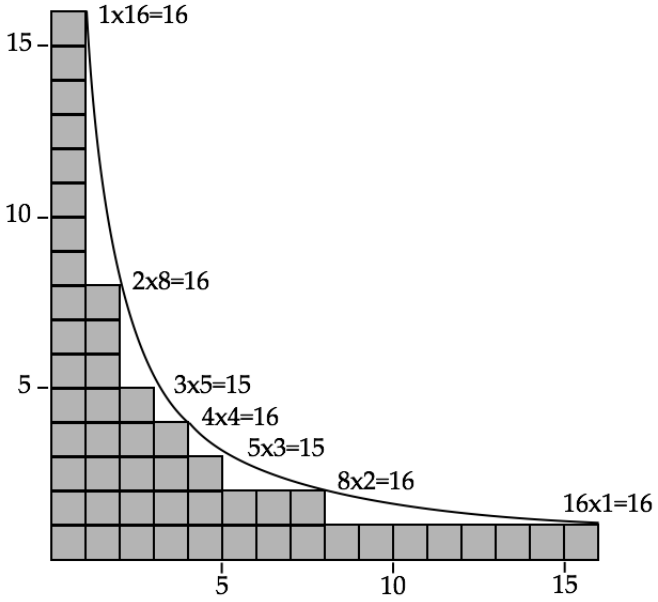


Fig. 7.7 The aggregate set of positions of tables having 16 or fewer position. To help the reader understand the figure, we include the curve $f(x,y) = xy$ which provides an upper envelope for the set. A careful look at this curve will reveal that it touches the set of positions at the points $\langle x,y\rangle \in \{\langle 1,16\rangle, \langle 2,8\rangle, \langle 4,4\rangle, \langle 8,2\rangle, \langle 16,1\rangle\}$, but it does *not* touch the set at the points $\langle x,y\rangle \in \{\langle 3,5\rangle, \langle 5,3\rangle\}$.

7.3.3 There Are No More *Ordered Pairs than Integers*

7.3.3.1 Comparing infinite sets via cardinalities

We have remarked earlier (and will do so again) that one must be very careful when reasoning about infinite sets. They can behave in ways that seem quite contradictory to our experience with finite sets. One of the most dramatic instances of this is encountered when we ask whether the set $\mathbb{N}^+ \times \mathbb{N}^+$ is "larger" than the set $\mathbb{N}^+$. We need to put the word "larger" in quotes because we do not know (yet) what the word means in the setting of infinite sets. Supplying a meaning for the word that is at once mathematically tractable and intuitively plausible was among the seminal contributions of the 19th-century German mathematician and logician Georg Cantor.

Cantor began by seeking an intuitively plausible and mathematically tractable formal notion that would allow us to verify or refute the assertion that one infinite set is "larger" than another. He addressed this question, and related ones, in his groundbreaking study of the relative "sizes" of infinite sets [19, 20]. We adapt enough of his formulation to suggest how such issues can be dealt with mathematically.

We take our lead from finite sets. Is there a notion of "bigger" for finite sets that can be extended to infinite sets?

We begin with a set A of apples and a set O of oranges, together with the challenge of determining which set is bigger.

If sets A and O are both finite, then we can just *count* the number of apples in A, call it a, and the number of oranges in O, call it o, and then compare the sizes of the (nonnegative) integers a and o. The Trichotomy Laws for integers (Section 4.3.A) guarantee that we shall be able to settle the question.
But we cannot count the elements in an infinite set, so this approach fails us when we have access to infinitely much fruit! So we need another approach.

Here is an approach that works for finite sets and that promises to extend to infinite sets. Let us assume that we can "prove"—we shall explain the word imminently—the following.

For every apple that we extract from set A *for the first time*, we can extract an orange from set O *for the first time*. It will then follow (at least in the finite case), that

There are at least as many oranges as apples!

This is really promising, because there is another way to describe the fruit-matching process that readily extends to infinite sets.

There is an injection,[7] *call it f, from O to A.*

In more formal terms: *Every time you pull an apple α from set A, I pull the orange $f^{-1}(\alpha)$ from O.*

Inspired by this formulation using injections—and by the work of Cantor—we craft the following definition.

Given sets A and O (finite or infinite), we say

Set O is at least as big as set A, denoted $|O| \geq |A|$

precisely when there is an injection from O to A.

Finally, we say that

Sets O and A have the same cardinality, denoted $|O| = |A|$

precisely when there is an injection from O to A and *an injection from A to O.*

Finally, back to numbers!

There has always been special interest in comparing the cardinalities of specific infinite sets with the cardinality of the integers. This interest has led to the following pair of adjectives.

- A (finite or infinite) set S is *countable* if $|S| \leq |\mathbb{N}|$.
- An infinite set S is *uncountable* if $|S| \not\leq |\mathbb{N}|$.

7.3.3.2 Comparing $\mathbb{N}$ and $\mathbb{N} \times \mathbb{N}$ via cardinalities

An obvious first candidate whose cardinality to compare with that of the integers $\mathbb{N}$ (or $\mathbb{Z}$, or $\mathbb{N}^+$) is the corresponding set of ordered pairs, $\mathbb{N} \times \mathbb{N}$ (or $\mathbb{Z} \times \mathbb{Z}$, or

[7] Recall from Chapter 3.2.4 that "*injection*" is synonymous with "*one-to-one function*".

$\mathbb{N}^+ \times \mathbb{N}^+$). Cantor discovered in the 1870s that pairing does not increase cardinality in infinite sets. We now prove this for the set $\mathbb{N}$, but we could easily repeat our argument for $\mathbb{Z}$ or $\mathbb{N}^+$.

Proposition 7.9 *The set $\mathbb{N} \times \mathbb{N}$ is countable: $|\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$.*

Proof. We prove the following propositions in turn.

(a) *There exists an injection from $\mathbb{N}$ to $\mathbb{N} \times \mathbb{N}$. Therefore, $|\mathbb{N}| \leq |\mathbb{N} \times \mathbb{N}|$; informally, $\mathbb{N} \times \mathbb{N}$ is at least as big as $\mathbb{N}$.*

Subproposition (a) follows easily from the existence of the following injection from $\mathbb{N}$ into $\mathbb{N} \times \mathbb{N}$.

$$(\forall n \in \mathbb{N})\ [f(n) = \langle n, n \rangle].$$

(b) *There exists an injection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$. Therefore, $|\mathbb{N} \times \mathbb{N}| \leq |\mathbb{N}|$; informally, $\mathbb{N}$ is at least as big as $\mathbb{N} \times \mathbb{N}$.*

We establish subproposition (b) by defining an injection from $\mathbb{N} \times \mathbb{N}$ into $\mathbb{N}$. We employ a function that is inspired by the Fundamental Theorem of Arithmetic (Theorem 7.1). Specifically, the Theorem assures us that the function

$$f_2(p, q) \stackrel{\text{def}}{=} 2^p 3^q$$

is an *injection* from $\mathbb{N} \times \mathbb{N}$ into $\mathbb{N}$.

Subproposition (a) and (b) prove the result. □

The *pairing functions* of Section 7.3 provide more interesting alternatives to the preceding proof of Proposition 7.9. Being *bijections* between $\mathbb{N}$ and $\mathbb{N} \times \mathbb{N}$, pairing functions can be adapted to prove any such proposition *in a single step*.

We now present a remarkable theorem that demonstrates that such single-step proofs of equality of cardinality are *always* available! There is *always* a bijection whenever there exist paired injections!

7.3.3.3 The Schröder-Bernstein Theorem

Although sets and their cardinalities are not the major focus of this chapter, it is worth a short digression to expand on the last remark in Section 7.3.3.2. It is not a coincidence that there exists both

a bijection between $\mathbb{N}$ and $\mathbb{N} \times \mathbb{N}$

and

an injection from $\mathbb{N}$ to $\mathbb{N} \times \mathbb{N}$ and an injection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$.

The celebrated theorem of Schröder and Bernstein, which is, alternatively, attributed to (Georg) Cantor and Bernstein, tells us that bijections and paired injections always travel together.

Theorem 7.3 (The Schröder-Bernstein Theorem). *Let S and T be (finite or infinite) sets such that there exists an injection $f : S \to T$ and an injection $g : T \to S$. Then there exists a bijection $h : S \leftrightarrow T$.*

The theorem has a rather complicated history. Picking just a few high points associated with the theorem's namesakes: The theorem was first stated without proof by Cantor in [21]. Roughly a decade later, Schröder provided a flawed proof in [79]. As reported in [32], Schröder soon thereafter provided a correct proof, as, independently, did Bernstein.

****

7.4 Finite Number Systems

The sets that underlie the number systems that we use in most daily tasks—namely $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$, and $\mathbb{C}$—are infinite: we can always find a number in each set that is bigger than all the numbers we have seen thus far. Indeed, the last two of these sets are "two-way" infinite: we can also always find a number in each set that is smaller than all the numbers we have seen thus far.[8] There do, however, exist several very important situations in which we use number systems that are *finite* and *cyclically repetitive*. We mention only two.

- The *clocks* that we use to indicate daily time are calibrated into a fixed finite number of major subdivisions, *hours*. We endow our days with 24 hours and depending on circumstances, have our clocks measure each day's time via repeating cycles of either 12 or 24 hours. Once a clock's limit of (12 or 24 hours) has been reached, it begins its numeration all over—with no memory of the past.
- We typically orient all manner of location specification relative to a fixed reference point in terms of *angles*. There are two coexisting, competing systems for such measurement. One system subdivides the "circle" around the reference point into 360 *degrees;* the other subdivides the "circle" into 2π *radians*. For our purposes, the main interesting point is that both of these systems are *cyclically repetitive*. Once we have circled the reference point by 360 degrees (or, equivalently, by 2π radians), then we measure further circumnavigation starting over at 0 degrees/radians.

This section is dedicated to integer-based *finite* number systems that were invented to describe and measure cyclically repetitive situations such as the two just described.

[8] The philosophically inclined reader might be interested in the essay "The Two Infinities" within the *Pensées* of the French mathematician-philosopher (or philosopher-mathematician?) Blaise Pascal , whose work we shall revisit in Chapter 5.1.1.2.C.

7.4.1 Congruences on Nonnegative Integers

For any positive integer $q \in \mathbb{N}^+$, we denote by $\mathbb{N}_q$ the q-element "prefix" of the set $\mathbb{N}$ of nonnegative integers:

$$\mathbb{N}_q \stackrel{\text{def}}{=} \{0, 1, \ldots, q-1\}.$$

For nonnegative integers $m, n \in \mathbb{N}$ and positive integer $a \in \mathbb{N}^+$, we say that *m is congruent to n modulo q*, denoted

$$m \equiv n \bmod q,$$

precisely when q divides $|m-n|$. We call q the *modulus* of the congruence (relation).

Proposition 7.10 *The relation of congruence modulo a positive integer is an equivalence relation on the set $\mathbb{N}$ of nonnegative integers.*

Proof. We verify in turn the three defining properties of an equivalence relation (see Chapter 3.2.3). Focus on nonnegative integers m, n, and r and an arbitrary positive integer modulus q.

1. Congruence modulo $q \in \mathbb{N}^+$ is a *symmetric* relation on $\mathbb{N}$.
   Verification: Because $|m-n| = |n-m|$, the assertions $[q$ divides $|n-m|]$ and $[q$ divides $|n-m|]$ must hold simultaneously, i.e., either both assertions are true or neither is.
2. Congruence modulo $q \in \mathbb{N}^+$ is a *reflexive* relation on $\mathbb{N}$.
   Verification: We always have $m \equiv m \bmod q$ because every positive integer divides $m-m=0$.
3. Congruence modulo $q \in \mathbb{N}^+$ is a *transitive* relation on $\mathbb{N}$.
   Verification: Say that $m \equiv n \bmod q$ and $n \equiv r \bmod q$. The arithmetic needed to verify that these two congruences imply that $m \equiv r \bmod q$ breaks down into cases defined by the relative sizes of m, n, and r. We supply the details for the case $m > n > r$, and we leave the other cases as exercises.
   Note first that the two assumed conguences can be rewritten as assertions of divisibility: q divides $|m-n|$, and q divides $|n-r|$. Therefore, in the chosen case $m > n >$, the congruences imply that there exist integers c_1 and c_2 such that:

   a. $c_1 q = m - n$, which implies that $n = m - c_1 q$;
   b. $c_2 q = n - q$, which implies that $n = q + c_2 q$.

   We therefore have $r - m = (c2 - c_1)q$, which means that $m \equiv r \bmod q$. In other words, The relation *equiv* mod q is transitive.

The preceding three properties define an equivalence relation, hence, jointly verify the proposition. □

7.4.2 Finite Number Systems via Modular Arithmetic

Once we embellish the sets $\mathbb{N}_q$ with arithmetic operations—namely, the "big four" of addition, subtraction, multiplication, and division—we shall see why we are able to use the resulting congruential systems in the same way as their infinite counterparts, $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$. In the coming subsections, we show that every set $\mathbb{N}_q$ can "mimic" $\mathbb{N}$ and $\mathbb{Z}$ with respect to addition, subtraction, and multipliciation, but only when q is a prime number can $\mathbb{N}_q$ "mimic" $\mathbb{Q}$ with respect to division.

7.4.2.1 Sums, differences, and products exist within $\mathbb{N}_q$

Our main result in this section demonstrates that every set $\mathbb{N}_q$, when embellished with the operations addition, subtraction, and multiplication, is *closed* under these operations, in the sense spelled out in the following result.

Proposition 7.11 *For every integer $q \in \mathbb{N}^+$ and all $m, n \in \mathbb{N}_q$, the sum $m+n \bmod q$ and the difference $m-n \bmod q$ and the product $m \cdot n \bmod q$ exist within $\mathbb{N}_q$.*

Proof. For the operations of addition and multiplication, the result is true by definition of congruence modulo q: since the sum $m+n$ and the product $m \cdot n$ exist within $\mathbb{N}^+$, their "reductions" modulo q exist within $\mathbb{N}_q$. For the case of subtraction, we augment the preceding sentence with the following equation. For all $r \in \mathbb{Z}$

$$q - r \equiv -r \bmod q.$$

One verifies this equation by noting the following chain of equalities and congruences (parentheses added to enhance legibility)

$$(q-r)-(-r) = (q-r)+r = q \equiv 0 \bmod q.$$

In all cases, therefore, the result of the operation remains in the set $\mathbb{N}_q$. □

We cannot generally add division to the set of operations listed in Proposition 7.11. For instance, the following table shows that the equation

$$2x \equiv 1 \bmod 6$$

is not solvable for all $x \in \mathbb{N}_6 \setminus \{0\}$.

x	$2 \cdot x \bmod 6$
1	2
2	4
3	0
4	2
5	4

The next subsection implicitly identifies the modulus 6's non-primality as the culprit in this example. In fact, the reader can easily show that $\mathbb{N}_q$ *is never closed under the operation of division when the modulus q is composite, i.e., nonprime.*

7.4.2.2 Quotients exist within $\mathbb{N}_p$ for every prime p

This section considers congruences modulo a prime number. We begin with our main result: *for every prime number p, every nonzero* $n \in \mathbb{N}_p$ *has a* multiplicative inverse, i.e., an element $m \in \mathbb{N}_p$ such that $m \cdot n \equiv 1 \bmod p$. Of course, the existence of multiplicative inverses allows one to *divide* any number in $\mathbb{N}_p$ by any nonzero number.

Proposition 7.12 *For every prime number p, every nonzero number* $n \in \mathbb{N}_p$ *has a multiplicative inverse within* $\mathbb{N}_p$.

Proof. Our proof combines applications of the Fundamental Theorem of Arithmetic (Theorem 7.1) and the Pigeonhole Principle (Section 2.2.6), alongside a proof by contradiction. It thereby exercises many of our important new proof techniques.

Focus on the set $\mathbb{N}_p$ for some prime p. Let n be any nonzero number in $\mathbb{N}_p$.

Lemma 7.1. *There do* not *exist nonzero numbers* m_1 *and* $m_2 \neq m_1$ *in* $\mathbb{N}_p$ *such that* $m_1 \cdot n \equiv m_2 \cdot n \bmod p$.

Proof. (*of Lemma 7.1*) Assume for contradiction that there *do* exist m_1 and $m_2 \neq m_1$ in $\mathbb{N}_p$ such that $m_1 \cdot n \equiv m_2 \cdot n \bmod p$. Say, with no loss of generality, that $m_1 > m_2$ within the set $\mathbb{N}$. We must then have

$$p \text{ divides } (m_1 - m_2) \cdot n. \tag{7.18}$$

The fact that p is a prime ensures—by Proposition 7.3—that p divides at least one of the integers n or $(m_1 - m_2)$. Because both of these integers belong to $\mathbb{N}_p$, hence lie strictly between 0 and $p-1$ (within the infinite set $\mathbb{N}$), the divisibility posited in (7.18) is impossible! The lemma follows. □

Lemma 7.1 guarantees that all of the following $p-1$ elements of $\mathbb{N}_p$ are nonzero and distinct:

$$1 \cdot n, \ 2 \cdot n, \ldots, \ (p-1) \cdot n.$$

Because $\mathbb{N}_p$ has precisely $p-1$ nonzero elements, these $p-1$ multiples of n must exhaust these elements. In other words, some multiple of n, say $c \cdot n$, must equal 1. This means that the number $c \in \mathbb{N}_p$ is n's multiplicative inverse within $\mathbb{N}_p$. □

Of course, once we have multiplicative inverses, we have the operation of division and, consequently, arbitrary quotients and fractions. Of course, fractions within finite number systems such as $\mathbb{N}_p$ are going to look strange to our eyes, as the following example indicates.

What does the number $7/4$ look like within $\mathbb{N}_5$? We develop the answer in steps.

$$\approx\approx\approx\approx\approx\approx\approx\approx\approx$$

We are able to proceed in the following manner because the relations $(\equiv \bmod q)$ are *congruences*, i.e., equivalence relations whose class structures are consistent with the algebraic structure of the arithmetic systems exemplified by $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{N}_p$ under the classical four arithmetic operations. A full treatment of this topic is beyond the scope of this text.

$$\approx\approx\approx\approx\approx\approx\approx\approx\approx$$

1. The numbers $4, 7 \in \mathbb{N}$ correspond, respectively, to the numbers $4, 2 \in \mathbb{N}_5$.
   Verification: $4 \equiv 4 \bmod 5$, and $7 \equiv 2 \bmod 5$.
2. The multiplicative inverse of 4 within $\mathbb{N}_5$ is 4.
   Verification: $4 \cdot 4 \ = \ 16 \ \equiv 1 \bmod 5$.
3. THEREFORE, we have the following "translation" of the quotient $7/4$:
   Within $\mathbb{N}$:
   the product of $7 \in \mathbb{N}$ by the multiplicative inverse of $4 \in \mathbb{N}$
   Within $\mathbb{N}_5$:
   the product of $2 \in \mathbb{N}_5$ by the multiplicative inverse of $4 \in \mathbb{N}_5$, which is 4
4. the product of $2 \in \mathbb{N}_5$ by $4 \in \mathbb{N}_5$ is 3.
   Verification: $2 \cdot 4 \ = \ 8 \ \equiv 3 \bmod 5$.

We thus have the unintuitive fact that the rational number $7/4$ corresponds to the number $3 \in \mathbb{N}_6$.

Of course, we do not often perform arbitrary arithmetic within the finite number systems $\mathbb{N}_p$, so we do not often struggle with the unfamiliar results of this subsection. That said, we do sometimes intermix "ordinary" numeration with "modular" numeration, as when we coordinate talk about elapsed time (measured in the "ordinary" way) with wall-clock time (which is a "modular" system). So, in summation, it is worth the effort to understand this seldom-used material. Plus, it can be amusing to announce at a party that you can "prove" that $1.75 = 3$.

Chapter 8
RECURRENCES

One of the intellectually most powerful strategies for all manner of human endeavor is to "learn from the past"—i.e., to *re*-use knowledge that one has acquired earlier in order to acquire new knowledge. Within the domain of computing, this strategy is exemplified by computations that derive the value of a function F at an argument $n \in \mathbb{N}^+$ by invoking the (hopefully, earlier-computed) values of F at arguments $1, 2, \ldots, n-1$. The classical first example of such a *recurrent* mode of computing involves the *factorial function* FACT of Section 5.1.1.1.C. The "direct" mode of computing FACT at an argument $n \in \mathbb{N}^+$ is:

$$\text{FACT}(n) \;=\; 1 \times 2 \times \cdots \times n.$$

The *recurrent* mode of computing FACT(n) is more compact—and it better exposes the inherent structure of the function.

$$\text{FACT}(n) \;=\; \begin{cases} n \times \text{FACT}(n-1) & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

This chapter is devoted to deriving and solving a variety of types of recurrences. In common with the rest of this text, our treatment of this subject emphasizes exploiting recurrent structure in reasoning and analysis: increased understanding will enable improved computing.

8.1 Linear Recurrences

This section is devoted to the solution of *linear* recurrences, as exemplified by the following function specifications.

Should we change the notation of F here? since it will be used for Fibo elements in the following section of the chapter. We can keep it as it is, and add a note about the abstract letters usually x for any variable and F for any function...

$$F(n) = \begin{cases} aF(n/b) + f(n) & \text{for } n \geq b \\ c & \text{for } n < b \end{cases} \tag{8.1}$$

I changed the bounds in the expression to keep more generality. I also changed F(1) for a constant c replacing 1

We should say here that a, b and c are integers

$$F(n) = \begin{cases} aF(n/b) + c & \text{for } n \geq b \\ 1 & \text{for } n < b \end{cases} \tag{8.2}$$

constant c ws used twice, this was a bit confuseing... Thus, I changed here F(1) to 1 since it is the simplified form...

Myriad basic algorithmic problems, including sorting, selection, matching, etc., can be solved using linear-recurrent algorithms [28] —and such algorithms yield to specification and analysis via linear recurrences.

In the next two subsections, we present analyses of recurrences (8.1) and (8.2). The reader will be able to extend the techniques we use in our analyses of these specific recurrences in order to analyze other members of the important family of linear-recurrent algorithms.

8.1.1 The Simple Recurrence

We focus first on the simpler of our sample recurrences, namely, (8.2). Happily, there is a single perspicuous proof that elegantly solves recurrences of this form.

By the time the reader has reached this paragraph, she has the mathematical tools necessary to prove and apply what is called *The Master Theorem for Linear Recurrences* [28]. The main tools in the proof of the Theorem are: summing geometric summations (Section 6.2.2) and employing elementary asymptotic notions and notations (Section 2.4.1).

Theorem 8.1 (The simplified form of Master Theorem for Linear Recurrences). *Let the function F be specified by the linear recurrence (8.2). Then the value of F on any argument n is given by*

$$\begin{aligned} F(n) &= (1+\log_b n)c && \text{if } a = 1 \\ &= \frac{1-a^{\log_b n}}{1-a}c \approx \frac{c}{1-a} && \text{if } a < 1 \\ &= \frac{a^{\log_b n}-1}{a-1}c && \text{if } a > 1 \end{aligned} \tag{8.3}$$

Proof. We expose the pattern generated by recurrence (8.2), by beginning to "expand" the specified computation—replacing occurrences of $F(\bullet)$ as mandated in (8.2). Once we discern the pattern, we jump to the general form.

$$\begin{array}{rcl} F(n) = & aF(n/b)+c & \\ = & a\left(aF(n/b^2)+c\right)+c & = a^2F(n/b^2)+(a+1)c \\ = & a^2\left(aF(n/b^3)+c\right)+(a+1)c & = a^3F(n/b^3)+(a^2+a+1)c \\ & \vdots & \vdots \\ = & \left(a^{\log_b n}+\cdots+a^2+a+1\right)c & \end{array} \tag{8.4}$$

The segment of (8.4) "hidden" by the vertical dots betokens an induction that is left to the reader. Equations (6.18) and (6.19) now enable us to demonstrate that (8.3) is the asserted case-structured solution to (8.2). □

8.1.2 The More General Recurrence

We now progress from the simple recurrence (8.2) to the more general recurrence (8.1). We simplify our problem in two ways, in order to avoid calculational complications (such as floors and ceilings) that can mask the principles that govern our analysis.

1. We focus on the case $f(n) = n$.
   It is a triviality to generalize to the slightly more ambitious $f(n) = \alpha n + \beta$ (i.e., to a *linear* function f), but this extension teaches no new lessons.
2. We assume that the argument n to functions F and f is a power of b.
3. We consider $c = 1$.

Removing these assumptions would significantly complicate our calculations, but it would not change our reasoning.

By "unfolding" (8.1) as in Section 8.1.1, we expose the algebraic pattern created by the recurrence. As in (8.4), once we discern this pattern, we jump to the general form (which can be verified via induction).

$$\begin{array}{rcl} F(n) = & aF(n/b)+n & \\ = & a\left(aF(n/b^2)+n/b\right)+n & = a^2F(n/b^2)+(an/b+n) \\ = & a^2\left(aF(n/b^3)+n/b^2\right)+(a/b+1)n & = a^3F(n/b^3)+(a^2/b^2+a/b+1)n \\ & \vdots & \vdots \\ = & a^{\log_b n}F(1)+\left(\displaystyle\sum_{i=0}^{\log_b(n)-1}(a/b)^i\right)n & \end{array}$$

We thus see that solving the more general recurrence (8.1) requires only augmenting the solution to the simpler recurrence (8.2) by "appending" to the simple solution a geometric summation whose base is the ratio a/b. The reader can now invoke the techniques from Section 6.2.2.2 to arrive at a solution to (8.1).

When "doing mathematics," one is often interested in discovering the *dominant behavior* of the function $F(n)$ specified via a recurrence such as (8.1). An important

lesson to garner from the analysis we have been performing throughout this section is the following:

- When $a > b$, the behavior of $F(n)$ is dominated by the first solution term

$$a^{\log_b n} \cdot F(1) \;=\; n^{\log_b a}$$

as we assumed F(1)=1, I remove F(1) in the last term

- When $a < b$, the behavior of $F(n)$ is dominated by the second solution term

$$n \cdot \sum_{i=0}^{\log_b(n)-1} (a/b)^i = \frac{\left(1 \;-\; (a/b)^{\log_b(n)}\right)}{1-\frac{a}{b}} n$$
$$\approx n\frac{b}{b-a}$$

Is there a reason to avoid to write the theorem here as in the previous section?

One can learn yet more lessons about $F(n)$, specifically about how to compute $F(n)$ (exactly or approximately) by observing the graphical depiction in Fig. 8.1 of the calculation described by recurrence (8.1). In particular, one observes in the

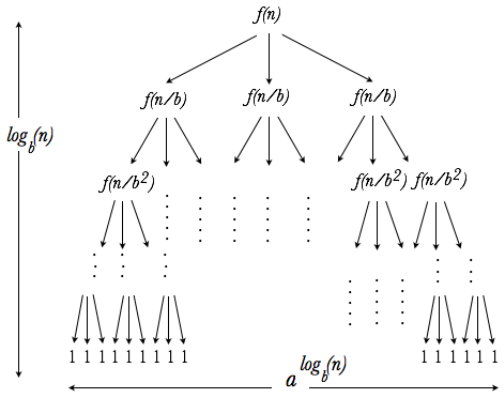


Fig. 8.1 Development of the calculation specified by recurrence (8.1). The total cost is obtained by the summation on each row: $a \times f(n/b)$ in the first row, $a^2 \times f(n/b^2)$ in the second row, and so on. This leads to $a^{log_b(n)}$.

figure that when $a = b$, the computations in each row are perfectly balanced. When $a = b = 2$, there are n leaves and the computations inside the tree are exactly n in each of the $\log_2(n)$ levels,, leading to $F(n) = n\log_2(n)$.

8.2 Bilinear Recurrences

8.2.1 Binomial Coefficients and Pascal's Triangle

In Section 5.1.1.2.C, we introduced and briefly discussed the binomial coefficient $\Delta_{n,k} \stackrel{\text{def}}{=} \binom{n}{k}$ in its guise as a binary operation on integers; see (5.3). And, we established in Proposition 5.1 the summation rule

$$\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$$

for $\Delta_{n,k}$. In fact, one can *define* binomial coefficient via the *bilinear recurrence* that underlies this rule. This change in viewpoint is the topic of the current subsection.

8.2.1.1 The formation rule for Pascal's Triangle

Let us define the bivariate integer function[1] $\hat{\Delta}(n,k)$ via the bilinear recurrence

$$\hat{\Delta}(n,k) = \begin{cases} 1 & \text{if } [n=1, k=0] \\ 1 & \text{if } [n=1, k=1] \\ \hat{\Delta}(n-1,k-1) + \hat{\Delta}(n-1,k) & \text{otherwise} \end{cases} \tag{8.5}$$

We claim that the function $\hat{\Delta}(n,k)$ thus defined is, in fact, the for binomial coefficient $\binom{n}{k}$. We establish this claim with the help of a two-dimensional array of integers known as *Pascal's Triangle*, so named in honor of the French polymath Blaise Pascal. Fig. 8.2 provides a "prefix" of this famed array, for $n,k \leq 5$. The *formation rule of the array* is that the array-entry at (row $n+1$, column $k+1$) is the sum of the array-entries at (row n, column k) and at (row n, column $k+1$).

By comparing the formation rule for Pascal's Triangle with equation (5.5), you can anticipate the following result.

Proposition 8.1 *The entries of Pascal's Triangle are the binomial coefficients. Specifically, for all n,k, the entry at (row n, column k) of the Triangle is $\binom{n}{k}$.*

Proof. We note by observation and direct calculation (see Fig. 8.2) that the proposition is true for $n = 1$ and $k \in \{0,1\}$. A simple double induction verifies that every binomial coefficient appears in the Triangle and every Triangle entry is a binomial coefficient. □

[1] We alter our notation for binomial coefficients in deference to our change in viewpoint: We promote the integer pair $\langle n,k \rangle$ from a subscript to an argument, and we embellish Δ with a hat.

$\binom{n}{k}$	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$\ldots$
$n=1$	1	1									$\ldots$
$n=2$	1	2	1								$\ldots$
$n=3$	1	3	3	1							$\ldots$
$n=4$	1	4	6	4	1						$\ldots$
$n=5$	1	5	10	10	5	1					$\ldots$
$n=6$	1	6	15	20	15	6	1				$\ldots$
$n=7$	1	7	21	35	35	21	7	1			$\ldots$
$n=8$	1	8	28	56	70	56	28	8	1		$\ldots$
$n=9$	1	9	36	84	126	126	84	36	9	1	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Fig. 8.2 A "prefix" of Pascal's Triangle, for $n,k \leq 9$.

****

induction on n, then for each value of n on $k \leq n$

SHOULD WE SPELL THIS OUT IN DETAIL? GIVE AS AN EXERCISE?

Yes, I think we can detail the classical recurrence proof. I added an alternative proof based on counting the number of paths in the Pascal's triangle – see figure

****

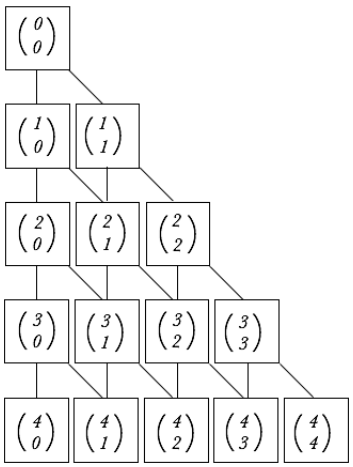


Fig. 8.3 Another representation of the Pascal's triangle

.

As an immediate consequence of the relation between binomial coefficients and Pascal's Triangle, we observe the following *a priori* nonobvious fact.

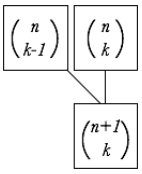


Fig. 8.4 Argument for an alternative proof where the coefficient in row $n+1$ is obtained by the two previous coefficients in row n. Then, the number of paths to reach this coefficient is equal to the sum of the number of paths in these two coefficients of the previous row.

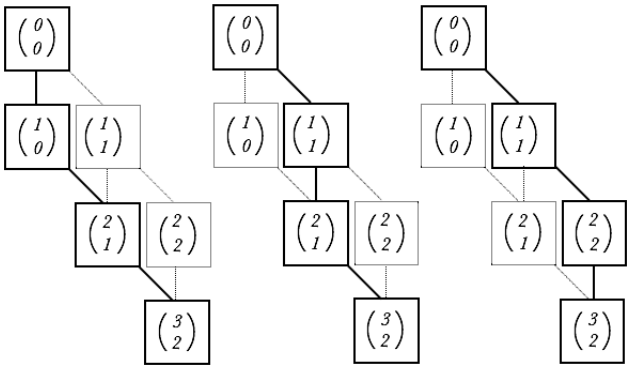


Fig. 8.5 There are three different paths for reaching $\binom{3}{2}$.

Proposition 8.2 *Every binomial coefficient is an integer.*

Proof. By the formation rule for Pascal's Triangle, every entry in that array is obtained from integers via repeated additions. The present result therefore follows from Proposition 8.1's proof that the elements of the Triangle are precisely the binomial coefficients. □

8.2.1.2 The summation formula for binomial coefficients

We conclude this section with a very consequential result about the binomial coefficients. We shall observe applications of this result as we explore a variety of topics, ranging from counting discrete structures and calculating probabilities to deriving basic properties of other recursively defined families.

Proposition 8.3 *For every positive integer n,*

$$\sum_{i=0}^{n} \binom{n}{i} = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n-1} + \binom{n}{n} = 2^n$$

Proof. This result is an immediate consequence of the Binomial Theorem (Theorem 5.3). That seminal result tells us that, for all $n \in \mathbb{N}$,

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^{n-i} y^i.$$

If we instantiate this polynomial equation with the values $x = y = 1$, then we obtain the present result. □

8.2.2 The Fibonacci Sequence

This section is devoted to one of the most storied topics in the world of mathematics—in terms of the topic's manifestation in the real world and in terms of the multiple names used to refer to its discoverer,[2] the 13th-century Italian mathematician variously known as:

Fibonacci	(Italian for: son of Bonaccio)
Leonardo of Pisa	(his hometown)
Leonardo Pisano	(variant of "of Pisa")
Leonardo Pisano Bigolo	(his hometown plus family name)
Leonardo Fibonacci	(for: son of Bonaccio Bigolo)
Leonardo Bonacci	(for: son of Bonaccio Bigolo)

The sequence discovered by this multi-named genius is defined as follows.

The *Fibonacci sequence*, or, *the Fibonacci numbers*, is an infinite sequence

$$F(0),\ F(1),\ F(2),\ \ldots$$

of elements of $\mathbb{N}^+$, the set of positive integers. As just denoted, we see that the numbers in the sequence are traditionally indexed by elements of $\mathbb{N}$, the set of non-negative integers and are often written using functional ($F(i)$) notation rather than subscripts (F_i). The classical definition of the sequence is as follows.

$$\begin{aligned} F(0) &= 1 \\ F(1) &= 1 \\ F(n) &= F(n-1) + F(n-2) \quad \text{for all } n > 1 \end{aligned} \tag{8.6}$$

The sequence is often specified just by listing its early elements:

$$1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ \ldots$$

[2] Not surprisingly, this marvelous sequence was discovered many times, in many places. Our story refers ony to its discovery in the West.

8.2.2.1 The story of the Fibonacci numbers

Leonardo Fibonnaci describes[3] discovering his eponymous sequence in the course of contemplating the rate of population growth of successive generations of an idealized immortal initial pair of rabbits. Rabbits mature quickly and, after attaining maturity at one month, can spawn a new pair of progeny every following month. So, at "time 0", there is one pair of rabbits. This persists at month 1, because there has not yet been time to produce new rabbits. By month 2, though, there are 2 pairs of rabbits. At month 3, only the first pair will have spawned, so there are 3 pairs of rabbits. At month 4, these 3 pairs are joined by 2 more. The reader can continue this story and discover that the number of pairs of rabbits observed after successive months are given by the sequence generated by the process implicit in recurrence (8.6) and illustrated by our initial list. Fig. 8.6 below illustrates the process up to month 4.

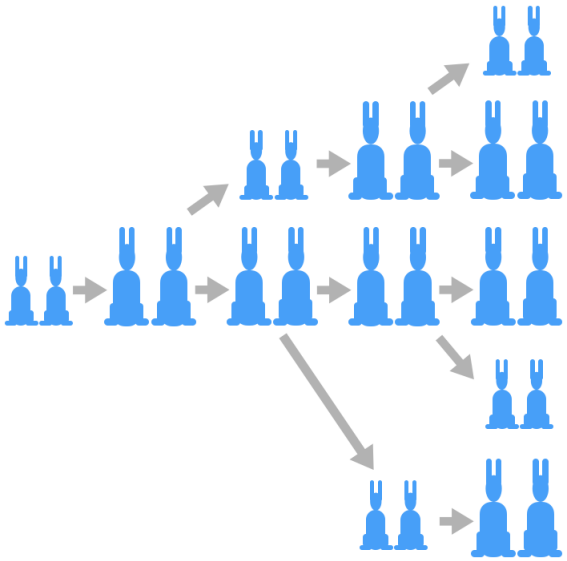


Fig. 8.6 The successive generations of rabbits are growing each month (depicted for the first four months).

The Fibonacci sequence's role in describing idealized rabbit population statistics is no more fascinating than its appearance elsewhere in the natural world—in structural features such as the patterns of seeds in flower heads, the numbers of petals of flowers, the growth patterns of pine cones and pineapples, and on and on; see [9].

The story of this fascinating sequence of numbers has a macroscopic aspect also. Many cultures, the ancient Greeks among them, have attributed mystical properties

[3] In his *Liber Abaci*

to (classes of) numbers; our discussions about the *prime numbers* in Section 7.2 and about the *perfect numbers* in Section 7.2.3 bear witness to this phenomenon. One specific number that has attracted such attention is the *golden ratio*, an irrational real number which is usually denoted Φ and which has the following (exact and approximate) values:

$$\Phi = \frac{1+\sqrt{5}}{2} \approx 1.618\ldots$$

It has been alleged that rectangles whose *aspect ratios* (Length $\div$ Width) are (roughly) Φ are the most pleasing to the human eye. In fact, the aspect ratio of the Parthenon, in Athens is (roughly) Φ, although it is not known whether this is intentional. The relevance of Φ to this section resides in the fact that *the sequence of ratios of successive Fibonacci numbers approaches* Φ.

You can perform your own test about pleasing rectangles and ratios of successive Fibonacci numbers by perusing Fig. 8.7.

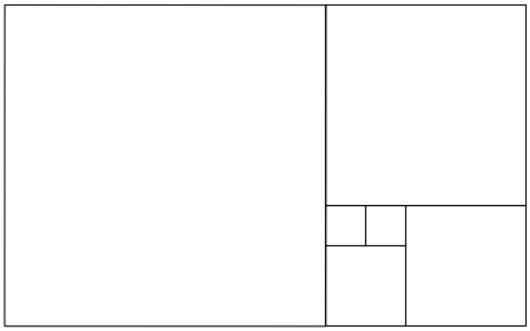


Fig. 8.7 Successive Fibonacci numbers interpreted geometrically, via a spiral of squares whose respective sides form a Fibonacci sequence.

The mathematical properties of this truly remarkable sequence will occupy our attention in the remainder of this section.

8.2.2.2 Fibonacci numbers and binomial coefficients

There is a strong, nonobvious, connection between the binomial coefficients of Section 8.2.1 and the Fibonacci numbers of the current section. We observe this connection by contemplating the diagonals of Pascal's Triangle. See Fig. 8.8.

Proposition 8.4 *For all $n \in \mathbb{N}$, the Fibonacci number $F(n)$ is the sum of the first $\lceil (n+1)/2 \rceil$ binomial coefficients $\binom{k}{i}$ such that $k+i=n$. Symbolically,*

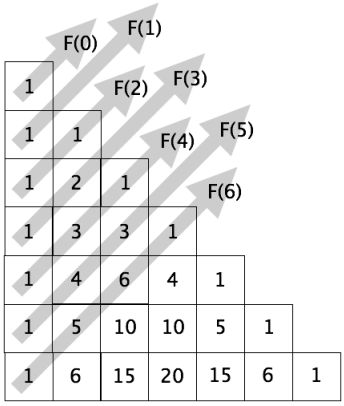


Fig. 8.8 Obtaining Fibonacci numbers as the sums of diagonal elements of the left-justified Pascal Triangle.

$$F(n) = \binom{n}{0} + \binom{n-1}{1} + \cdots + \binom{\lfloor (n+1)/2 \rfloor}{\lceil (n+1)/2 \rceil - 1}. \tag{8.7}$$

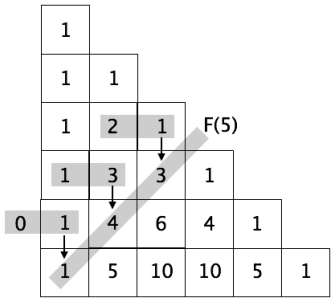


Fig. 8.9 Each term of the diagonal is obtained by summing the two preceding ones.

Proof (Sketch). Because of the heavy calculational content of a complete proof, we provide here just a short sketch.

Fig. 8.9 depicts a portion of Pascal's Triangle with shaded diagonal and horizontal annotations. The shaded diagonal annotation depicts the three numbers in the Triangle that sum to the Fibonacci number $F(5)$.

1. Looking at the three horizontal shaded areas *individually* illustrates how each of the three numbers on the shaded diagonal (1, 4, 3), being a binomial coefficient, arises as a sum of two numbers on the preceding row of the array—as instantiations of the formation rule for Pascal's Triangle. The illustrated instance of the rule asserts that:

$$\binom{5}{0} = 0 + \binom{4}{0} = 0+1 = 1$$

$$\binom{4}{1} = \binom{3}{0} + \binom{3}{1} = 1+3 = 4$$

$$\binom{3}{2} = \binom{2}{1} + \binom{2}{2} = 2+1 = 3$$

2. Looking at the three horizontal shaded areas *in tandem* illustrates that the numbers along the shaded diagonal are sums of the numbers along the two diagonals that are above the shaded one—as instantiations of the formation rule for Fibonacci numbers. The illustrated instance of the rule asserts that

$$F(5) = F(4)+F(3) = 5+3 = 8.$$

The preceding reasoning provides the infrastructure of an induction that will prove that the proposition holds for every Fibonacci number. As suggested by the statement of the proposition, the required calculations on indices can obscure the rather elegant basis for the result. □

8.2.2.3 Alternative generating recurrences for the Fibonacci sequence

Although the classical recurrence (8.6) is the structurally simplest generator of the Fibonacci sequence, there exist other generators that are not much more complex. We now present two other multi-linear recurrences that generate the sequence, in addition to a family of binary generating recurrences.

A. Two multi-linear generating recurrences

Proposition 8.5 *For all integers $n \geq 2$,*

$$\begin{aligned} F(n) &= 1 + F(0) + F(1) + F(2) + \cdots + F(n-2) \\ &= 1 + \sum_{k=0}^{n-2} F(k) \end{aligned} \tag{8.8}$$

Proof. We proceed by induction.
The *base case*, $n = 2$, holds because $F(2) = 2 = 1+F(0)$.
Assume, *for induction*, that (8.8) holds for all arguments $2 \leq n < m$.
I think the indices n and m are confused here. I suggest to invert them in the following proof...
We *extend* the induction as follows. Our inductive hypothesis assures us that for all $m \geq 3$,

$$F(m-1) \;=\; 1 \;+\; F(0) \;+\; F(1) \;+\; F(2) \;+\cdots+\; F(m-3).$$

Combining this with the classical recurrence (8.6), we therefore have

$$\begin{aligned} F(m) &= F(m-2) \;+\; F(m-1) \\ &= F(m-2) \;+\; 1 \;+\; F(0) \;+\; F(1) \;+\; F(2) \;+\cdots+\; F(m-3) \end{aligned}$$

This extends the induction and completes the proof. □

While recurrence (8.8) in Proposition 8.8 employs all of the Fibonacci numbers up to the desired bound, recurrence (8.9) in the next proposition employs only every other such number.

Proposition 8.6 *For all integers $n \geq 2$,*

$$F(n) = F(n-1) \;+\; F(n-3) \;+\; F(n-5) \;+\cdots+\; C(n) \tag{8.9}$$

where

$$C(n) = \begin{cases} F(0) = 1 \text{ if } n \text{ is even} \\ F(1) = 1 \text{ if } n \text{ is odd} \end{cases}$$

Attention here on the values of C, verify! I think the constant are the same, see details in the next paragraph... We thereby sum every other term up to the $(n-1)$th and use a "clean-up" term $C(n)$ to complete the sum.

Proof. We develop the claimed summation (8.9) by iteratively expanding the right-hand term ($F(n-2)$) of the classical recurrence (8.6). This expansion begins

$$\begin{aligned} F(n) &= F(n-1) \;+\; F(n-2) \\ &= F(n-1) \;+\; F(n-3) \;+\; F(n-4) \\ &= F(n-1) \;+\; F(n-3) \;+\; F(n-5) \;+\; F(n-6) \end{aligned} \tag{8.10}$$

We continue this expansion process as long as we can, and then we add a single *clean-up term*, which we have designated $C(n)$ in the statement of the proposition.

Determining the value of $C(n)$ is facilitated by noticing that our expansion is "trying" to have all term-indices in the expanded summation have the same parity. To wit, the successive indices of the expanded summation differ by 2, beginning with $n-1$, $n-3$, $n-5$, and so on; therefore, the index $n-k$ that we expand always has the same parity as n. Let us observe the consequence of this fact for the "end game" of the expansion process.

- When n is even, our expansion eventually comes down to

$$\begin{aligned} &\cdots + F(5) \;+\; F(4) \\ \longrightarrow\; &\cdots + F(5) \;+\; F(3) \;+\; F(2) \\ \longrightarrow\; &\cdots + F(5) \;+\; F(3) \;+\; F(1) \;+ F(0) \end{aligned}$$

  The expansion must end at this point because we have run out of Fibonacci numbers! We therefore have

$$C(n) \;=\; F(0) \;=\; 1$$

- When n is odd, our expansion eventually comes down to

$$\begin{array}{r} \cdots + F(4) \;+\; F(3) \\ \longrightarrow \cdots + F(4) \;+\; F(2) \;+\; F(1) \end{array}$$

  The expansion must end at this point because we have run out of Fibonacci numbers! We therefore have

$$C(n) \;=\; F(1) \;=\; 1$$

In either case, our expanded summation produces the correct value of $F(n)$. □

I found a little mistake in the original writing, redoing the calculations, t appears the constants are the same in both cases... Please, check my calculus

B. A family of binary generating recurrences

What we have earlier called "generating recurrences" or "formation rules" for binomial coefficients and Fibonacci numbers can also be viewed as (mathematical) identities on the quantities of interest. In our usage, the line between "generating recurrences" and "identities" centers on computational issues: multilinear recurrences can feasibly be used to generate the desired numbers; nonlinear recurrences such as we expose in this subsection will likely not be used as generators. Indeed, for several of the results we cover here, it is the methodology of proof and analysis that we wish to stress.

Proposition 8.7 *For all $n \in \mathbb{N}$ and $0 < k < n$*

$$F(n) \;=\; F(k) \cdot F(n-k) \;+\; F(k-1) \cdot F(n-k-1). \tag{8.11}$$

Of course, the classical recurrence (8.6) is instance $(k = 1)$ of the family of recurrent equations (8.11).

Proof. We first explain how one might guess at the existence of the family of recurrences (8.11), and then we validate the recurrences in the family.

We begin with the classical recurrence (8.6) and iteratively use this recurrence to "expand" the classical recurrence. In detail, we begin by combining the first two instances of (8.6), namely,

$$\begin{array}{ll} F(n) & = F(n-1) + F(n-2) \\ F(n-1) & = F(n-2) + F(n-3) \end{array}$$

and we combine them algebraically to produce the following.

$$F(n) \;=\; 2F(n-2) \;+\; F(n-3).$$

And then we iterate! The following table illustrates the result of the first four iterations of the process.

$$\begin{aligned}
F(n) &= F(n) + F(n-1) \\
&= 2F(n-1) + F(n-2) \\
&= 3F(n-2) + 2F(n-3) \\
&= 5F(n-3) + 3F(n-4) \\
&= 8F(n-4) + 5F(n-5) \\
&\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots
\end{aligned}$$

Note that the coefficients of the successive occurrences of the Fibonacci numbers $F(i)$ that occur in our table are themselves Fibonacci numbers. By analyzing the emerging pattern—*remember our advice in Chapter 2 to always look for patterns*—we arrive at the family (8.11) of recurrent equations.

> Keep in mind that, at this point, we are still in the realm of conjecture! We must now verify the universal validity of the family.

We proceed by induction on the number k of iterated expansions of the classical recurrence (8.6).

The *basis for our induction* resides in the observation we shared right after stating the proposition: Instance $(k = 1)$ of the posited family of recurrent equations is just the classical recurrence (8.6).

Let us assume that instance k of family (8.11), namely, the equation

$$F(n) \;=\; F(k)\cdot F(n-k) \;+\; F(k-1)\cdot F(n-k-1)$$

is valid, take care here since k should be strictly less than $n-1$ and let us observe the result of producing instance $k+1$ from this instance. We algebraically combine the just-cited equation with the following instantiation of the classical recurrence:

$$F(n-k) \;=\; F(n-k-1) \;+\; F(n-k-2)$$

We find that

$$\begin{aligned}
F(n) &= F(k)\cdot F(n-k) \;+\; F(k-1)\cdot F(n-k-1) \\
&= F(k)\cdot \big[F(n-k-1) \;+\; F(n-k-2)\big] \;+\; F(k-1)\cdot F(n-k-1) \\
&= \big[F(k) \;+\; F(k-1)\big]\cdot F(n-k-1) \;+\; F(k)\cdot F(n-k-2) \\
&= F(k+1)\cdot F(n-k-1) \;+\; F(k)\cdot F(n-k-2).
\end{aligned}$$

The induction is thus extended, which establishes the proposition. □

8.2.2.4 ⊕ A closed-form expression for the nth Fibonacci number

We close our survey of the Fibonacci numbers by exposing a *closed-form expression*[4] for the numbers in this fascinating family. The detailed derivation of this form

[4] The term "closed-form expression" is defined and illustrated in Section 6.2.1.2.A.

is beyond the scope of this text, so we settle for a heuristic explanation of the closed-form expression. By "heuristic", we mean here the kind of intuitive explanation that mathematicians often use to garner intuition during the exploratory phase of studying a complex topic.

If you write out a sufficiently long initial sequence of Fibonacci numbers, then you observe that they grow quite fast. Indeed, by this point in the text, you have hopefully "played" with enough sequences that you might guess that the Fibonacci numbers grow exponentially with the index n. That is, you might guess that there exists a base $\beta > 1$ and a constant of proportionality $c > 0$ such that $F(n) = c\beta^n$, at least approximately. In order to (hopefully!) garner intuition for the actual growth behavior of the Fibonacci numbers, let us observe an important corollary of this guess. If the guess were true, then it would combine with the classical recurrence (8.6) in the following way.

$$\begin{array}{ll} (1)\ F(n) = c\beta^n & \text{by our guess} \\ (2)\ F(n) = F(n-1) + F(n-2) & \text{by recurrence (8.6)} \end{array}$$

By combining (1) and (2), we therefore find that

$$\beta^n = \beta^{n-1} + \beta^{n-2}$$

so that β^n is a root of the quadratic equation

$$x^2 - x - 1 = 0$$

By the quadratic formula (see Proposition 5.4), this polynomial has two roots well done, I like the way this is presented, but where is the section on quadratic formula?

$$\Phi = \frac{1+\sqrt{5}}{2} \quad \text{and} \quad \Phi' = \frac{1-\sqrt{5}}{2}$$

Note that Φ, which is known as the *golden ratio*, exceeds 1 while Φ' does not. Since we know that the Fibonacci numbers *grow* with n rather than shrink with n, our initial guess would assign $F(n)$ the value

$$F(n) = \Phi^n = \left(\frac{1+\sqrt{5}}{2}\right)^n$$

In fact, this guessed value of $F(n)$ is off by only a small constant factor, at least for very large values of n, in the sense of part (a) of the following result. Part (b) of the result actually provides a closed-form expression for $F(n)$.

Proposition 8.8 (a) (An approximating expression) *For all sufficiently large n,*

$$F(n) \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$$

The meaning of the symbol "≈" here means that the error incurred by approximating $F(n)$ via this expression shrinks exponentially as n grows.

(b) (An exact expression) *For all n,*

$$F(n) = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

8.2.3 ⊕ Relatives of Fibonacci Numbers and Binomial Coefficients

8.2.3.1 Lucas numbers

A constant preoccupation of mathematicians is to understand why important mathematical structures exhibit their observed properties. A common way to seek such understanding is to perturb the definition of the important structure and study the effects of the perturbation. While this stratagem leads to interesting, valuable results only sometimes, it is an invaluable tool in the hands of a gifted mathematician. This chapter is devoted to a brief survey of such a study by the 19th-century French mathematician François Edouard Anatole Lucas (commonly known as Edouard Lucas).

A. Definition

Lucas, who is credited with giving the name "Fibonacci numbers" to the sequence discovered by Leonardo Pisano, investigated the consequences of perturbing the initial conditions, $F(0) = F(1) = 1$, in the definition (8.6) of the Fibonacci sequence.

we may add an observation here. There is another "natural" choice of perturbation with 1 and 2, but this leads to the same sequence as before (shifted)

Lucas's overall goal was simply to replace the Fibonacci sequence's initial values $\langle 1,1\rangle$, with the values $\langle 2,1\rangle$. It turns out to be much more fruitful—in terms of more striking results and simpler proofs—to make a somewhat more drastic perturbation:

The *Lucas sequence* is the infinite sequence of positive integers

$$L(-1),\ L(0),\ L(1),\ L(2),\ldots$$

generated by the recurrence

$$\begin{aligned} L(-1) &= 2 \\ L(0) &= 1 \\ L(n) &= L(n-1) + L(n-2) \quad \text{for all } n \geq 1 \end{aligned} \tag{8.12}$$

Because we conventionally index sequences by *nonnegative* numbers, we henceforth ignore $L(-1)$ and use the following *standard definition* of the Lucas sequence.

$$\begin{aligned} L(0) &= 1 \\ L(1) &= 3 \\ L(n) &= L(n-1) + L(n-2) \quad \text{for all } n > 1 \end{aligned} \tag{8.13}$$

The following finite sequences present the first few elements of both the Lucas sequence (for illustration) and the Fibonacci sequence (for comparison).

n:	0,	1,	2,	3,	4,	5,	6,	7,	8,	9, …
$L(n)$:	1,	3,	4,	7,	11,	18,	29,	47,	76,	123, …
$F(n)$:	1,	1,	2,	3,	5,	8,	13,	21,	34,	55, …

We begin our brief study of the Lucas sequence by noting that just a minor tweak converts the Fibonacci-related identity revealed in Proposition 8.5 to an identity about Lucas numbers.

Proposition 8.9 *For all integers $n \geq 0$,*

$$L(n+2) \;=\; 1 \;+\; L(-1) \;+\; L(0) \;+\; L(1) \;+\; L(2) \;+\cdots+\; L(n) \tag{8.14}$$

Proof (Sketch). We can literally repeat the proof of Proposition 8.5, with only a change in the induction's base case, which becomes

$$L(2) \;=\; L(-1)+L(0)+1 \;=\; 2+1+1 \;=\; 4.$$

The body of the inductive argument holds for the Lucas sequence as well as for the Fibonacci sequence. □

B. Relating the Lucas and Fibonacci numbers

There are several simple equations that relate the Lucas and Fibonacci numbers. We present a few of the most aestetically pleasing ones,[5] in terms of their exposing an intimate relationship between the two sequences.

Proposition 8.10 *For all $m, n \geq 1$*

$$(\mathbf{a})\;\; L(n) = F(n+1)+F(n-1) \tag{8.15}$$

$$(\mathbf{b})\;\; F(n+1) = \frac{1}{2}\big(F(n) \;+\; L(n)\big) \tag{8.16}$$

$$(\mathbf{c})\;\; F(m+n-1) = \frac{1}{2}\big(F(m)\cdot L(n) \;+\; F(n)\cdot L(m)\big) \tag{8.17}$$

$$(\mathbf{d})\;\; F(2n) = F(n)\cdot L(n). \tag{8.18}$$

Proof. We consider the three identities in turn.

[5] Aesthetically pleasing, that is, to the authors. As noted by the author Margaret Wolfe Ungerford in *Molly Bawn* (1878), "Beauty is in the eye of the beholder."

(a) We proceed by induction.

The base case. Equation (8.15) holds when $n = 1$ because

$$L(1) = 3 = F(2) + F(0) = 2 + 1.$$

The inductive hypothesis. Assume that equation (8.15) holds for $L(2), L(3), \ldots, L(n)$.

The inductive extension. Let us compute $L(n+1)$:

- By definition (8.13),

$$L(n+1) \;=\; L(n) \;+\; L(n-1). \tag{8.19}$$

- When we apply the inductive hypothesis to both addends in (8.19), we obtain (after rearranging terms):

$$L(n+1) \;=\; F(n+1) \;+\; F(n) \;+\; F(n-1) \;+\; F(n-2) \tag{8.20}$$

- Finally, we invoke the defining recurrence (8.6) of the Fibonacci numbers on the first two addends in (8.20) and on the last two addends. We thereby transform (8.20) to equation (8.15), which validates the latter identity.

Notice that a proof similar to the preceding one yields the identity $L(n) = F(n+2) + F(n-2)$. Similar, but more complicated, identities hold for larger arguments. For the cases $n+3$ and $n+4$, for instance, one can establish the following pair of identities.

$$L(n) = \frac{1}{2}(F(n+3) + F(n-3)) \tag{8.21}$$

$$L(n) = \frac{1}{3}(F(n+4) + F(n-4)) \tag{8.22}$$

Perfect exercises, no?
YES

(b) By direct calculation, we derive the desired result:

$$\begin{aligned} 2F(n+1) &= F(n+1) \;+\; F(n) \;+\; F(n-1) \\ &= L(n) \;+\; F(n). \end{aligned}$$

(c) This identity is verified via a somewhat complicated induction. We fix parameter n in the argument $F(m+n)$ and induce on parameter m.

The base case. Because $L(0) = F(0) = 1$ the instance $m = 0$ of identity (8.17) reduces to identity (8.16), which we have just proved. To wit,

$$F(n+1) \;=\; \frac{1}{2}\big(L(n) \;+\; F(n)\big) \;=\; \frac{1}{2}\big(F(0)\cdot L(n) \;+\; F(n)\cdot L(0)\big).$$

The inductive hypothesis. Let us assume that identity (8.17) holds for all $m \leq k$.

Again, we should check for every proof the indices, the basic expressions are on n, fine. The induction is on m or k up to n, and then, we derive for $n+1$...

The inductive extension. Let us focus on instance $m = k+1$ of identity (8.17). Note first that the classical Fibonacci recurrence (8.6) implies that

$$F(n+k+1) \;=\; F(n+k) \;+\; F(n+k-1).$$

When we apply the inductive hypothesis to both $F(n+k)$ and $F(n+k-1)$, we obtain the following two identities.

$$\begin{aligned} F(n+k) &= \frac{1}{2}\big(F(k-1)\cdot L(n) \;+\; F(n)\cdot L(k-1)\big) \\ F(n+k-1) &= \frac{1}{2}\big(F(k-2)\cdot L(n) \;+\; F(n)\cdot L(k-2)\big). \end{aligned}$$

Because both the Fibonacci and Lucas sequences obey the body of recurrence (8.6), the preceding equations combine to extend the induction. To wit,

$$\begin{aligned} 2F(n+k+1) &= 2F(n+k) \;+\; 2F(n+k-1) \\ &= \big(F(k-1)\cdot L(n) \;+\; F(n)\cdot L(k-1)\big) \;+\; \big(F(k-2)\cdot L(n) \;+\; F(n)\cdot L(k-2)\big) \\ &= L(n)\cdot\big(F(k-1) \;+\; F(k-2)\big) \;+\; F(n)\cdot\big(L(k-1) \;+\; L(k-2)\big) \\ &= L(n)\cdot F(k) \;+\; F(n)\cdot L(k) \end{aligned}$$

The thus-extended induction verifies identity (8.17).

(d) Identity ((8.18) is actually the case $m = n$ of identity ((8.17).

This validates our final identity, which completes the proof. □

8.2.3.2 Tree-Profile numbers

In the course of analyzing a genre of search tree called *2,3-trees*,[6] in [58, 72], a new number sequence was discovered. Named *Tree-Profile numbers* [65], this family of positive integers was found to be a close relative of the family of binomial coefficients, both in its defining recurrence and in the quite similar properties that the two families share.

[6] These search trees are the lowest-index instances of the *B-trees* that have proved so useful in database implementations [28].

A. Definition

The *tree-profile numbers* are a doubly-indexed family

$$\big\{P(n,k)\big\}_{n\geq 1;\ k\geq 0}$$

of positive integers specified by the following recursive definition.

$$\begin{aligned} P(n,0) &\equiv 1 \quad \text{for all } n \geq 1 \\ P(n,1) &= \begin{cases} 1 & \text{for } n = 1 \\ 2 & \text{for all } n > 1 \end{cases} \\ P(n+1,k+1) &= P(n,k) + 2P(n,k-1) \quad \text{for all } n > 1, k > 0 \end{aligned} \tag{8.23}$$

This somewhat complicated definition can be better understood with the help of an analogue of Pascal's Triangle that we call the *Tree-profile Triangle*. The reader may want to compare Fig. 8.2 with Fig. 8.10.

$P(n,k)$	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	…
$n=1$	1	1									
$n=2$	1	2	3	2							
$n=3$	1	2	4	7	8	4					
$n=4$	1	2	4	8	15	22	20	8			
$n=5$	1	2	4	8	16	31	52	64	48	16	
$n=6$	1	2	4	8	16	32	63	114	168	176	
$n=7$	1	2	4	8	16	32	64	127	240	396	
$n=8$	1	2	4	8	16	32	64	128	255	494	
$n=9$	1	2	4	8	16	32	64	128	256	511	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Fig. 8.10 A "prefix" of the Tree-Profile Triangle, for $n,k \leq 9$.

B. Relating Triangle-Profile numbers with binomial coefficients

Proposition 8.11 *For all $n \geq 1$ and all $k \geq 0$,*

$$P(n,k) \;=\; 2^{k-n} \cdot \sum_{i=0}^{2n-k} \binom{n}{i} \tag{8.24}$$

Proof. We proceed by induction on n.

The base case. The case $n = 1$ of (8.24) follows from the "boundary cases" of definition (8.23).

same remark as before about the indices...

The inductive hypothesis. Let us assume that (8.24) holds for all n up to (but not including) some integer m. Focus on an arbitrary Tree-Profile number $P(m,k)$.

- If $k \in \{0,1\}$, then the "boundary cases" of definition (8.23) assure us that

$$P(m,k) \;=\; 2^k \;=\; 2^{k-n}\cdot 2^n \;=\; 2^{k-n}\cdot \sum_{i=0}^{2n-k} \binom{n}{i}$$

- If $k > 1$, then the defining recurrence in (8.23) combines with the inductive hypothesis to yield:

$$\begin{aligned} P(m,k) &= P(m-1,k-1) \;+\; 2P(m-1,k-2) \\ &= 2^{k-m}\cdot \sum_{i=0}^{2m-k-2} \binom{m-1}{i} \;+\; 2^{k-m}\cdot \sum_{j=0}^{2m-k-1} \binom{m-1}{i} \\ &= 2^{k-m}\cdot \binom{m}{0} \;+\; 2^{k-m}\cdot \sum_{i=1}^{2m-k-1} \binom{m-1}{i} \;+\; \binom{m-1}{i-1} \\ &= 2^{k-m}\cdot \sum_{i=0}^{2m-k-1} \binom{m}{i}. \end{aligned}$$

The induction is thus extended, thereby establishing the proposition. □

Proposition 8.11 explains the proliferation of powers of 2 in the Tree-Profile Triangle.

Corollary 8.1 *For all $n > k$, $P(n,k) = 2^k$.*

Finally, we derive the successor Tree-Profile values that allow us to generate the Tree-Profile Triangle.

Proposition 8.12

$$(\mathbf{a})\;\; P(n,k+1) = 2P(n,k) - 2^{k-n+1}\binom{n}{k-n+1}$$

$$(\mathbf{b})\;\; P(n+1,k) = P(n,k) + 2^{k-n-1}\left[\binom{n}{k-n} + \binom{n+1}{k-n}\right]$$

Proof. The major recurrence in (8.23) can be decomposed into the following triplet of recurrences.

$$P(n,k) = P(n-1,k-1) \;+\; 2P(n-1,k-2) \tag{8.25}$$

$$P(n,k+1) = P(n-1,k) \;+\; 2P(n-1,k-1) \tag{8.26}$$

$$P(n+1,k) = P(n,k-1) \;+\; 2P(n,k-2) \tag{8.27}$$

We use the recurrences in this triplet to attack the two alleged recurrences in the proposition.

indent the first equation of the previous array

(a) Combining recurrences (8.25) and (8.26) leads, via Proposition 8.11, to the following chain of equalities[7].

$$\begin{aligned} P(n,k+1) \;-\; 2\,P(n,k) &= P(n-1,k) \;-\; 4P(n-1,k-2) \\ &= 2^{k-n+1}\cdot\left[\sum_{i=0}^{2n-k-3}\binom{n-1}{i} \;-\; \sum_{i=0}^{2n-k-1}\binom{n-1}{i}\right] \\ &= -2^{k-n+1}\cdot\left[\binom{n-1}{2n-k-2} \;+\; \binom{n-1}{2n-k-1}\right] \\ &= -2^{k-n+1}\cdot\binom{n}{k-n+1}. \end{aligned}$$

This chain thus yields part **(a)** of the proposition.

(b) This part of the proposition follows by direct calculation from recurrence (8.27) and Proposition 8.11. To wit,

$$\begin{aligned} P(n+1,k) \;-\; P(n,k) &= P(n,k-1) \;+\; 2P(n,k-2) \;-\; P(n,k) \\ &= 2^{k-n-1}\cdot\left[\sum_{i=0}^{2n-k+1}\binom{n}{i} \;-\; \sum_{i=0}^{2n-k+2}\binom{n}{i} \;-\; 2\sum_{i=0}^{2n-k}\binom{n}{i}\right] \\ &= 2^{k-n-1}\cdot\left[2\binom{n}{2n-k} \;+\; \binom{n}{2n-k+1}\right] \\ &= 2^{k-n-1}\cdot\left[\binom{n}{k-n} \;+\; \binom{n+1}{k-n}\right] \end{aligned}$$

This chain thus yields part **(b)** of the proposition, completing the proof. □

C. The summation formula for Triangle-Profile numbers

We observed in Proposition 8.3 that the binomial coefficients in successive rows of Pascal's Triangle sum to successive powers of 2. While not quite matching that level of elegance, we show now that the Tree-Profile numbers in successive rows of the Tree-Profile Triangle sum to 1 less than successive powers of 3.

Proposition 8.13 *For all* $n \in \mathbb{N}^+$,

$$S_n \;\stackrel{def}{=}\; \sum_{k=0}^{2n-1} P(n,k) \;=\; 3^n - 1. \tag{8.28}$$

[7] Nothing magical here! The idea to combine $P(n,k+1)$ and $-2P(n,k)$ is for removing the common term $P(n-1,k-1)$

Proof. We begin with the following consequence of Proposition 8.11:

$$S(n) \;=\; \sum_{k=0}^{2n-1} P(n,k) \;=\; 1 \;+\; \sum_{k=0}^{2n-1} P(n,k+1).$$

If we now invoke Proposition 8.12(a), then we find that

$$S(n) \;=\; 1 \;+\; 2 \cdot \sum_{k=0}^{2n-1} \Big(P(n,k) \;-\; 2^{k-n} \binom{n}{k-n+1} \Big).$$

We can combine the preceding expressions with the "restricted" Binomial Theorem (Theorem 6.1) to generate the following chain of equalities.

The first equality is not straightforward, I think an intermediate step is needed here...

$$\begin{aligned} S(n) &= \sum_{j=0}^{2n-1} 2^{n-j} \binom{n}{j} \;-\; 1 \\ &= 2^n \cdot \sum_{j=0}^{2n-1} 2^{-j} \binom{n}{j} \;-\; 1 \\ &= 2^n \cdot (3/2)^n \;-\; 1 \\ &= 3^n - 1. \end{aligned}$$

The summation formula (8.28) follows. □

8.2.4 ⊕ *Computing Products of Consecutive Fibonacci Numbers*

One feature of the Fibonacci numbers that has captivated an entire community of mathematically oriented people is the plethora of simply presented identities involving the numbers.[8] We now present a particularly beautiful identity.

Proposition 8.14 *For all $n \geq 1$,*

$$F(n) \cdot F(n-1) \;=\; \sum_{k=0}^{n-1} (F(k))^2. \tag{8.29}$$

Proof. One can observe identity (8.29) "in action" in Fig. 8.7. We augment this visual validation of the identity with the following induction.

Base case. Instance $n = 1$ of identity (8.29) is valid because

$$F(0) \cdot F(1) \;=\; 1 \cdot 1 \;=\; 1^2 \;=\; (F(0))^2.$$

[8] Indeed, an entire research journal, *The Fibonacci Quartlerly*, is dedicated to the mathematics of Fibonacci numbers and their kin, including the sharing of such identities.

Inductive hypothesis. Assume that identity (8.29) is valid for all $n \leq m$.

Inductive extension. Let us focus on the product $F(m+1) \cdot F(m)$. Invoking the defining Fibonacci recurrence (8.6) and the inductive hypothesis, we generate the following chain of identities.

$$\begin{aligned} F(m+1) \cdot F(m) &= (F(m) + F(m-1)) \cdot F(m) \\ &= (F(m))^2 + \big(F(m) \cdot F(m-1)\big) \\ &= (F(m))^2 + \big(\sum_{k=0}^{m-1} (F(k))^2\big) \\ &= \sum_{k=0}^{m} (F(k))^2. \end{aligned}$$

The induction is thus extended, which completes the proof. □

8.3 ⊕ Recurrences "in action": The Token Game

In order to truly appreciate the power of recurrences as an analysis tool, one must witness them "in action". To this end, we now describe the (single-player) combinatorial *Token Game*. By employing recurrences to analyze plays of the game, we are able to derive an optimal strategy for playing the game.

8.3.1 The Token Game

The equipment. For each $n \in \mathbb{N}^+$, the order-n version of the Token Game is played with a *bank* which has n slots, labeled $1, \ldots, n$, and with a *pile* of n tokens.

Initial and terminal configurations. Each play of the game begins with the bank empty and the pile full, as depicted in Fig. 8.11. The goal of each play is to transfer

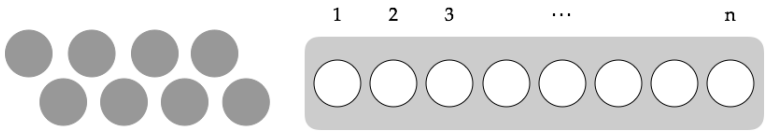


Fig. 8.11 The initial configuration of the Token Game: Each of the n tokens appears as a grey circle, and the empty bank has n slots. In the figure, $n = 7$.

all n tokens from the pile into the bank.

The repertoire of Game moves. The player transfers tokens from the pile to the bank by executing a sequence of *moves*. Each successive move has one of the following types.

1. Change the state of bank-slot #1, which is the first (i.e., leftmost) slot in the bank:
   If slot #1 is empty, then move a token from the pile to that slot.
   If slot #1 is full (i.e., contains a token), then remove this token and return it to the pile; see Fig. 8.12.

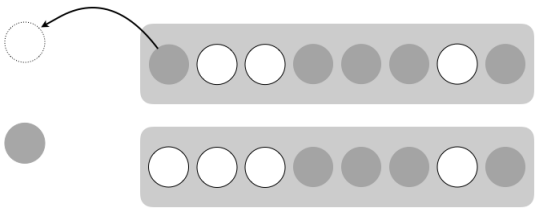


Fig. 8.12 (TOP) Slot #1 contains a token. (BOTTOM) Therefore, remove it (i.e., move it back to the pile).

2. Change the state of the bank-slot—call it slot #s—that is immediately to the right of the first (i.e., leftmost) *empty* slot:
   If slot #s is empty, then move a token from the pile to that slot.
   If slot #s is full (i.e., contains a token), then remove this token and return it to the pile; see Fig. 8.13.

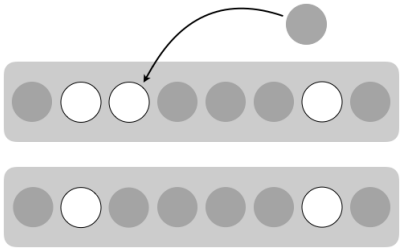


Fig. 8.13 (TOP) Bank-slot #s, which is immediately to the right of the first empty slot ($s = 3$ in this example) is empty. (BOTTOM) Therefore, move a token from the pile into slot #3.

Objective of a play of the Game: To minimize the number of moves from an initially empty bank to the finally full bank.

8.3.2 An Optimal Strategy for Playing the Game

The initial move of the game always involves a Type-1 move, because the bank is initially empty. But, thereafter, the player may possibly have access to more than one move. The question is how the player should choose successive moves, with the goal of filling the bank as quickly as possible.

While one can garner some strategic observations about how to play the Game by playing small instances—a trivial example: Do not choose two successive type-1 moves, because the second one just undoes the first—it is not easy to discern a complete strategy. In fact, though, one can rather easily specify a *recursive* solution to the game, which can be proved optimal using recurrences.

A recursive solution for Game instances with $n > 2$ can be derived from the following reasoning.

A token can be placed into the last bank-slot via the type-2 move

MOVE TOKEN FROM PILE INTO BANK-SLOT n.

In order for this move to be eligible for execution, the bank must be in the following configuration, reading rightward from bank-slot 1:

$$\big[\text{tokens in slots } 1,2,\ldots,n-2\big], \big[\text{no token in slot } n-1\big], \big[\text{no token in slot } n\big]$$

Once having achieved this configuration, and then executed the move

MOVE TOKEN FROM PILE INTO BANK-SLOT n,

the player can execute a sequence of $n-2$ type-2 moves of the form

MOVE TOKEN FROM BANK-SLOT k TO THE PILE,

for $k = n-2,\ n-3,\ldots,\ 1$, in turn. At this point, if one henceforth ignores the token in bank-slot n, then the player is now confronted with the initial configuration of the order-$(n-1)$ version of the Game. You can see the recursion coming!

Thus, the Game can be played via a recursion that iteratively executes the "super-steps" depicted in Figure 8.14, on successively smaller banks and piles. When the *active* size of the bank is k:

1. *Topmost bank configuration* $\longrightarrow$ *Second bank configuration*
   Move tokens into the leftmost $k-2$ slots of the bank, leaving the rightmost two slots empty.
2. *Second bank configuration* $\longrightarrow$ *Third bank configuration*
   Move a token into bank-slot k, i.e., the current rightmost slot
3. *Third bank configuration* $\longrightarrow$ *Fourth bank configuration*
   Empty bank-slots $1,2,\ldots k-1$; i.e., leave the current rightmost slot filled, but empty all slots to its left.
4. *Final bank configuration*
   The Game is complete!

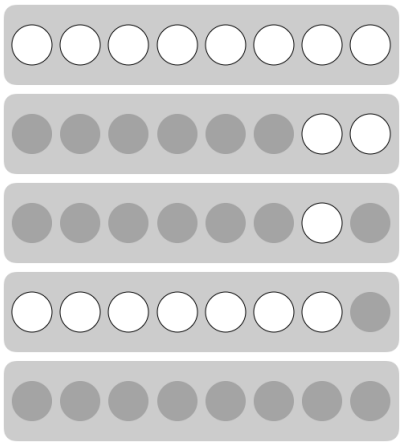


Fig. 8.14 A schematic of the recursive play of the current-sized version of the Game. The top four bank configurations indicate the iterating four supersteps in the recursions. The bottom bank configuration is the final one: the bank is entirely filled.

We can specify the preceding recursive algorithm in the following more formal format:

Recursive-Procedure Fill-Bank(n)

/* Fill the leftmost n slots of the currently empty bank */

Case	**Move-sequence**	**Action**	
$n = 1$		Move-Token to slot #1	via a Type-1 move
$n = 2$	1	Move-Token to slot #2	via a Type-2 move
	2	Move-Token to slot #1	via a Type-1 move
$n > 2$	1	Fill-Bank($n-2$)	via recursive invocation
	2	via a Type-2 move	
	3	Erase leftmost $n-2$ bank-slots	as described in text
	4	Fill-Bank($n-1$)	via recursive invocation

8.3.3 An analysis of the recursive strategy

We now develop an analysis of our recursive strategy for playing the Token Game. Not surprisingly, the analysis is embodied in a *recurrence* for the cost of playing the Game as a function of the bank-size n. Also not surprisingly, the structure of the recurrence mirrors the recursion structure of our playing strategy.

For $i = 1, \ldots, n$ let $f(i)$ denote the cost for filling the bank-slots from slot #1 through slot #i, measured in terms of the number of atomic moves, each of the form PLACE A TOKEN or REMOVE A TOKEN. Because of the dual forms of our atomic moves—each move fills one slot that is empty or empties one slot that is full—the

cost of filling an empty length-i prefix of the bank with tokens equals the cost of emptying a full length-i prefix of the bank. The total cost of a play of the Game is, by definition, the cost of filling the initially empty n-slot bank with tokens.

Proposition 8.15 *Let $f(n)$ be the cost of a play of the n bank-slot version of the Token Game. For all $n \in \mathbb{N}^+$, the value of $f(n)$ is*

$$f(n) \;=\; \begin{cases} \frac{1}{3}\left(2^{n+1}-1\right) & \text{if } n \text{ is odd} \\ \frac{1}{3}\left(2^{n+1}-2\right) & \text{if } n \text{ is even} \end{cases} \tag{8.30}$$

Proof. Our discussion has revealed that the analysis of our recursive playing strategy resides in solving the following recurrence.

$$f(n) \;=\; \begin{cases} 1 & \text{if } n=1 \\ 2 & \text{if } n=2 \\ f(n-1) \;+\; 2f(n-2)+1 & \text{if } n>2 \end{cases} \tag{8.31}$$

We can dramatically simplify recurrence (8.31) by focusing on the function

$$g(n) \stackrel{\text{def}}{=} f(n) \;+\; f(n-1) \quad \text{for } n \geq 2$$

instead of on f. Elementary calculation based on (8.31) shows that $g(n)$ satisfies the recurrence

$$g(n) \;=\; \begin{cases} 3 & \text{if } n=2 \\ 2g(n-1)+1 & \text{if } n>2 \end{cases} \tag{8.32}$$

We have, thereby, replaced the *bilinear* recurrence (8.31) by the *(singly) linear* recurrence (8.32). We learned in Section 6.2.2—specifically, see Proposition 6.5—how to evaluate the geometric summations that solve recurrences such as (8.32). In our case, we find that

$$g(n) \;=\; 2^{n-1} \;+\; 2^{n-2} \;+\cdots+\; 2^2 \;+\; 2 \;+\; 1 \;=\; 2^n-1 \tag{8.33}$$

We can now return to evaluating $f(n)$ via recurrence (8.31), in the light of our analysis of $g(n)$ in (8.32) and (8.33). We find that

$$f(n) \;=\; \begin{cases} 1 & \text{if } n=1 \\ g(n)-f(n-1) \;=\; (2^n-1)-f(n-1) & \text{if } n>1 \end{cases} \tag{8.34}$$

We begin to solve the *singly* linear recurrence (8.34) for $f(n)$ using the strategy we developed in Section 6.2.2; namely, we expand the recurrence in order to discern its pattern and then aanalyze the summation that the pattern leads us to. In this case, we find that

$$\begin{aligned} f(n) &= 2^n - 2^{n-1} + f(n-2) + 1 - 1 \\ &= 2^n - 2^{n-1} + 2^{n-2} - f(n-3) - 1 \\ &= 2^n - 2^{n-1} + 2^{n-2} - 2^{n-3} + f(n-4) + 1 - 1 \end{aligned}$$

$$\vdots$$

What we observe emerging—an inviting induction for the reader lurks in those words—is a geometric summation of powers of 2, with adjacent terms *alternating* in sign; the terminal units, ± 1, cancel after each pair of steps. We must be careful, though, because the numbers of terms in the summations differ based on the parity of n: when n is even, the last term is -2; when n is odd, the last term is -1.

We have now reached the *penultimate* step in finding the value of $f(n)$; specifically, we have derived the following parity-specified summations.

$$\text{For even values of } n \quad f(n) = \sum_{k=1}^{n} (-1)^k \, 2^k \tag{8.35}$$

$$\text{For odd values of } n \quad f(n) = \sum_{k=0}^{n} (-1)^{k+1} \, 2^k \tag{8.36}$$

Solving these parity-specific summations for $f(n)$ requires a moderate bit of mathematical dexterity. For pedagogical reasons, we illustrate two quite distinct approaches for determining the value of $f(n)$ for the case of odd n: we want to expose the reader to the quite-different intuitions that each approach elicits. We shall then derive the value of $f(n)$ for the case of even n from the value for the case of odd n.

An algebraic approach for the case of odd n. We look in detail at summation (8.36), which specifies $f(n)$ when n is odd, and we invoke algebraic manipulation to determine the value of $f(n)$ in this case.

In the following chain of equalities, we: gather the positive and negative terms in summation (8.36) [line 1 in the chain], perform some elementary manipulations on the result [lines 2 and 3 in the chain], and then invoke Proposition 6.5 [line 4 in the chain, which evaluates the resulting geometric summation]. We thereby find that, for odd values of n:

$$\begin{aligned} f(n) &= \left(2^n + 2^{n-2} + \cdots + 2\right) - \left(2^{n-1} + 2^{n-3} + \cdots + 1\right) \\ &= 2^{n-1} + 2^{n-3} + \cdots + 1 \\ &= 2^{n-1} \cdot \left(1 + \frac{1}{4} + \frac{1}{16} + \cdots + \frac{1}{2^{n-1}}\right) \\ &= \frac{1}{3}\left(2^{n+1} - 1\right) \end{aligned}$$

A geometric approach for the case of odd n. We begin again by looking again at summation (8.36). Then, noting that 2^{n-1} is a perfect square whenever n is odd, we set out to represent $f(n)$ as the aggregated area of a shrinking sequence of squares, of successive dimensions

$$2^{(n-1)/2} \times 2^{(n-1)/2}, \;\; 2^{(n-3)/2} \times 2^{(n-3)/2}, \;\; 2^{(n-5)/2} \times 2^{(n-5)/2}, \;\ldots, \;\; 1 \times 1$$

Fig. 8.15 depicts such a representation of $f(7) = 64 + 16 + 4 + 1 = 85$. To facilitate

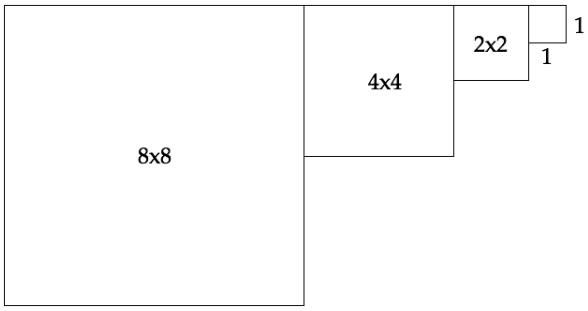


Fig. 8.15 A representation of summation (8.36) for the case $n = 7$.

our upcoming manipulation of the configuration depicted in the figure, let us refer to the configuration as *the cascade of squares determined by* $f(n)$. Note that, because each cascade is associated with an odd value of n, the smallest square in the cascade (at the far right in the figure) is the unit square, of dimensions 1×1; hence, it contributes $+1$ to the aggregate area of the cascade.

We can now use a geometric construction to evaluate $f(n)$ on an arbitrary odd argument n. We take three copies of the cascade in Fig. 8.15 and we manipulate the copies into the form is depicted in Fig. 8.16. In detail:

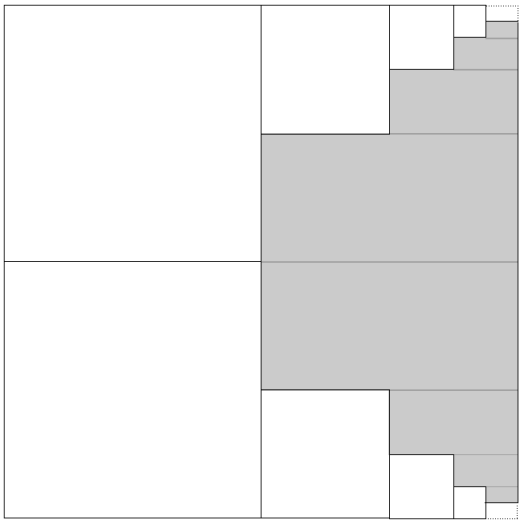


Fig. 8.16 Evaluating $f(n)$ for odd n by "almost" filling a large square.

1. We choose one of the three copies as the "anchor" of the construction. We position it in space so that it serves as the upper white cascade in Fig. 8.16.

2. We then take a second copy, flip it across the horizontal axis, and abut the top edge of its largest square with the bottom edge of the largest square in the anchor cascade. It then becomes the lower white cascade in Fig. 8.16.
   Note that, importantly, the abutted white cascades fit into a $2^{(n+1)/2} \times 2^{(n+1)/2}$ square.

   All of our observations about figures fitting within other figures are verified by direct calculations. These calculations are not too hard because all squares have side-dimensions that are powers of 2.

   Indeed, the top edge of the top white cascade and the bottom edge of the bottom white cascade lie, respectively, along the top and bottom edges of the $2^{(n+1)/2} \times 2^{(n+1)/2}$ square. *But*, the cascades' edges are 1 unit shorter that the edges of the big square; i.e., they both have length

$$2^{(n-1)/2} + 2^{(n-3)/2} + 2^{(n-5)/2} + \cdots + 1 = 2^{(n+1)/2} - 1$$

3. Finally, we take the third copy, color it grey, and nest it into the abutting white cascades in the following way.

   a. Take the biggest square in the grey cascade and nest it against the abutted biggest squares in the paired white cascades, in the manner depicted in Fig. 8.16. Note that the nest places one half of its biggest grey square abutting the biggest white square of the top white cascade and one half abutting the biggest white square of the bottom white cascade. Observe (from Fig. 8.16) that the resulting configuration fits within the $2^{(n+1)/2} \times 2^{(n+1)/2}$ square, and that the fit is *exact* along the left and right edges, which are shared by the abutting white cascades and the big square.
   b. For all of the other grey square, in decreasing order of size: We bisect—i.e., cut exactly in half—each square along its its equator, and we nest the resulting two halves of that square symmetrically within the abutting white cascades, in the manner depicted in Fig. 8.16. Once again, we observe that the resulting configuration fits within the $2^{(n+1)/2} \times 2^{(n+1)/2}$ square, and that the fit is *exact* along the left and right edges, which are shared by the abutting white cascades and the big square.
      The placement of the bisected squares from the grey cascade leaves two small *empty* regions within the $2^{(n+1)/2} \times 2^{(n+1)/2}$ square. The empty regions each has area $1/2$, because they are created by the "inadequate" placement of the bisected unit-side square from the grey cascade; the empty regions appear at the top right and bottom right corners of the $2^{(n+1)/2} \times 2^{(n+1)/2}$ square.

Once we have completed the described construction of the composite object depicted in Fig. 8.16, we calculate that the combined areas of the three cascades is one unit less that the area of the $2^{(n+1)/2} \times 2^{(n+1)/2}$ square (which, of course, has area 2^{n+1}). We have thus shown geometrically that $3f(n) + 1 = 2^{n+1}$, which *of course!* agrees with the value derived algebraically in (8.37).

We leave as an exercise the (somewhat easier) derivation of the value of $f(n)$ for the case of even n. □

My instinct is that the following example is not "pretty" or "dramatic" in the way that Proposition 8.14 is ... and it does not seem to teach any really new lessons.

This is another one we should discuss. As with the section "Another Identity", Cassini's Identity does not strike me as "pretty" as the "Consecutive products", and the proof does not open the way to much new methodology. I am troubled by Carroll's Puzzle because its resolution builds on principles that we do not cover anywhere. Since this is not a true paradox, this material does not belong in that section.

Right, put this as an excercice.

Both of my preceding comments build on the question, Why should we include this material? Obviously, when new techniques are involved, or when new, highly applicable, concepts are revealed, then the material should be included. In other situations, I am just pulled by my gut feeling. Should we discuss?

Chapter 9
COMBINATORICS, PROBABILITY, AND STATISTICS

9.1 Combinatorial interpretation of Fibonacci numbers

Let us count the number of binary vectors whose components donot have two consecutive 1. Call $F(n)$ this number.

Look at the last bit of the binary representation of n.

- If it is equal to 1 thus, the previous bit (in position $n-1$ should be 0. In this first case, the number the number is equal to $F(n-2)$
- If the last bit is 0, the number is $F(n-1)$

Thus, $F(n) = F(n-1) + F(n-2)$

This alternative view of looking at the Fibonacci numbers allows us to establish some elegant proofs. This is for instance the case for Property 8.5.

9.2 The Fundamentals of Counting

9.2.1 Binary Strings and Power Sets

Proposition 9.1 *For every integer $b > 1$, there are b^n b-ary strings of length n.*

Proof. The asserted numeration follows most simply by noting that there are always b times as many b-ary strings of length n as there are of length $n-1$. This is because we can form the set of b-ary strings of length n as follows. Take the set A_{n-1} of b-ary strings of length $n-1$, and make b copies of it, call them $A_{n-1}^{(0)}, A_{n-1}^{(1)}, \ldots, A_{n-1}^{(b-1)}$. Now, append 0 to every string in $A_{n-1}^{(0)}$, append 1 to every string in $A_{n-1}^{(1)}$, ..., append $\bar{b} = b-1$ to every string in $A_{n-1}^{(b-1)}$. The thus-amended sets $A_{n-1}^{(i)}$ are mutually disjoint (because of the terminal letters of their respective strings), and they collectively contain all b-ary strings of length n. □

Proposition 9.2 *The power set $\mathscr{P}(S)$ of a finite set S contain $2^{|S|}$ elements.*

Proof. Let us begin by taking an arbitrary finite set S—say of n elements—and laying its elements out in a line. We thereby establish a correspondence between S's elements and positive integers: there is the first element, which we associate with the integer 1, the second element, which we associate with the integer 2, and so on, until the last element along the line gets associated with the integer n.

Next, let's note that we can specify any subset S' of S by specifying a length-n *binary (i.e., base-2) string*, i.e., a string of 0's and 1's. The translation is as follows. If an element s of S appears in the subset S', then we look at the integer we have associated with s (via our linearization of S), and we set the corresponding bit-position of our binary string to 1; otherwise, we set this bit-position to 0. In this way, we get a distinct subset of S for each distinct binary string, and a distinct binary string for each distinct subset of S.

Let us pause to illustrate our correspondence between sets and strings by focussing on the set $S = \{a,b,c\}$. Just to make life more interesting, let us lay S's elements out in the order b,a,c, so that b has associated integer 1, a has associated integer 2, and c has associated integer 3. We depict the elements of $\mathscr{P}(S)$ and the corresponding binary strings in the following table.

Binary string	Set of integers	Subset of S
000	$\emptyset$	$\emptyset$
001	$\{3\}$	$\{c\}$
010	$\{2\}$	$\{a\}$
011	$\{2,3\}$	$\{a,c\}$
100	$\{1\}$	$\{b\}$
101	$\{1,3\}$	$\{b,c\}$
110	$\{1,2\}$	$\{a,b\}$
111	$\{1,2,3\}$	$\{a,b,c\} = S$

Back to the Proposition: We have verified the following: *The number of length-n binary strings is the same as the number of elements in the power set of S!* The desired numeration thus follows by the ($b = 2$) instance of Proposition 9.1. □

The binary string that we have constructed to represent each set of integers $N \subseteq \{0,1,\ldots,n-1\}$ is called the *(length-n) characteristic vector of the set N*. Of course, the finite set N has characteristic vectors of all finite lengths. Generalizing this idea, *every* set of integers $N \subseteq \mathbb{N}$, whether finite or infinite, has an *infinite* characteristic vector, which is formed in precisely the same way as are finite characteristic vectors, but now using the set $\mathbb{N}$ as the base set.

9.3 The Elements of Probability

within discrete frameworks, including introducing discrete probability/likelihood as a ratio:

$$\frac{\text{number of targeted events}}{\text{number of possible events}}$$

Elements of probability theory and statistics infuse every area of computing. The practicality of many algorithms that are experientially the most efficient for their target tasks depend on the *distribution* of inputs in "real" situations. Design methodologies for crucial complex circuits must acknowledge the *mean times to failure* of the critical components of the circuits. Sophisticated searching algorithms must take into account the relative *likelihoods* of finding one's goal in the various optional search directions. Analyzing and understanding large corpora of data requires a variety of methodologies that build on the concepts of *clustering* and/or *decomposition*.

A student needs at least an introduction to the foundations of probability and statistics to even understand, all the moreso to master, the terms highlighted in the preceding paragraph. We outline many of the key concepts that a student must be exposed to in the following subsections.

9.3.1 The Basic Elements of Combinatorial Probability

Perhaps the easiest and most engaging way to introduce "probability via counting" is by calculating the comparative likelihoods of various deals in 5-card poker and of various rolls of a pair of dice. The arithmetic required for this discussion is elementary and the "application" to gambling of interest even to non-gamblers: "Why is such a deal in poker (say, a straight) worth more than another (say, three of a kind)?" One can also introduce in this setting concepts such as randomness, bias, etc., that are so important in the design of experiments and the analysis of their outcomes.

9.4 Toward a Basic Understanding of Statistics

Most students whose interest tend to the empirical will likely "do" statistics with the aid of apps, rather than by explicitly writing programs that perform the required calculations. That said, all students should understand the crucial notion of *random variable* and should be conversant with the most common statistical distributions. "Conversant" in this context should include complete understandings of the (low-numbered) moments of *at least* the *uniform* and *exponential* distributions. They should know how to compute, say, the means and variances of various distributions and, most importantly, they should *understand* the sense in which the variance of a distribution give *important* information that is not available from the mean. All of this is prerequisite to rigor in experimentation.

9.4.0.1 The Elements of Empirical Reasoning

Empirical reasoning does not convey the certitude that formal reasoning does. Students should understand how to craft experiments in a way that collects the "right" data. The should then be able—perhaps just with statistical packages—to interpret the results they collect and to understand what conclusions are justifiable. *It is essential that all students understand the ditinction between* positive correlation *and* causation*!* (Most of the public would seem to flunk that test.)

In order to satisfy the preceding demands, students should understand enough about statistics—including definitions and meanings related to distributions and their moments—to understand what conclusions can be made based on experimental results, and to understand how to describe conclusions in a way that is supported by the statistics.

9.5 Beyond the Basics

As students are introduced to modern topics within computing, whether at the level of a Computing Literacy course or a post-core technical course, they will have to master a variety of more specialized topics that combine pieces of the elements we have discussed in this essay. While these topics are beyond the level of generality aimed at in this essay, some may be appropriate prerequisites to programs that have some specialized foci.

- Issues relating to *clustering* find application in applicatios as diverse as: *linear-algebraic computations*, *data mining*, *design and layout of digital circuitry*.
- Issues building on *graph separation/decomposition* are encountered when studying: *linear-algebraic computing*, *fault-tolerant design*, *load balancing*.
- Many issues relating to *fault/failure tolerance* and *data analytics* benefit from study using *random walks* (at least in one dimension).
- Many useful ideas regarding the *encoding and manipulation of data* can be gleaned from the elements of *information theory* and *computer arithmetic*.

The preceding list is really endless. Hopefully readers will be inspired by our few examples to compile a longer version that is appropriate for their particular environments.

Chapter 10
AN INTRODUCTION TO GRAPHS AND TREES

Graphs provide one of the richest technical and conceptual frameworks in the world of computing. They provide concrete representations of manifold data structures, hence must be well understood in preparation for a "Data Structures and Algorithms" course. They embody tangible abstractions of relationships of all sorts, hence must be well understood in order to discuss entities as varied as web-search engines and social networks with precision and rigor. As with most of the topics we discuss in this text, graph-oriented concepts must be taught "in layers". All students should be conversant with the use of graphs to represent and reason about a vast array of complicated relationships—ranging from taxonomies (including intra-family structures) to link-based data structures to interconnectivity within social media, and on and on—but the degree of sophistication that an individual student requires depends both on the abilities of the student and the range of graph-modeled concepts that will appear in the student's program. The most-basic concepts in this chapter should be understood by all students in any academic program that includes a computation-oriented component—although each concept can be developed with more texture and nuance within the context of specific application domains; the more advanced concepts should be selected with care, based on the instructor's perception of students' needs, in the light of the ever-growing importance of concepts involving interconnectivity.

Many developments in computing technology over recent decades have made it imperative that graphs no longer be viewed by students as the static objects introduced early in the history of computational studies. For instance, while it was innovative in the 1960s to employ graphs computationally as abstractions of data structures, such a view is standard today. Similar remarks, perhaps with differing dates, can be made about graphs as vehicles for representing the flow of control and information and as vehicles for representing interconnectivity among both concepts and populations. Applications ranging from databases to web-search engines to social networks demand an appreciation of graphs as dynamic objects. This change in perspective affects many aspects of the mathematical prerequisites for any academic program that includes a computation-oriented component.

10.1 Basic Concepts

10.1.1 Generic Graphs: Directed and Undirected

10.1.1.1 Connectivity-related concepts

The basic components of a graph $\mathscr{G}$ are its *nodes/vertices*[1] (one encounters both terms in the literature) and its *edges* that interconnect them. (The singular form of "vertices" is *vertex*.) When graph $\mathscr{G}$ is *undirected*, each of its edges connotes some sort of sibling-like relationship among nodes of "equal" status. When graph $\mathscr{G}$ is *directed* (sometimes referred to as a *digraph*), each of its edges connotes an "unequal" relationship such as parenthood or priority or dependence; edges in directed graphs are often termed *arcs*. In many situations involving directed graphs, it is important to deal with the *dual* of a digraph $\mathscr{G}$. This dual—which is usually denoted by some notational embellishment of "$\mathscr{G}$", such as $\widehat{\mathscr{G}}$—is the digraph obtained by *reversing* all of $\mathscr{G}$'s arcs. One sometimes encounters situations when arguments about, or operations on, a digraph $\mathscr{G}$ can be "translated" to arguments about, or operations on, $\widehat{\mathscr{G}}$ with only clerical effort. A *subgraph* $\mathscr{G}'$ of a graph $\mathscr{G}$ is a graph whose nodes are a subset of $\mathscr{G}$'s and whose edges are a subset of $\mathscr{G}$'s that interconnect only nodes of $\mathscr{G}'$. A *path* in an undirected graph is a sequence of nodes within which every adjacent pair is connected by an edge. A path is a *cycle* if all nodes in the sequence are distinct, except for the first and last, which are identical. Paths and cycles in directed graphs are defined similarly, except that every adjacent pair of nodes must be connected by an arc, and all arcs must "point in the same direction."

The special class of graphs called *trees* are identified mathematically as graphs that contain no cycles or, equivalently, as graphs in which each pair of nodes is connected by a unique path: a tree is thus the embodiment of "pure" connectivity (see Fig. 10.1 for an example of tree). As one would expect from the vernacular, a set of trees is called a *forest*.

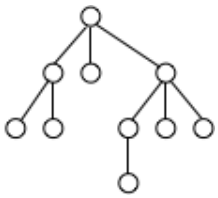


Fig. 10.1 A tree with 10 vertices.

A *directed graph* (*digraph*, for short) $\mathscr{G}$ is given by a set of *nodes* $\mathscr{N}_{\mathscr{G}}$ and a set of *arcs* (or *directed edges*) $\mathscr{A}_{\mathscr{G}}$. Each arc of $\mathscr{G}$ has the form $(u \to v)$, where

[1] The singular form of "vertices" is *vertex*.

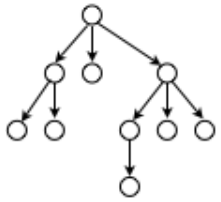


Fig. 10.2 A directed tree (out tree).

$u, v \in \mathscr{N}_{\mathscr{G}}$; we say that this arc goes *from u to v*. A *path* in the digraph $\mathscr{G}$ is a sequence of arcs that share adjacent endpoints, as in the following $(n-1)$-arc path from node u_1 to node u_n in $\mathscr{G}$:

$$(u_1 \to u_2),\ (u_2 \to u_3),\ \ldots,\ (u_{n-2} \to u_{n-1}),\ (u_{n-1} \to u_n) \tag{10.1}$$

The path (10.1) is often written in the more succinct form

$$u_1 \ \to\ u_2 \ \to\ u_3 \ \to \cdots \to\ u_{n-2} \ \to\ u_{n-1} \ \to\ u_n$$

The just-described path makes sense only when every node u_i belongs to $\mathscr{N}_{\mathscr{G}}$ and every one of its arcs, $(u_i \to u_{i+1})$, belongs to $\mathscr{A}_{\mathscr{G}}$. The *length* of path (10.1) is the number of arcs, i.e., $n-1$. The existence of the path means that the *distance* from u_1 to u_n in $\mathscr{G}$ is no greater than $n-1$. (There may exist shorter paths in $\mathscr{G}$ from u_1 to u_n.)

It is sometimes useful to endow the arcs of a digraph with labels from an alphabet Σ. When so endowed, the path (10.1) would be written in a form such as

$$(u_1 \overset{\lambda_1}{\to} u_2),\ (u_2 \overset{\lambda_2}{\to} u_3),\ \ldots,\ (u_{n-2} \overset{\lambda_{n-2}}{\to} u_{n-1}),\ (u_{n-1} \overset{\lambda_{n-1}}{\to} u_n)$$

where the λ_i denote symbols from Σ. Labeled paths also are often written in a succinct manner, as:

$$u_1 \overset{\lambda_1}{\to} u_2 \overset{\lambda_2}{\to} u_3 \overset{\lambda_3}{\to} \cdots \overset{\lambda_{n-3}}{\to} u_{n-2} \overset{\lambda_{n-2}}{\to} u_{n-1} \overset{\lambda_{n-1}}{\to} u_n$$

If $u_1 = u_n$, then we call path (10.1) a *(directed) cycle*, and we call its labeled version a *labeled (directed) cycle*. (The qualifier "directed" is usually included only for emphasis.)

An *undirected graph* $\mathscr{H}$ is given by a set of nodes $\mathscr{N}_{\mathscr{H}}$ and a set $\mathscr{E}_{\mathscr{H}}$ of 2-element subsets of $\mathscr{N}_{\mathscr{H}}$. Each of these subsets is called an *edge (of $\mathscr{H}$)*. One can, thus, view the undirected graph $\mathscr{H}$ as being obtained from a directed graph $\vec{\mathscr{H}}$ by removing the directionality of $\vec{\mathscr{H}}$'s arcs. Whereas we say:

the *arc* (u, v) goes *from* node *u to* node *v*

we say:

the undirected edge $\{u, v\}$ goes *between* nodes *u* and *v*

or, more simply:

the undirected edge $\{u, v\}$ *connects* nodes *u* and *v*.

One can view an undirected graph as asserting "pure" connectivity, whereas directed graphs assert some form of priority or directionality.

A *path* in an undirected graph is a sequence of edges—i.e., of 2-element sets of nodes—such that adjacent edges share a node. For illustration, an $(n-1)$-edge path that connects nodes u and v in the undirected graph $\mathscr{H}$ has the form

$$\{u,u_1\},\ \{u_1,u_2\},\ \{u_2,u_3\},\ \ldots,\ \{u_{n-2},u_{n-1}\},\ \{u_{n-1},v\} \tag{10.2}$$

The path described in (10.2) makes sense only when every node u_i belongs to $\mathscr{N}_{\mathscr{H}}$ and every edge $\{u_i\ u_j\}$ belongs to $\mathscr{E}_{\mathscr{H}}$. The *length* of path (10.2) is the number of edges—which is $n-1$ in this example; and the existence of the path means that the *distance between* u and v in $\mathscr{H}$ is no greater than $n-1$. (There may exist shorter paths that connect u and v.)

Undirected graphs are usually the default concept, in the following sense: When $\mathscr{G}$ is described as a "graph," with no qualifier "directed" or "undirected," it is usually understood that $\mathscr{G}$ is an undirected graph.

For each edge $\{u,v\} \in \mathscr{E}_{\mathscr{H}}$, we call nodes u and v *neighbors* (in $\mathscr{H}$). The *degree* of a node $u \in \mathscr{N}_{\mathscr{H}}$ is the number of neighbors that u has.

Even at this early moment in our study of graphs, we can observe a few important facts that can be useful when analyzing a broad range of computation-related issues involving graphs (either as auxiliary notions or as subjects of discourse).

Proposition 10.1 **(a)** *An n-node digraph $\mathscr{G}$ has no more than n^2 arcs.*

(b) *An n-node graph $\mathscr{H}$ has no more than* $\binom{n}{2}$ *edges.*

(c) *An n-node (connected) tree has precisely $n-1$ edges.*

Proof. **(a)** The set $\mathscr{A}_{\mathscr{G}}$ of arcs of $\mathscr{G}$ is a subset of the set of ordered pairs of nodes of $\mathscr{G}$. This latter number is clearly n^2, becaause one can choose the first node in a pair in n ways and then *independently* choose the second node in n ways.

(b) The stated number is the number of 2-node subsets of $\mathscr{N}_{\mathscr{H}}$. To wit, start by listing the n^2 ordered pairs of nodes of $\mathscr{H}$. First, eliminate from the list all n pairs whose first and second elements are equal: a set of the form $\{u,u\}$ has only one element, hence is not an edge of $\mathscr{H}$. Then, for each distinct pair of nodes $u,v \in \mathscr{N}_{\mathscr{H}}$, eliminate one of the two ordered pairs, $\langle u,v\rangle$ and $\langle u,v\rangle$: both of these ordered pairs lead to the same unordered set $\{u,v\}$, hence the same edge of $\mathscr{H}$. After these eliminations, we are left with

$$\frac{n^2-n}{2} = \binom{n}{2}$$

2-element subsets of $\mathscr{N}_{\mathscr{H}}$, from which we choose the edges of $\mathscr{H}$.

(c) Let us proceed by induction on n.

The base case $n=2$ is obvious, because one edge is both necessary and sufficient to connect two nodes.

Assume for induction that the indicated tally is correct for all trees having no more than k nodes.

Consider, for the purpose of extending the induction, any tree $\mathscr{T}$ on $k+1$ nodes. Easily, $\mathscr{T}$ must contain at least one leaf-node—i.e., a node v of degree 1—or else $\mathscr{T}$ would contain a cycle. If we remove v and its incident edge, we now have a tree on k nodes which, by induction, has $k-1$ edges. When we reattach node v, to restore $\mathscr{T}$ to its original state, we see that $\mathscr{T}$ has $k+1$ nodes and k edges.

Because $\mathscr{T}$ was an arbitrary $(k+1)$-node tree, the induction is extended. □

Proposition 10.2 *In any undirected graph, the number of nodes of odd degree is even.*

Proof. The result follows directly from the following equation that holds for any undirected graph $\mathscr{G}$:

$$\sum_{v \in \mathscr{N}_{\mathscr{G}}} \text{DEGREE}(v) \;=\; 2 \cdot |\mathscr{E}_{\mathscr{G}}|.$$

The equation holds because each edge e of $\mathscr{G}$ "touches" two nodes of $\mathscr{G}$, namely, e's two endpoints. Since the sum of $\mathscr{G}$'s node-degrees is even, each odd node-degree must be paired (in the sum) with another odd node-degree. □

We sometimes use the term *neighbor* within the context of *directed* graphs also. If we say that nodes u and v are neighbors in the directed graph $\mathscr{G}$, then we mean that $\mathscr{A}_{\mathscr{G}}$ contains at least one of the arcs $(u \to v)$ or $(v \to u)$. More typically, we use terminology that is more faithful to digraph $\mathscr{G}$'s directedness. If $\mathscr{A}_{\mathscr{G}}$ contains the arc $(u \to v)$, then we would call v a *(direct) successor* of u and we would call u a *(direct) predecessor* of v The term *parent* often replaces "predecessor node", and the term *child* often replaces "successor node", especially when $\mathscr{G}$ is a directed *tree*. Acknowledging the distinction between predecessors and successors in digraphs, we usually split the notion of the degree of a node within a digraph into the *indegree* and the *outdegree* of node u:

- The *indegree* of node $u \in \mathscr{N}_{\mathscr{G}}$ is the number of nodes $v \in \mathscr{N}_{\mathscr{G}}$ such that $(u \to v)$ is an arc of $\mathscr{G}$.
- Symmetrically, the *outdegree* of node $u \in \mathscr{N}_{\mathscr{G}}$ is the number of nodes $v \in \mathscr{N}_{\mathscr{G}}$ such that $(v \to u)$ is an arc of $\mathscr{G}$.

The reader will note that we nowhere guarantee that there is always a path that connects each node u with each other node v. We say that a graph $\mathscr{G}$ is *connected* if every pair of nodes $u, v \in \mathscr{N}_{\mathscr{G}}$ is connected by a path in $\mathscr{G}$. If graph $\mathscr{G}$ is *not* connected, then it is the disjoint union of some number c of connected subgraphs, usually called $\mathscr{G}$'s *(connected) components*. Of course, g is connected just when $c = 1$; i.e., there is a single connected component.

10.1.1.2 Distance-related concepts

Distance and diameter in a digraph. Extrapolating from our discussion of path (10.1): The *distance* from node u_1 to node u_n in the digraph $\mathscr{G}$ is the smallest number

of arcs in any path from u_1 to u_n. In detail:

$$\text{DISTANCE}(u_1,u_n)\ \begin{cases} = 0 & \text{if } u_1 = u_n \\ \leq n-1 & \text{if there is a path } (10.1) \text{ from } u_1 \text{ to } u_n \\ = \infty & \text{if there is no path } (10.1) \text{ from } u_1 \text{ to } u_n \end{cases} \tag{10.3}$$

The *diameter* of a directed graph $\mathscr{G}$ is the largest distance between two nodes of $\mathscr{G}$, i.e., the largest number d for which there exist nodes $u_1, u_n \in \mathscr{N}_{\mathscr{G}}$ such that $\text{DISTANCE}(u_1,u_n) = d$. Note that when discussing digraphs, we always use *directed* paths when defining distance.

Distance and diameter in an undirected graph. Extrapolating from our discussion of path (10.2): The *distance between* node u_1 and node u_n in the graph $\mathscr{G}$ is the smallest number of edges in any path from u_1 to u_n. In detail:

$$\text{DISTANCE}(u,v)\ \begin{cases} = 0 & \text{if } u = v \\ \leq n-1 & \text{if there is a path } (10.1) \text{ between } u \text{ and } v \\ = \infty & \text{if there is no path } (10.1) \text{ between } u \text{ and } v \end{cases} \tag{10.4}$$

The *diameter* of an undirected graph $\mathscr{H}$ is the largest distance between two nodes of $\mathscr{H}$, i.e., the largest number d for which the exist nodes $u_1, u_n \in \mathscr{N}_{\mathscr{H}}$ such that $\text{DISTANCE}(u_1,u_n) = d$. Note that when discussing undirected graphs, we always use *undirected* paths when defining distance.

Our discussion of internode distances within graphs have focused on shortest (or longest) path problems in *unweighted* graphs. A variety of important applications can be modeled via path-distance problems in graphs $\mathscr{G}$ each of whose edges, say, $\{u,v\}$, is weighted with a number that measures the cost of going between nodes u and v in $\mathscr{G}$. Of course, when graph $\mathscr{G}$ is directed, then the arcs $(u \to v)$ and $(v \to u)$ can have different weights, to model situations wherein going from u to v is easier/cheaper than going from v to u. Happily, determining shortest (or longest) paths in a directed or undirected graph $\mathscr{G}$ can be accomplished "efficiently"—which in the algorithmic world means "in a number of steps that is polynomial in the size of $\mathscr{G}$".

10.1.1.3 Matchings in graphs

The notion of a *matching* in a graph is fundamental to many situations that can be modeled using graphs. A *matching* in an undirected graph $\mathscr{G}$ is a set of edges of $\mathscr{G}$ that have no nodes in common. It is, thus, a formal mechanism for pairing nodes of a graph. The broad array of activities that can be modeled using graph matching include: pairing competitors for a tennis tournament; helping a person select a potential spouse (which even in the vernacular is often termed "matchmaking"); determining (near-)optimal layouts for a keyboard in language X (based on the relative "affinities" of various pars of letters for one another in language X); selecting persons to command the police stations in city Y (based on the perceived "match"

between a candidate's qualifications and the needs of specific stations). Even this small sampler of situations that involve matchings makes it clear that there are many variations on this formal theme. This section is devoted to describing, and briefly discussing, a few of the most commonly encountered versions of matching in graphs.

> Although the definitions of the various versions of matching are readily accessible to even the beginning student of mathematics, munch of the more sophisticated mathematical knowledge about matchings is beyond any beginning text. The interested reader might consult a more advanced source, such as [11], to get a feeling for what is known about this simple, yet rich, topic.

Matchings in unweighted graphs. The most straightforward notion of matching involves an undirected graph $\mathscr{G}$ with unlabeled edges. The optimization criterion most often invoked with this genre of matching is to maximize the number of edges of $\mathscr{G}$ that belong to the matching.

The target in this "vanilla-flavored" matching problem is often a matching that is *maximal*, in the sense that adding any further edge of $\mathscr{G}$ to the matching leaves one with a set of edges that is no longer a matching.

Among maximal matchings in a graph $\mathscr{G}$, the "ultimate treasure" is a matching that is *perfect*, in the sense that every node of $\mathscr{G}$ belongs to some edge of the matching.

Proposition 10.3 *Maximal matchings exist for any graph $\mathscr{G}$. One can find such a matching in a number of steps proportional to $|\mathscr{N}_{\mathscr{G}}|$.*

Proof. We leave to the reader the challenge of verifying that the following *greedy*[2] process satisfies the conditions of the Proposition.
The Process:
Begin by laying the nodes of $\mathscr{G}$ out, left to right, in any way. Repeat the following process until no nodes remain in the layout.
Select the leftmost node, u, in the remaining layout of $\mathscr{N}_{\mathscr{G}}$. Select the leftmost neighbor, v, of u that remains in the layout.

1. If you succeed in finding both u and v, then add edge $\{u, v\}$ to the matching we are building. Remove both u and v from the layout.
2. If there is no neighbor v of u in the remaining layout, then remove node u from the layout.

The real challenge here is to find a data structure that allows an efficient search for a "remaining" neighbor-node v at each step of the selction process. □

In contrast to maximal matchings, there exist myriad simple graphs that do not admit any perfect matching. Contemplating, for instance, matchings within any cycle with an odd number of nodes may prepare the reader for the challenge of verifying the following necessary condition for a graph to admit a perfect matching.

[2] In the world of algorithmics, the term "greedy" describes any process that seeks to satisfy a criterion as quickly as possible, with no consideration of how this choice affects future choices.

Proposition 10.4 *Let $\mathscr{G}$ be a graph that admits a perfect matching. Then:*

- *$\mathscr{G}$ has an even number of nodes.*
- *The cardinality of the (perfect) matching—i.e., the number of edges in the matching—is exactly $\frac{1}{2}|\mathscr{N}_{\mathscr{G}}|$.*

Matchings in weighted graphs. The other very popular genre of matching problem focuses on graphs each of whose edges, say, $\{u,v\}$, is weighted with a number that measure the "affinity" of nodes u and v for each other. The challenge is to find a matching that is *maximal* in the sense of having a cumulative sum of edge-weights that is not exceeded by any other matching's.

We note in closing that, while edge-weightings often complicate computational processing of graphs, they need not render such computations practically infeasible. For instance, the problem of discovering a perfect matching of minimal weight in an edge-weighted graph can be solved moderately efficiently—i.e., in a number of steps that is polynomial in the size of the graph. (An algorithm that achieves this efficiency can be based on the colorfully named *Hungarian assignment method*; see the original source [48] or the encyclopedic algorithms text [28].)

10.1.2 Trees

The special class of graphs called *trees* occupy a place of honor within both the mathematical field called *graph theory* and within the vernacular.

Mathematically speaking, a tree is a graph that contains no cycles, i.e., is *cycle-free*. Equivalently, a tree is a graph in which each pair of distinct nodes is connected by precisely one path. A tree is thus the embodiment of "pure" connectivity: it provides the minimal interconnection structure that provides paths that connect every pair of nodes. As one might expect from the vernacular, a set of trees is called a *forest*.

We have just given two distinct definitions of "tree". The reader should prove that both definitions define the same class of graphs.

Proposition 10.5 *Prove that the two definitions of "tree" are equivalent. In other words, prove that the following assertions about a connected graph $\mathscr{T}$ are equivalent, in the sense that one assertion holds if, and only if, the other does.*

- *The graph $\mathscr{T}$ is cycle-free.*
- *Each pair of distinct nodes of $\mathscr{T}$ is connected by precisely one path.*

A GOOD EXERCISE?

One of the major uses of trees and forests is as a way of succinctly "summarizing" the connectivity structure inherent in an undirected graph. This role is inherent in the notion of a *spanning tree* of a connected graph $\mathscr{G}$. A spanning tree of $\mathscr{G}$ is a tree $\mathscr{T}(\mathscr{G})$ whose node-set is identical to $\mathscr{G}$'s:

$$\mathcal{N}_{\mathcal{T}(\mathcal{G})} = \mathcal{N}_{\mathcal{G}}$$

and all of whose edges are edges of $\mathcal{G}$:

$$\mathcal{E}_{\mathcal{T}(\mathcal{G})} \subseteq \mathcal{E}_{\mathcal{G}}.$$

Not surprisingly, a connected graph $\mathcal{G}$ typically has *many* spanning trees. All such trees share $\mathcal{G}$'s node-set, but they may choose quite different sets of edges.

Chance for some exercises here

For a graph $\mathcal{G}$ that is not connected, we replace the notion of spanning tree of $\mathcal{G}$ with the analogous notion of a *spanning forest* of $\mathcal{G}$. We shall typically discuss only spanning trees in this section, leaving the reader to extrapolate the discussion to include spanning forests of unconnected graphs.

The major use of spanning trees in applications is to "summarize" the full connectivity structure of a graph—and of the entities that the graph models, such as a map, the layout of a museum, etc. When used in this way, the edges of spanning trees are typically *weighted*, in order to model a "cost" of incorporating that edge in the tree. The types of computational problem modeled via edge-weighted spanning trees include: the optimal placement of firehouses, or hospitals, in a town and the optimal deployment of security mechanisms in an art museum. Reflecting problems wherein edge-weights measure transit costs, it is a classical computational problem to seek a *minimum-weight spanning tree* (or, in the vernacular, a *minimum spanning tree*). Happily, this classical optimization problem can be solved within a number of steps that is linear in the number of edges of $\mathcal{G}$ [28].

Giving just a *hint* at a solution for MST to real beginners is probably useless

Just as with graphs, there is a *directed* version of trees which is formed by replacing the (unoriented) edges of an undirected tree by (oriented) arcs; see Fig. 10.2. Within a directed tree $\mathcal{T}$, one often says that an arc goes from a *parent node* to a *child node* . Extending this anthropomorphic metaphor, one often talks about the *ancestor(s)* and *descendant(s)* of a tree-node. We single out two special classes of nodes: A *root (node)* of $\mathcal{T}$ is defined by its having no entering arcs, i.e., indegree 0; a *leaf (node)* of $\mathcal{T}$ is defined by its having no exiting arcs, i.e., outdegree 0. The reader is certainly familiar with the use of rooted directed trees to represent family trees and corporate hierarchies.

Sociologically, the historical *atomic family tree* has two roots, representing the matriarch and patriarch of the family. The entirety of the tree represents a single family generationally, before any children form their own families. All child-nodes in this genre of tree are the roots of singly-rooted subtrees of the entire family tree. The leaves of the tree are the childless descendants of the roots. Note that, while we are using anthropomorphic language here, we could be discussing other genres of "family", as, e.g., many types of biological taxonomies.

Among rooted directed trees, an important subclass comprises those that have a *single root* which has a directed path to every other node. The length of each such directed path is often used to label the *generation* of the node at the end of the path:

root, child, grandchild, great-grandchild, etc. Every node of the tree is the root of a singly-rooted directed subtree of the entire tree. All subtrees that are rooted at nodes of the same generation are mutually disjoint.

A singly-rooted tree represents a *hierarchy*. Given two directed subtrees within a hierarchy, either the root of one of the subtrees is a descendant of the root of the other, or the two subtrees are are mutually disjoint.

More formally: *rooted trees* are a class of *acyclic* digraphs. Paths in trees which start at the root are often called *branches*. The *acyclicity* of a tree $\mathscr{T}$ means that for any branch of $\mathscr{T}$ of the form (10.1), we cannot have $u_1 = u_n$, for this would create a cycle. Each singly-rooted tree $\mathscr{T}$ has a designated *root node* $u_n \in \mathscr{N}_{\mathscr{T}}$ that resides at the end of a branch (10.1) that starts at $r_{\mathscr{T}}$ (so $u_1 = r_{\mathscr{T}}$) is said to reside at *depth* $n-1$ in $\mathscr{T}$; by convention, $r_{\mathscr{T}}$ is said to reside at depth 0. $\mathscr{T}$'s root $r_{\mathscr{T}}$ has some number (possibly 0) of arcs that go from $r_{\mathscr{T}}$ to its *children,* each of which thus resides at depth 1 in $\mathscr{T}$; in turn, each child has some number of arcs (possibly 0) to its children, and so on. For each arc $(u \rightarrow v) \in A_{\mathscr{T}}$, we call u a *parent* of v, and v a *child* of u, in $\mathscr{T}$; clearly, the depth of each child is one greater than the depth of its parent. Every node of $\mathscr{T}$ except for $r_{\mathscr{T}}$ has precisely one parent; $r_{\mathscr{T}}$ has no parents. A childless node of a tree is a *leaf*, i.e., a node of degree 1. The transitive extensions of the parent and child relations are, respectively, the *ancestor* and *descendant* relations. The *degree* of a node v in a tree is the number of children that the node has, call it c_v. If every nonleaf node in a tree has the same degree c, then we call c the *degree of the tree*.

It is sometimes useful to have a symbolic notation for the ancestor and descendant relations. To this end, we write $(u \Rightarrow v)$ to indicate that node u is an *ancestor* of node v, or equivalently, that node v is a *descendant* of node u. If we decide that we are not interested in *really distant* descendants of the root of a tree $\mathscr{T}$, then we can *truncate* $\mathscr{T}$ at a desired depth d by removing all nodes whose depths exceed d. We thereby obtain the *depth-d prefix* of $\mathscr{T}$.

10.1.3 Computationally Significant "Named" Graphs

The mathematical discipline called graph theory is an important source of formal aids for the activities of designing, analyzing, utilizing, and verifying computer systems. Of course, computer systems are designed by humans. Among other consequences of this fact is the observations that the graphs that are among the most commonly used to structure systems tend to be rather uniform in structure, in a variety of possible ways. Such graphs, when drawn, often exhibit a lot of structural symmetry. One popular form of symmetry is *degree regularity*: an undirected graph $\mathscr{G}$ is *regular* if all nodes of $\mathscr{G}$ have the same degree. Not surprisingly, there is a directed version of "regular" embodied in the symmetric notions of *in-regularity* and *out-regularity*.

We now describe five families of regular graphs that have proven useful over the history of digital computing, and we expose some basic properties of each, including its diameter. Each of these graphs is available in both a directed and an undirected version, although, as we note, one of these versions is more commonly encountered. We have selected these specific graphs for rather different reasons.

- The first two graphs, the *cycle-graph* of paragraph A and the *complete graph* of paragraph B were selected for respectively representing the lowest-degree and highest-degree graphs that share two properties: (1) every node of each graph is accessible from every other node; (2) all nodes "look alike" to someone traversing the graph. Elaborating on (2): If we put you down on a node of either graph, there is no way that you can determine the identity of that node. This is an important feature to ponder, because it is a simple instance of the anonymity problem inherent to many modern distributed computing environments. *How does one orchestrate cooperative activities when all agents "are identical"?*
- The remaining graphs, the *mesh and torus networks*[3] of paragraph C, the *hypercube network* of paragraph D, and the *de Bruijn network* of paragraph E, were selected for their importance within the world of parallel and distributed computing—as abstract platforms for developing efficient computational and communicational processes, and as abstract versions of the networks that underlie parallel architectures by interconnecting its processors.

Throughout, the parameters that describe our graph families range over the positive integers; i.e., each occurrence of n below ranges over $\mathbb{N}^+$.

10.1.3.1 The cycle-graph $\mathscr{C}_n$

For each positive integer $n \in \mathbb{N}^+$, both the *undirected order-n cycle-graph* $\mathscr{C}_n$ and the *directed order-n cycle-graph* $\widehat{\mathscr{C}}_n$ have *node-set*

$$\mathscr{N}_{\mathscr{C}_n} = \mathscr{N}_{\widehat{\mathscr{C}}_n} = \{0, 1, \ldots, n-1\}.$$

- $\mathscr{C}_n$ has n edges; its *edge-set* is

$$\mathscr{E}_{\mathscr{C}_n} = \big\{\{i, i+1 \bmod n\} \mid i \in \{0, 1, \ldots, n-1\}\big\}.$$

  Fig. 10.3 illustrates a cycle with 8 vertices.

  – $\mathscr{C}_n$ is a regular network: each node has degree 2.
  Specifically, each node i of $\mathscr{C}_n$ has its *predecessor* $i-1 \bmod n$ and its *successor* $i+1 \bmod n$.
  – $\mathscr{C}_n$ has diameter $\lfloor n/2 \rfloor$.

[3] These two structures, though distinct, are usually discussed together because they share so many important properties.

Direct calculation shows that $\mathscr{C}_n$'s diameter is no larger than this. The fact that this is, in fact, the graphs's diameter is witnessed by the distance between each node $k \in \mathscr{N}_{\mathscr{C}_n}$ and its antipodal node $k + \lfloor n/2 \rfloor \bmod n$.

- $\widehat{\mathscr{C}}_n$ has *arc-set*

$$\mathscr{A}_{\widehat{\mathscr{C}}_n} = \big\{(i \to i+1 \bmod n) \mid i \in \{0, 1, \ldots, n-1\}\big\}$$

  - $\widehat{\mathscr{C}}_n$ is a regular network: each node has the same indegree and the same outdegree. Coincidentally, both the common indegree and the common outdegree are 2.
  - $\widehat{\mathscr{C}}_n$ has (directed) diameter $n-1$.
    Of course, $n-1$ is an upper bound on the diameter of any n-node digraph. The fact that this is exactly the graph's diameter is witnessed by the directed distance from each node k of $\widehat{\mathscr{C}}_n$ to its predecessor node $k-1 \bmod n$.

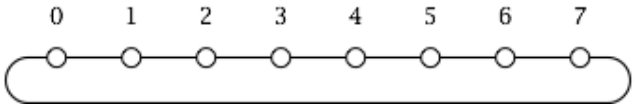


Fig. 10.3 A cycle of 8 vertices.

10.1.3.2 The complete graph, or, clique $\mathscr{K}_n$

For each positive integer $n \in \mathbb{N}^+$, we denote by $\mathscr{K}_n$ the *undirected* order-n *complete-graph* (or, *clique*), and by $\widehat{\mathscr{K}}_n$ the *directed* order-n *complete-graph* (or, *clique*). Both $\mathscr{K}_n$ and $\widehat{\mathscr{K}}_n$ have *node-set*

$$\mathscr{N}_{\mathscr{K}_n} = \mathscr{N}_{\widehat{\mathscr{K}}_n} = \{0, 1, \ldots, n-1\}.$$

- $\mathscr{K}_n$ has $\binom{n}{2}$ edges; its *edge-set* is

$$\mathscr{E}_{\mathscr{K}_n} = \big\{\{i, j\} \mid i, j \in \{0, 1, \ldots, n-1\},\ i \neq j\big\}.$$

  - $\mathscr{K}_n$ is a regular network: each node has degree $n-1$; every node $i \in \mathscr{N}_{\mathscr{K}_n}$ is connected with all other nodes.
  - $\mathscr{K}_n$ has diameter 1.
    $\mathscr{K}_n$'s diameter is a direct consequence of its node-degrees, and vice versa.

- $\widehat{\mathscr{K}}_n$ has $(n-1)n$ arcs; its *arc-set* is

$$\mathscr{A}_{\widehat{\mathscr{K}_n}} = \big\{(i \to j) \mid i, j \in \{0, 1, \ldots, n-1\}\big\},\ i \neq j$$

- $\widehat{\mathscr{K}}_n$ is a regular network: each node has the same indegree and the same outdegree. Both the common indegree and the common outdegree are $n-1$.
- $\widehat{\mathscr{K}}_n$ has (directed) diameter 1.
  As in the undirected case, there is a causal relationship between $\widehat{\mathscr{K}}_n$'s diameter and its (in- and out-) node-degrees.

Hearkening back to our discussion of matchings in (unweighted) graphs: The structure of the set of perfect matchings in general graphs is decidely nontrivial. For clique-graphs, though, the structure is much easier to discuss.

Proposition 10.6 *The number of perfect matchings admitted by the clique-graph $\mathscr{K}_n$ is either* 0*—if n is odd—or exponential in n if n is even.*

Proof. The assertion about cliques with odd numbers of nodes is immediate from Proposition 10.4.

We verify the assertion about cliques of the form $\mathscr{K}_{2k}$ by induction on k. To this end, let M_n denote the number of perfect matchings that the clique $\mathscr{K}_n$ admits.

The base of our induction is the case $k = 1$. Because $\mathscr{K}_2$ consists of a single edge, it admits only one perfect matching; i.e., $M_1 = 1$.

To garner intuition, we also explicitly solve the case $k = 2$, which is illustrated in Fig. 10.4. As the figure illustrates, $\mathscr{K}_4 = \mathscr{K}_{2\cdot 2}$ can be viewed as a 4-cycle (drawn

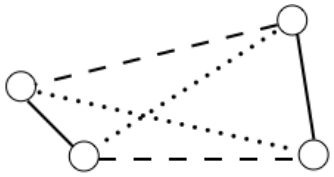


Fig. 10.4 The three different perfect matchings in the 4-node clique $\mathscr{K}_4$: the matchings' edges are drawn, respectively, with bold lines, dashed lines, and dotted lines.

with bold and dashed lines), augmented by two "cross-edges" (drawn with dotted lines). Easily, then, $\mathscr{K}_4$ admits 3 different perfect matchings, which can be identified (and specified) by the edge that contains the northwesterly node—call it v—in the figure. Node v has the choice of three nodes to "boldly" match with. (In the figure, v has chosen the southwesterly node as its "bold" match.) Once v has chosen its match, there is only one viable choice for the second edge in the matching. Thus, $M_2 = 3$.

We jump now to the case of any arbitrary $k > 2$. We remark that there are precisely $2k-1$ nodes of $\mathscr{K}_{2k}$ that node 1 can "choose" as its mate in a perfect matching. Once we set node 1 and its chosen mate aside, we confront an independent instance of the problem with parameter $k-1$, i.e., the problem of counting the number of perfect matchings in $\mathscr{K}_{n-2} = \mathscr{K}_{2k-2}$. We thereby note that as k grows, the

quantity M_k obeys the following recurrence:

$$M_k \;=\; (2k-1)\cdot M_{k-1}$$

In other words:

M_k is the product of the first k odd numbers.

To gauge the growth rate of M_k, we concentrate on cases $k > 2$ and ignore the $\lfloor k/2 \rfloor$ smallest odd numbers. We then replace each of the remaining odd numbers by its smallest possible value. We thereby find that

$$M_k \;=\; \prod_{i=1}^{k}(2i-1) \;\geq\; \prod_{\lceil k/2 \rceil}^{k}(2i-1) \;\geq\; (2\lceil k/2 \rceil - 1)^{k/2} \;>\; k^{k/2}$$

In summary, M_k grows exponentially with the parameter k, as claimed. □

The two families of graphs we have just discussed, cycles and cliques, are recommended to our attention by their structural simplicity—they epitomize, respectively, the most sparse way (the cycle) and the most dense way (the clique) to completely interconnect n nodes. The remainder of this section is devoted to three graph-structures that are structurally more complex than cycles and cliques, which have been designed to meet specific needs within the real technological world of computing and communicating. Indeed, the three upcoming graph families are among the most important ones when discussing parallel and distributed computing (PDC, for short). All three families have been used both to design computer architectures that support PDC and to craft algorithms that exploit the potential efficiencies—in computation and communication—that one can achieve using PDC.

10.1.3.3 The mesh ($\mathscr{M}_{m,n}$) and torus ($\widetilde{\mathscr{M}}_{m,n}$) networks

For positive integers $m, n \in \mathbb{N}^+$, both the $m \times n$ *mesh (network)* $\mathscr{M}_{m,n}$ and the $m \times n$ *toroidal network* (or, *torus*) $\widetilde{\mathscr{M}}_{m,n}$ have *node-set*

$$\begin{aligned}\mathscr{N}_{\mathscr{M}_{m,n}} \;=\; \mathscr{N}_{\widetilde{\mathscr{M}}_{m,n}} &= \{1,\,2,\ldots,\,m\} \;\times\; \{1,\,2,\ldots,\,n\} \\ &= \big\{\langle i,\,j\rangle \;\mid\; \big[i \in \{1,\,2,\ldots,\,m\}\big],\;\; \big[j \in \{1,\,2,\ldots,\,n\}\big]\big\}\end{aligned}$$

- $\mathscr{M}_{m,n}$ has $(m-1)n \;+\; (n-1)m$ edges; its *edge-set* is

$$\begin{aligned}\mathscr{E}_{\mathscr{M}_{m,n}} = \big\{&\{\{i,j\},\,\{i+1,j\} \;\mid\; 1 \leq i < m,\;\; 1 \leq j \leq n\} \\ &\cup \{\{i,j\},\,\{i,j+1\} \;\mid\; 1 \leq i \leq m,\;\; 1 \leq j < n\}\big\}\end{aligned}$$

  - · The subgraph of $\mathscr{M}_{m,n}$ defined by the node-set

$$\{\langle i,\,j\rangle \;\mid\; [i \in \{1,2,\ldots,m\}],\;\; [1 \leq j < n]\}$$

and all edges both of whose endpoints belong to that set is called the *i*th *row* of $\mathscr{M}_{m,n}$ Dually, the subgraph of $\mathscr{M}_{m,n}$ defined by the node-set

$$\{\langle i,\ j\rangle \ \mid\ [j \in \{1,2,\ldots,n\}],\ [1 \le i < m]\}$$

and all edges both of whose endpoints belong to that set is called the *j*th *column* of $\mathscr{M}_{m,n}$.

- Nodes $\langle 1,\ 1\rangle$, $\langle 1,\ n\rangle$, $\langle m,\ 1\rangle$, and $\langle m,\ n\rangle$ are the *corner nodes* (or, just *corners*) of $\mathscr{M}_{m,n}$.
- The path-graph consisting of the node-set

$$\{\langle 1,\ 1\rangle,\ \langle 1,\ 2\rangle,\ldots,\ \langle 1,\ n\rangle\}$$

together with all edges of $\mathscr{M}_{m,n}$ both of whose endpoints belong to this set, is the *top edge* of $\mathscr{M}_{m,n}$.
The other edges of $\mathscr{M}_n$ are defined analogously:
The *bottom edge* of $\mathscr{M}_{m,n}$ is the path-graph built upon the node-set

$$\{\langle m,\ 1\rangle,\ \langle m,\ 2\rangle,\ldots,\ \langle m,\ n\rangle\}$$

The *left edge* of $\mathscr{M}_{m,n}$ is the path-graph built upon the node-set

$$\{\langle 1,\ 1\rangle,\ \langle 2,\ 1\rangle,\ldots,\ \langle m,\ 1\rangle\}$$

The *right edge* of $\mathscr{M}_{m,n}$ is the path-graph built upon the node-set

$$\{\langle 1,\ n\rangle,\ \langle 2,\ n\rangle,\ldots,\ \langle m,\ n\rangle\}$$

- $\mathscr{M}_{m,n}$ is *not* a regular graph. Its corner nodes each has degree 2; its non-corner edge nodes each has degree 3; its *internal nodes*—which are all non-edge nodes—each has degree 4
- The diameter of $\mathscr{M}_{m,n}$ is $m+n-2$, as witnessed by the distance between nodes $\langle 1,\ 1\rangle$ and $\langle 2,\ n\rangle$.

- $\widetilde{\mathscr{M}}_{m,n}$ has $2mn$ arcs; its *arc-set* is

$$\begin{aligned}\mathscr{A}_{\widetilde{\mathscr{M}}_{m,n}} = \big\{&\{(\langle i,\ j\rangle \to \langle i+1 \bmod m,\ j\rangle)\ \mid\ 1 \le i \le m,\ \ 1 \le j \le n\}\\ &\cup\{(\langle i,\ j\rangle \to \langle i,\ j+1 \bmod n\rangle)\ \mid\ 1 \le i \le m,\ \ 1 \le j \le n\}\big\}\end{aligned}$$

  - The subgraph of $\widetilde{\mathscr{M}}_{m,n}$ defined by the node-set

$$\{\langle i,\ j\rangle \ \mid\ [i \in \{1,2,\ldots,m\}],\ [1 \le j \le n]\}$$

and all edges both of whose endpoints belong to that set is called the ith *row* of $\widetilde{\mathscr{M}}_{m,n}$ Dually, the subgraph of $\widetilde{\mathscr{M}}_{m,n}$ defined by the node-set

$$\{\langle i,\ j\rangle \ \mid\ [j \in \{1,2,\ldots,n\}],\ [1 \leq i \leq m]\}$$

and all edges both of whose endpoints belong to that set is called the jth *column* of $\widetilde{\mathscr{M}}_{m,n}$.

- $\widetilde{\mathscr{M}}_{m,n}$ is a regular network; each node has degree 4. Despite the fact that $\widetilde{\mathscr{M}}_{m,n}$ is an *undirected* graph, its arcs are commonly referred to via an anthropomorphic labeling, either as "up, down, left, and right" or as "north, south, west, and east".
- $\widetilde{\mathscr{M}}_{m,n}$'s diameter is $\lfloor m/2 \rfloor + \lfloor n/2 \rfloor$. This can be verified in analogy to the diameter of the cycle-graph $\mathscr{C}_n$.

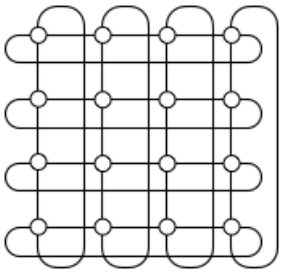


Fig. 10.5 The 4×4 torus.

10.1.3.4 The (boolean) hypercube network $\mathscr{Q}_n$

The graphs we focus on in this section have had a major impact on the world of coding, especially in regard to codes that are *error correcting* [62], and the world of computing, especially in regard to parallel and distributed computing [43, 77, 81]. The cited sources give a range of perspectives on the importance of *hypercube networks.*

The *order-n boolean hypercube*, traditionally denoted $\mathscr{Q}_n$, is the 2^n-node graph defined as follows.

- *The recursive definition.*
  - The order-0 boolean hypercube, $\mathscr{Q}_0$, has a single node, and no edges.
  - The order-$(k+1)$ boolean hypercube, $\mathscr{Q}_{k+1}$, is obtained by taking two copies of $\mathscr{Q}_k$, call them $\mathscr{Q}_k^{(1)}$ and $\mathscr{Q}_k^{(2)}$, and creating an edge that connects each node of $\mathscr{Q}_k^{(1)}$ with the corresponding node of $\mathscr{Q}_k^{(2)}$.

For illustration:

- $\mathscr{Q}_1$ consists of two nodes connected by a single edge.
- $\mathscr{Q}_2$ can be viewed as a "square", or equivalently, a copy of $\mathscr{C}_4$.
- $\mathscr{Q}_3$ can be viewed as a "cube", i.e., as two copies of $\mathscr{C}_4$ with edges connecting corresponding nodes: Each of the following pairs of nodes are connected by an edge: the upper right corner-nodes, the upper left corner-nodes, the lower right corner-nodes, and the lower left corner-nodes.

- *The direct definition*. For each $n \in \mathbb{N}$, the nodes of the order-n boolean hypercube, $\mathscr{Q}_n$, are all length-n binary strings. For illustration:

$$\begin{aligned}
\mathscr{N}_{\mathscr{Q}_0} &= \{\varepsilon\}, \quad \text{the length-0 } \textit{null string} \\
\mathscr{N}_{\mathscr{Q}_1} &= \{0,\ 1\} \\
\mathscr{N}_{\mathscr{Q}_2} &= \{00,\ 01,\ 10,\ 11\} \\
\mathscr{N}_{\mathscr{Q}_3} &= \{000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111\}
\end{aligned}$$

The iteration-based construction of big hypercubes from the next smaller ones is illusrated in Fig. 10.6.

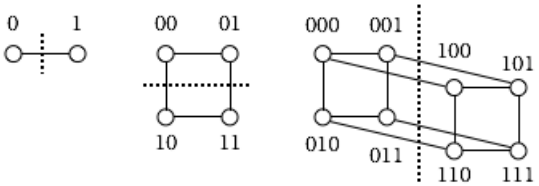


Fig. 10.6 The iteration-based construction of order-n hypercubes: Take two copies of the order-$(n-1)$ hypercube. Prepend a 0 to the node-labels of the first copy and a 1 to the node-labels of the second copy.

Easily, for each value of n, $\mathscr{Q}_n$ has 2^n nodes, for this is the number of length-n binary strings.

For each value of n, each edge of $\mathscr{Q}_n$ connects two node-strings that differ in precisely one position. This means that $\mathscr{Q}_n$ has $n2^{n-1}$ edges: To wit, each of its 2^n nodes has n neighbors, so the quantity $n2^n$ counts each of $\mathscr{Q}_n$'s edges twice—one for each endpoint. For illustration:

$$\begin{aligned}
\mathscr{A}_{\mathscr{Q}_1} &= \big\{\{0,\ 1\}\big\} \\
\mathscr{A}_{\mathscr{Q}_2} &= \big\{\{00,\ 01\},\ \{00,\ 10\},\ \{01,\ 11\},\ \{10,\ 11\}\big\} \\
\mathscr{A}_{\mathscr{Q}_3} &= \big\{\{000,\ 001\},\ \{000,\ 010\},\ \{000,\ 100\},\ \{001,\ 011\}, \\
&\qquad \{001,\ 101\},\ \{010,\ 011\},\ \{010,\ 110\},\ \{100,\ 101\}, \\
&\qquad \{100,\ 110\},\ \{101,\ 111\},\ \{011,\ 111\},\ \{110,\ 111\}\big\}
\end{aligned}$$

It is easy to observe $\mathscr{Q}_n$'s basic structural properties.

- $\mathscr{Q}_n$ is a regular network: each of its 2^n nodes has degree n.
  This follows from the fact that each arc of $\mathscr{Q}_n$ rewrites a single bit-position in the length-n binary string that is the arc's source node.
- $\mathscr{Q}_n$ has diameter $n = \ln(|\mathscr{N}_{\mathscr{Q}_n}|)$.[4]
  We address this issue formally.

Proposition 10.7 *For all $n \in \mathbb{N}^+$, $\mathscr{Q}_n$ has diameter $n = \ln(|\mathscr{N}_{\mathscr{Q}_n}|)$.*

Proof. We prove this diameter bound by construction. Focus on two arbitrary nodes of $\mathscr{Q}_n$:

$$x = \alpha_1\alpha_2\cdots\alpha_n \quad \text{and} \quad y = \beta_1\beta_2\cdots\beta_n$$

One of the several paths in $\mathscr{Q}_n$ from x to y is described schematically as the following left-to-right, bit-by-bit rewriting of x as y using arcs of $\mathscr{Q}_n$

$$x = \alpha_1\alpha_2\cdots\alpha_n \to \beta_1\alpha_2\cdots\alpha_n \to \beta_1\beta_2\cdots\alpha_n \to\cdots\to \beta_1\beta_2\cdots\beta_n = y$$

Since each bit of each string is rewritten once, the bound follows. □

The fact that $\mathscr{Q}_n$'s diameter is *logarithmic* in its size makes $\mathscr{Q}_n$ an efficient network for many tasks related to parallel computing and communication.

A powerful avenue for understanding the structure of a given family of networks is to understand how the perceived "shape" of graphs in the family can apparently change just by relabeling/renaming the nodes, or the edges/arcs, of the graphs. The formal mechanism for studying such relabelings/renamings is the concept of *graph isomorphism*. Let $\mathscr{G}$ and $\mathscr{H}$ be undirected graphs that have the same numbers of nodes and edges. (The following definition can easily be adapted to deal with *directed* graphs.) An *isomorphism* between $\mathscr{G}$ and $\mathscr{H}$ is a *bijection*[5]

$$\beta : \mathscr{N}_{\mathscr{G}} \leftrightarrow \mathscr{N}_{\mathscr{H}}$$

such that

- For each edge $\{u,v\}$ of $\mathscr{G}$ (i.e., $\{u,v\} \in \mathscr{E}_{\mathscr{G}}$), the doubleton set $\{\beta(u),\beta(v)\}$ is an edge of $\mathscr{H}$ (i.e., $\{\beta(u),\beta(v)\} \in \mathscr{E}_{\mathscr{H}}$).
- For each edge $\{x,y\}$ of $\mathscr{H}$ (i.e., $\{x,y\} \in \mathscr{E}_{\mathscr{H}}$), the doubleton set $\{\beta^{-1}(x),\beta^{-1}(y)\}$ is an edge of $\mathscr{G}$ (i.e., $\{\beta^{-1}(x),\beta^{-1}(y)\} \in \mathscr{E}_{\mathscr{G}}$).

We can immediately exemplify this notion via the following example.

Proposition 10.8 *The order-4 hypercube $\mathscr{Q}_4$ is isomorphic to the 4×4 torus $\widetilde{\mathscr{M}}_{4,4}$.*

Proof is an EXERCISE

We relegate the proof of this result to a Exercise, supplying as a hint the coding scheme (or, bijection) depicted in Fig. 10.7.

[4] Recall that $\ln n = \log_2 n$; see Section 5.4.1.2.

[5] Recall, from Chapter 3 that a bijection is a function that is one-to-one (i.e., injective) and onto (i.e., surjective).

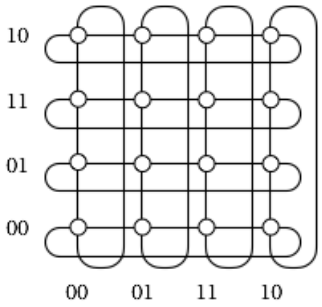


Fig. 10.7 Sketching the coding scheme that yields an isomorphism between the order-4 hypercube $\mathscr{Q}_4$ and the 4×4 torus $\widetilde{\mathscr{M}}_{4,4}$.

10.1.3.5 The de Bruijn network $\mathscr{D}_n$

While the family of hypercube networks has few competitors in the world of parallel and distributed computing, in terms of performance and ease of designing algorithms, it does have one major shortcoming regarding its realization in hardware. The basic problem is that the order-n hypercube has large node-degrees, specifically logarithmic in the size of the network. This feature makes the hypercubes actual performance much lower than its theoretical performance. Specifically, the size of the hypercube grows exponentially with the common degrees of the network's nodes—this is the "inverse" way of talking about logarithmic node-degrees—while the space in which we (and our computers) live grows only cubically with linear distance. The resulting disparity means that the wires in a large hypercube must inevitably be very long—in contrast to the unit-size of idealized network-edges. Consequently, electrical signals within a large hyeprcube must travel long distances, which means that the physical computer is much slower than its idealized version. (One finds a more technical discussion of this phenomenon in [84].)

The preceding shortcoming of hypercubes led researchers to seek a family of networks whose node-degrees stay constant even as one deploys successively larger instances of the network. A candidate such network was discovered within the domain of coding theorists (as, coincidentally, was the hypercube).

In the mid-20-century, Dutch mathematican Nicolaas Govert de Bruijn discovered a way to generate compact sequences that contain all possible strings of a prespecified length. Focussing on *binary* strings—although de Bruijn's strategy works for any finite alphabet—de Bruijn could generate a string of length $2^n + n - 1$ that contains every length-n binary string as a substring. Quite appropriately, such a string is called a *de Bruijn sequence*. It is not obvious that de Bruijn sequences exist for every n, but we now plant the seeds of a proof that they do. We begin by illustrating two sample sequences in (10.5).

n	LENGTH-n BINARY STRINGS	ORDER-n DE BRUIJN SEQUENCE
1	00, 01, 10, 11	00110
2	000, 001, 010, 011, 100, 101, 110, 111	0001110100

(10.5)

The table in (10.5) spawns many interesting questions:

- Do de Bruijn sequences exist for every n?
- If so,
  - How does one compute them?
  - Can one always find a de Bruijn sequence of length $2^n + n - 1$?
  - Can one find de Bruijn sequences of length $< 2^n + n - 1$?

(SOME GOOD EXERCISES HERE)

The answers to all of these questions—and the connection of de Bruijn sequences to the current chapter—reside in the family of directed graphs called *de Bruijn graphs* (or, *networks*). (The term used varies by intended application—mainly, coding theory and [the interconnection networks of] parallel computer architectures. We use the names interchangeably.)

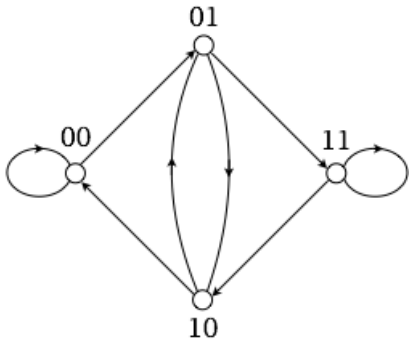


Fig. 10.8 The 4-node, order-2 de Bruijn network.

For every integer $n \in \mathbb{N}^+$, the *order-n de Bruijn network* is the *directed* graph $\mathscr{D}_n$ whose nodes comprise the set of length-n binary strings[6] The sets $\mathscr{N}_{\mathscr{D}_2}$ and $\mathscr{N}_{\mathscr{D}_3}$ appear in (10.5).

$\mathscr{D}_n$ is a regular directed graph; its nodes all have in-degree 2 and outdegree-2. Each node of $\mathscr{D}_n$ is a binary string of length $n \geq 1$; hence it can be written in the form βx, where $\beta \in \{0, 1\}$ is a *bit* (short for *binary digit*) and x is a length-$(n-1)$ binary string.

[6] While *binary* de Bruijn networks are the most frequently encountered ones, one can also find de Bruijn networks whose nodes comprise all length-n strings over larger finite alphabets. Such extended families also find applications in coding theory.

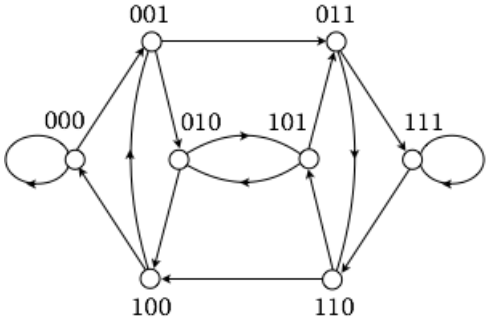


Fig. 10.9 The 8-node, order-3 de Bruijn network.

The 2^{n+1} arcs of $\mathscr{D}_n$ come in pairs specified as follows. For each $\beta \in \{0,1\}$ and for each length-$(n-1)$ binary string x, $\mathscr{D}_n$ has the two arcs

$$(\beta x \to x0) \quad \text{and} \quad (\beta x \to x1)$$

We enumerate $\mathscr{A}_{\mathscr{D}_3}$ in (10.6).

SOURCE NODE		TARGET NODE		TARGET NODE
000	GOES TO	000	AND TO	001
001		010		011
010		100		101
011		110		111
100		001		000
101		011		010
110		101		100
111		111		110

(10.6)

For each $n \in \mathbb{N}^+$, $\mathscr{D}_n$ has diameter n. To see why this is true, note that following any one of $\mathscr{D}_n$'s arcs, say from node x to node y, consists of "rewriting" the length-n string x as the length-n string y. The diameter bound therefore follows by showing that, for any two string-nodes of $\mathscr{D}_n$, say node u and node v, one can rewrite string u as string v by traversing a sequence of arcs—i.e., a directed path—of length at most n. Observe, for instance, that the path in $\mathscr{D}_3$ described schematically as follows

$$000 \ \to \ 001 \ \to \ 011 \ \to \ 111$$

leads node 000 to node 111, by rewriting string 000 as string 111. The diameter bound is now an immediate consequence of the following result. The result builds upon a notion that we have not yet encountered but will study in some detail in Section 10.2.2. .

A *Hamiltonian cycle* in an n-node graph $\mathscr{G}$ is a length-n cycle that contains every node of $\mathscr{G}$ precisely once. A *directed Hamiltonian cycle* in an n-node digraph $\mathscr{H}$ is a length-n directed cycle that contains every node of $\mathscr{H}$ precisely once.

Proposition 10.9 *For all $n \in \mathbb{N}^+$, $\mathscr{D}_n$ contains a* directed Hamiltonian cycle*, i.e., a length-2^n directed cycle of the form*

$$x \to y_1 \to y_2 \to \cdots \to y_{2^n-1} \to x \tag{10.7}$$

that contains every node of $\mathscr{D}_n$ precisely once; i.e.:

- $\{x, y_1, y_2, \ldots, y_{2^n-1}\} = \mathscr{N}_{\mathscr{D}_n}$.
- *All of the "y-nodes" that appear in cycle (10.7) differ from x and from each other.*

The simplest proof of this result has two steps, each of which introduces a topic we have not yet developed.

(1) For any directed graph $\mathscr{G}$, the *line digraph* of $\mathscr{G}$, denoted $\Lambda(\mathscr{G})$, is the following directed graph.

- The nodes of $\Lambda(\mathscr{G})$ are the arcs of $\mathscr{G}$:

$$\mathscr{N}_{\mathscr{L}(\mathscr{G})} = \mathscr{A}_{\mathscr{G}}$$

- For each pair of arcs of $\mathscr{G}$ of the form

$$\big[a_{x,y} = (x \to y)\big] \quad \text{and} \quad \big[a_{y,z} = (y \to z)\big]$$

  i.e, arcs such that the endpoint of the first arc is the source of the second arc, $\Lambda(\mathscr{G})$ contains an arc $(a_{x,y} \to a_{y,z})$.

The relevance of this topic to this section is that the line graph of every de Bruijn network $\mathscr{D}_n$ is the "next bigger" de Bruijn network, $\mathscr{D}_{n+1}$. Let us verify this claim.

Proposition 10.10 *For all $n \in \mathbb{N}^+$, $\mathscr{D}_{n+1}$ is the line digraph of $\mathscr{D}_n$: $\mathscr{D}_{n+1} = \Lambda(\mathscr{D}_n)$.*

Proof. Each node of $\Lambda(\mathscr{D}_n)$ is an arc of $\mathscr{D}_n$, hence has the form

$$(\beta x \to x\gamma)$$

for x a length-$(n-1)$ binary string and $\beta, \gamma \in \{0,1\}$. Let us associate node $\beta x \gamma$ of $\mathscr{D}_{n+1}$ with this node of $\Lambda(\mathscr{D}_n)$.

Note first that each arc of $\mathscr{D}_{n+1}$ has the form

$$(\delta y \varepsilon \to y \varepsilon \varphi),$$

where y is a length-$(n-2)$ binary string and $\delta, \varepsilon, \varphi \in \{0,1\}$. By our association of nodes of $\mathscr{D}_{n+1}$ with arcs of $\mathscr{D}_n$, this arc of $\mathscr{D}_{n+1}$ does, indeed, correspond to two successive arcs of $\mathscr{D}_n$. The first of these successive arcs *enters* node $y\varepsilon$ of $\mathscr{D}_n$; the second *leaves* that node.

Note next that, given any two successive arcs of $\mathscr{D}_n$, say

$$(\rho\sigma z \to \sigma z \tau) \quad \text{and} \quad (\sigma z \tau \to z \tau \xi)$$

where z is a length-$(n-2)$ binary string and $\rho, \sigma, \tau, \xi \in \{0,1\}$, there is, indeed, an arc of $\mathscr{D}_{n+1}$ of the form

$$(\rho\sigma z\tau \ \rightarrow \ \sigma z\tau\xi)$$

This means that the digraph $\mathscr{D}_{n+1}$ is identical to the digraph $\Lambda(\mathscr{D}_n)$, modulo a renaming of nodes and arcs.[7]

The described correspondence between the nodes and arcs of $\mathscr{D}_{n+1}$ and $\Lambda(\mathscr{D}_n)$ completes the proof. □

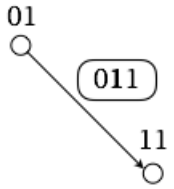


Fig. 10.10 Illustrating how to label each arc of a de Bruijn network by concatenating the labels of the nodes incident to the arc and compacting the common intermediate bits. In the depicted example, the node-labels 01 and 11 combine to yield the arc-label 011.

(2) *Eulerian cycles (or tours).* A *directed Eulerian cycle* in a digraph $\mathscr{G}$ is a directed cycle that contains each arc of $\mathscr{G}$ precisely once. We will see, later in this chapter, a truly elementary argument, based on node-degrees, which proves that every de Bruijn digraph has a directed Eulerian cycle. This demonstration will combine with Proposition 10.10 to complete the proof of Proposition 10.9. □

Stepping back from the structural specifics of $\mathscr{D}_n$, we now see that de Bruijn networks provide us with a *bounded-degree—specifically, a degree*-2 family of networks each of whose constituent graphs has *logarithmic diameter*! In this regard, at least, de Bruijn networks have exactly the same cost-performance as hypercubes—i.e., 2^n-node graphs with diameter n—with bounded degrees. Even more dramatic, it has been shown that sophisticated algorithmic techniques can achieve roughly equivalent computational efficiency, on a broad range of significant computational problems, using de Bruijn networks as using like-sized hypercubes [3, 12, 84].

10.2 Path and Cycle Discovery Problems in Graphs

Just as various genres of *spanning trees* are used to "summarize" aspects of the connectivity structure of a connected graph $\mathscr{G}$, various genres of *paths* and *cycles* in

[7] Technically, we are asserting that the digraphs $\mathscr{D}_{n+1}$ and $\mathscr{L}(\mathscr{D}_n)$ are *isomorphic* to one another. The topic of graph isomorphism is beyond the scope of this text, but our informal description provides all the details one would need to formalize the described isomorphism.

$\mathscr{G}$ are often useful to "summarize" aspects of $\mathscr{G}$'s traversal structure. This section is devoted to a range of problems related to determining the existence in a graph $\mathscr{G}$ of a path or a cycle that *completely* "summarizes" $\mathscr{G}$'s traversal structure, either by containing each node of $\mathscr{G}$ precisely once or by containing each edge of $\mathscr{G}$ precisely once. We shall generally focus in this section only on *undirected* paths and cycles in *undirected*, *unweighted* graphs. Extrapolating the notions we discuss to their directed analogues in directed and/or weighted graphs, will be accomplished via carefully crafted exercises. In a similarly, but simpler, vein, we shall generally discuss only prolems concerning *cycles*, leaving to the reader the analogous notions that concern paths. We begin by delimiting the two main classes of cycle-discovery problems that we study in this section.

Eulerian cycles (or, tours). A cycle in a graph $\mathscr{G}$ that traverses each of $\mathscr{G}$'s edges precisely once is called an *Eulerian cycle*, (or, often, an *Eulerian circuit*). The edge-exhausting cycles/circiuts/tours referred to by these several names were introduced as a topic of study in 1736 by the Swiss mathematician Leonhard Euler, whose name we have already encountered multiple times. Euler allegedly discovered the topic while contemplating how to design a tour of the town of Königsberg that would cross each of the town's bridges precisely once. The quest for Eulerian cycles in graphs is, thus, one of the oldest problems in the fields now called *Operations Research* and *graph theory*. (An edge-exhausting *path* in $\mathscr{G}$ is referred to in the obvious analogous way.) When one views an Eulerian cycle as a "map" for traversing a graph—as did Euler when contemplating this problem—one often calls the cycle an *Eulerian tour*. Traditionally, a graph that admits an Eulerian cycle is said to be *Eulerian*.

Dually: A cycle in a graph $\mathscr{G}$ that encounters each of $\mathscr{G}$'s nodes precisely once is called a *Hamiltonian cycle*, (or, often, a *Hamiltonian circuit*). This cycle-discovery problem is named in honor of the British mathematician William Rowan Hamilton, who is credited with inventing the concept in the mid-19th century. (A node-exhausting *path* in $\mathscr{G}$ is referred to in the obvious analogous way.) When one views a Hamiltonian cycle as a "map" for traversing a graph, one often calls the cycle a *Hamiltonian tour*. Traditionally, a graph that admits a Hamiltonian cycle is said to be *Hamiltonian*.

Despite the conceptual duality between the edge-encountering goal that underlies Eulerian paths and cycles, on the one hand, and the node-encountering goal that underlies Hamiltonian paths and cycles, these two graph-traversing goals differ in virtually every significant respect. It is rather easy to characterize the family of graphs that admit Eulerian tours and to find such a tour if it exists (Section 10.2.1); in contrast, there is no known characterization of the family of graphs that admit Hamiltonian tours, and the computational problem of efficiently determining whether a graph admits such a tour is one of the major classical problems in the field of computational complexity (Section 10.2.2).

10.2.1 Eulerian Cycles and Paths

The main results in this section characterize the families of directed and undirected graphs that admit an *Eulerian cycle* or an *Eulerian path*. The proofs of these characterizations are constructive: they consist of algorithms that efficiently find such a cycle or such a path when one exists. We focus on graphs that are connected: the algorithms we present can actually be adapted to find an *Eulerian cycle* or an *Eulerian path* in each connected component of a general graph. As we embark on our adventure, we note that the problem of finding an Eulerian cycle in a graph $\mathscr{G}$ is equivalent to the problem of drawing $\mathscr{G}$ without ever lifting one's pencil.

There is a simple and elegant characterization of graphs that admit Eulerian cycles and paths, which is accompanied by a simple and efficent algorithm for testing a given graph for finding such a tour in a graph that admits one. We encapsulate four versions of the desired result in the following statement.

Proposition 10.11 (Eulerian Cycles) **(a)** *An undirected graph $\mathscr{G}$ admits an Eulerian cycle if, and only if, $\mathscr{G}$ is connected and every node of $\mathscr{G}$ has even degree.*

(b) *A directed graph $\mathscr{H}$ admits a directed Eulerian cycle if, and only if, $\mathscr{H}$ is connected and, for every node v of $\mathscr{G}$,* $\text{INDEGREE}(v) = \text{OUTDEGREE}(v)$.

Proof. The *necessity* of the conditions in both parts **(a)** and **(b)** is established thus.

- By definition, a graph that is not connected does not admit any cycle.
- Each cycle that a graph admits accounts for two edges per node, because the cycle must "enter" the node by one edge and "exit" the node by a different edge.

We turn now to the verification of the *sufficiency* of the Proposition's conditions. Our argument resides in the following "streamlined" version of an induction—this is closer to the form of an induction that one would encounter in practice, rather than in a textbook.

The base case of the induction resides in proving the sufficiency of the Proposition's conditions for "small" graphs. While we largely leave this step to the reader, we do want to discuss the definition of "small". Until we understand how to deal with arbitrary n-node graphs, we will not know how the general step of our induction reduces the graph size n at each step. Because of this, it is a good practice to "play" with several small graph sizes initially. Hopefully, in addition to giving our inductive argument a robust base case, such "playing" will give us valuable intuition for the general case of the induction.

- Focusing on *undirected* graphs: We note that 2-node graphs cannot have even node-degrees—and, indeed, as promised by the Proposition, they cannot be Eulerian. One can exhaustively enumerate 3-node and 4-node graphs and verify that the Proposition correctly separates the Eulerian ones from the non-Eulerian ones.
- Focusing on *directed* graphs: We note that 2-node graphs can be Eulerian—as witnessed by the 2-node digraph each of whose nodes hosts a single arc that points to the other node. Once again, one can perform an exhaustive enumeration of 3-node and 4-node graphs andverify that the Proposition correctly separates the Eulerian ones from the non-Eulerian ones.

We now develop a complete proof of the *sufficiency* of the conditions in part **(a)** of the Proposition, for general n-node undirected graphs. Throughout the discussion, we intersperse hints regarding the sufficiency of the conditions in part **(b)** of the Proposition.

Let us consider a connected multi-node undirected graph $\mathscr{G}$ all of whose nodes have nonzero even degree.

Step 1 Initialize the *progress parameter* k to 0. This parameter will help orchestrate our discovery of the Eulerian cycle within $\mathscr{G}$.

Step 2 Choose an arbitrary node of $\mathscr{G}$ some of whose incident edges have not yet been traversed. Call this node v_k, and let us henceforth refer to v_k as a *special* node. Follow a walk along edges of $\mathscr{G}$ beginning at special node v_k. The rules of this walk are:

- Every step of the walk will traverse an edge of $\mathscr{G}$ that *has not yet been traversed* during any walk.
- The walk terminates when it encounters a node of $\mathscr{G}$ *all of whose edges have already been traversed*.

The facts that
(1) each node of $\mathscr{G}$ has even degree;
(2) the walk begins at node v_k (which crosses one of v_k's edges);
(3) no edge of $\mathscr{G}$ is traversed more than once
mean that the last node encountered in this walk is v_k. In other words: *This walk begins and ends with node* v_k.
Note that node v_k will occur in this walk multiple times—specifically with multiplicity $\frac{1}{2}\text{DEGREE}(v_k)$.

Step 3 Say that we have completed the walk in $\mathscr{G}$ that begins at node v_k.

If every edge of $\mathscr{G}$ has been traversed by the end of the current walk, then we *go to* **Step 4** and invoke procedure **Build Eulerian Cycle**, which stitches our series of walks into an Eulerian cycle in $\mathscr{G}$.

Else, there is a node of $\mathscr{G}$, call it v_{k+1}, one of whose incident edges has not yet been traversed. Note that we are here increasing the value of our progress parameter from k to $k+1$. We now *repeat* **Step 2** with the updated progress parameter, $k+1$.

Step 4 We have reached this step because our sequence of walks that begin and end at special nodes has terminated with all of $\mathscr{G}$'s edges having been traversed. We are now ready to stitch together these walks in order to expose an Eulerian cycle in $\mathscr{G}$. The resulting procedure **Build Eulerian Cycle** proceeds as follows. For each special node v_k, during the walk that begins and ends at v_k, we encounter other special nodes, call them $v_{k,1}$, $v_{k,2}$, ..., v_{k,m_k}, in the order of their being encountered along the walk. We then define the following recursive procedure.

Procedure Build Eulerian Cycle(v_k)

Phase 1	Follow the walk that begins at v_k until it encounters $v_{k,1}$ Execute **Build Eulerian Cycle**($v_{k,1}$)
Phase 2	Continue the walk that begins at v_k until it encounters $v_{k,2}$ Execute **Build Eulerian Cycle**($v_{k,2}$)
	• • •
Phase m_k	Continue the walk that begins at v_k until it encounters v_{k,m_k} Execute **Build Eulerian Cycle**(v_{k,m_k})
Phase m_k+1	Complete the walk that begins at v_k.

The process invocation

Execute **Build Eulerian Cycle**(v_0)

produces the Eulerian cycle in $\mathscr{G}$.

When dealing with a *directed* graph $\mathscr{H}$, we proceed exactly as with undirected graphs, with one critical difference: for each node v of $\mathscr{H}$, we always enter v along one of its *in-arcs*, and we always exit v along one of its *out-arcs*. As in the undirected case, each arc of $\mathscr{H}$ is traversed precisely once during the described process. □

The significance of the de Bruijn network in both coding and computation (as discussed in Section 10.1.3.5) lends considerable weight to the following important application of Proposition 10.11: When combined with Proposition 10.10, we obtain a proof of Proposition 10.9.

Corollary 10.1 *Every de Bruin network $\mathscr{D}_n$ is (directed)-Eulerian..*

The simplicity of the preceding characteriation degrades a trifle when one seeks Eulerian *paths* rather than *cycles*.

Proposition 10.12 (Eulerian Paths) **(a)** *An undirected graph $\mathscr{G}$ admits an Eulerian path if, and only if, $\mathscr{G}$ is connected and at most two nodes of $\mathscr{G}$ have odd degree.*

(b) *A directed graph $\mathscr{H}$ admits an Eulerian path if, and only if: $\mathscr{H}$ is connected; either $\mathscr{H}$ admits an Eulerian cycle, or $\mathscr{H}$ contains one node u such that*

$$\text{INDEGREE}(u) \;=\; \text{OUTDEGREE}(u)+1$$

and one node v such that

$$\text{INDEGREE}(v) \;=\; \text{OUTDEGREE}(v)-1.$$

The proof of the path-oriented Proposition 10.12 has the same overall structure as the cycle-oriented Proposition 10.11, with one major difference. Whereas a cycle has neither beginning nor end, a path has both. Proposition 10.12(a) asserts that an undirected graph which admits an Eulerian path but not an Eulerian cycle has precisely two nodes of odd degree. These odd-degree nodes play the role of the two ends of the Eulerian path. In similar fashion, Proposition 10.12(b) asserts that

a directed graph which admits an Eulerian path but not an Eulerian cycle contains one node, u, whose out-degree exceeds its in-degree and one node, v, whose in-degree exceeds its out-degree. Node u plays the role of the beginning node of the Eulerian path, and node v plays the role of the end node of the path. With these hints, we invite the reader to adapt the proof of Proposition 10.11 to obtain a proof of Proposition 10.12.

10.2.2 Hamiltonian Paths and Cycles/Tours

We turn now to the problem of determining when a connected graph $\mathscr{G}$ has a *Hamiltonian cycle*—and the allied problem of finding such a cycle when one exists. One can envision a number of benefits rendered accessible by the presence of a Hamiltonian cycle in a graph $\mathscr{G}$. Most obviously, the cycle specifies a tour of $\mathscr{G}$ (or of a map whose structure $\mathscr{G}$ abstracts) which visits each of $\mathscr{G}$'s nodes precisely once. This is the sense in which the cycle "sumarizes $\mathscr{G}$'s traversal structure.

10.2.2.1 Seeking more inclusive notions of Hamiltonianicity

Many graphs—even ones with "nice" structures—do not admit Hamiltonian cycles. The reader can generate *mesh-graphs* (Section 10.1.3.3) that admit no Hamiltonian cycle. The existence of such non-Hamiltonian graphs has spawned several independent paths of inquiry. One path seeks "modest" ways to weaken the property of *Hamiltonianicity* in a way that retains many of Hamiltonianicity's benefits while encompassing a braoder range of graph structures. We describe two avenues toward weakened, more inclusive, notions of Hamiltonianicity.

Be satisfied with paths, rather than cycles. A *Hamiltonian path* in a graph $\mathscr{G}$ is a path that passes through each of $\mathscr{G}$'s nodes precisely once. Hamiltonian paths can easily be shown to be a strictly weaker notion than Hamiltonian cycles, in the following sense. Obviously, every graph that admits a Hamiltonian cycle also admits a Hamiltonian path: one just drops any single edge of such a cycle to obtain such a path. However, there are many graphs that admit a Hamiltonian path that do not admit any Hamiltonian cycle. As suggested earlier, there exist mesh-graphs that admit no Hamiltonian cycle, even though every mesh-graph admits a Hamiltonian path. This latter claim is verified by a path that traverses the rows of a mesh-graph *seriatim*, in alternating directions.

Be satisfied with short paths, rather than edges. A Hamiltonian cycle in graph $\mathscr{G}$ is a circular enumeration of $\mathscr{G}$'s nodes in which adjacent nodes are at unit distance from one another—i.e., are connected by an edge. We can weaken (or, generalize) this notion to create a *Hamiltonian k-cycle* in $\mathscr{G}$, for any positive integer k: This is a circular enumeration of $\mathscr{G}$'s nodes in which adjacent nodes are at distance $\leq k$ from one another—so a Hamiltonian 1-cycle is what we have been calling a Hamiltonian cycle. In fact, the following result shows that one need not let k be very big before

one encompasses all connected graphs. Regrettably, the proof of this result is beyond the current text.

Proposition 10.13 **(a)** [24] *Let $\mathscr{G}$ be any connected graph. One can cyclically enumerate the nodes of $\mathscr{G}$ in such a way that nodes that are adjacent in the cycle are at distance ≤ 3 in $\mathscr{G}$.*
(b) [34] *Let $\mathscr{H}$ be any graph that is* 2-connected *(or, biconnected) in the sense that, for every two nodes, u and v, of $\mathscr{H}$, there exist at least two node-disjoint paths in $\mathscr{H}$ that connect u and v. One can cyclically enumerate the nodes of $\mathscr{H}$ in such a way that nodes that are adjacent in the cycle are at distance ≤ 2 in $\mathscr{H}$.*

10.2.2.2 Understanding Hamiltonianicity in "named" graphs

Yet another direction of inquiry is to determine whether specific graphs of interest are Hamiltonian. We illustrate this avenue by reviewing the five important families of graphs we studied in Section 10.1.3.

Proposition 10.14 **(a)** *Every cycle-graph $\mathscr{C}_n$ is Hamiltonian.*
(b) *Every clique-graph $\mathscr{K}_n$ is Hamiltonian.*
(c)*.1. For all m,n: the mesh-graph $\mathscr{M}_{m,n}$:*
(i) is path-Hamiltonian.
(ii) contains no odd-length cycle; hence, is not Hamiltonian if mn is odd.
(iii) is Hamiltonian whenever mn is even
(c)*.2. For all m,n: the torus-graph $\widetilde{\mathscr{M}}_{m,n}$ is Hamiltonian.*
(d) *Every hypercube $\mathscr{Q}_n$ is Hamiltonian.*
(e) *Every de Bruijn network $\mathscr{D}_n$ is (directed)-Hamiltonian.*

Proof. **(a)** This is a tautology, by definition of $\mathscr{C}_n$.

(b) This is immediate because, by definition, $\mathscr{K}_n$ contains every n-node graph—including $\mathscr{C}_n$—as a subgraph.

(c).1.i. As we noted earlier in the text, one can "snake" a path through $\mathscr{M}_{m,n}$, row by row, from the top-most to the bottom-most. By "snake", we mean that one should traverse adjacent rows in alternating directions.
(c).1.ii. This is a consequence of the fact that $\mathscr{M}_{m,n}$ is *bipartite*: One can color $\mathscr{M}_{m,n}$'s nodes ed and blue in such a way that every edge connect nodes of different colors. Details are left to the reader.
(c).1.iii. We sketch the construction of a Hamiltonian cycle in $\mathscr{M}_{m,n}$ when mn is even. Say, with no loss of generality that m is even, so that $\mathscr{M}_{m,n}$ has an even number of rows. Temporarily remove column 1 of $\mathscr{M}_{m,n}$, and consruct the "snaking" Hamiltonian path described in part **(c)**.1.i of this proof. Because m is even, this path begins and ends in column 2 of $\mathscr{M}_{m,n}$. One can, therefore, replace column 1 and use it to connect the ends of the "snaking" Hamiltonian path. This describes a Hamiltonian cycle in $\mathscr{M}_{m,n}$.

(c).2. When mn is even, the Hamiltonianicity of $\widetilde{\mathscr{M}}_{m,n}$ follows from the fact that $\mathscr{M}_{m,n}$ is a spanning subgraph of $\widetilde{\mathscr{M}}_{m,n}$. (Think about it!) When mn is odd, one needs just traverse $\widetilde{\mathscr{M}}_{m,n}$'s nodes row by row, going to the cyclically next node after completing each row. Details are left to the reader.

(d) One can craft a Hamiltonian cycle in $\mathscr{Q}_n$ by generating an *order-n binary reflected Gray code*—so named for its inventor, Bell Laboratories researcher Frank Gray; see [62]. Such a "code" is a cyclic enumeration of all 2^n binary strings of length n having the property that cyclically adjacent strings differ in only one bit-position. Length-n strings x_i and x_j are *cyclically adjacent* in the Gray code $\langle x_0, x_1, \ldots, x_{2^n-1}\rangle$ if $j = i+1 \bmod 2^n$.
It is computationally easy to generate an order-n Gray code from an order-$(n-1)$ Gray code, as follows.

We note first that the order-1 code is the sequence $\langle 0, 1\rangle$.

Inductively, to generate the order-$(k+1)$ Gray code from the order-k code:

- Concatenate the order-k code with a *reversed* copy of itself. (It is the code-sequence that is reversed, not the individual strings. For instance, as we go from the order-2 code $\langle x_0, x_1, x_2, x_3\rangle$ to the order-3 code, we concatenate that sequence with $\langle x_3, x_2, x_1, x_0\rangle$.)
- Augment each length-k string in one copy of the order-k Gray code to length $(k+1)$ by prepending a 0 to each string; and, augment each length-k string in the other (reversed) copy of the order-k Gray code to length $(k+1)$ by prepending a 1 to each string.

The following table illustrates the first few steps of this process.

Order 1	Order 2	Order 3	Order 4
0	00	000	0000
1	01	001	0001
	11	011	0011
	10	010	0010
		110	0110
		111	0111
		101	0101
		100	0100
			1100
			1101
			1111
			1110
			1010
			1011
			1001
			1000

(10.8)

We now sketch a proof that for each index $n \in \mathbb{N}^+$, the order-n Gray code sequence specifies a Hamiltonian cycle in $\mathscr{Q}_n$; exercises will give the reader the opportunity to fill in details. We verify the following two assertions in turn:

1. *The order-n Gray code contains all 2^n length-n binary strings.*

2. *Every pair of cyclically adjacent strings in the order-n Gray code differ in a single bit-position.*

(1) We sketch the induction. When $n = 1$, the Gray code consists of the two distinct strings 0 and 1. Assume that the assertion holds for $n = k$. The order-$(k+1)$ code is obtained by taking two copies of the order-k code and prepending 0 to the strings in one copy and 1 to the strings in the other copy. The 2^k distinct binary strings from the order-k code thereby produce 2^{k+1} distinct binary strings in the order-$(k+1)$ code.

(2) We distinguish three situations. Let the adjacent strings be string x, which appears in position i of the code, and string y, which appears in position $i+1 \bmod 2^n$ of the code.

- Say that $i = 2^n - 1$. In this case x is the last string in the code, and y is the first string. By the "refective" nature of the construction of the code, we know that $x = 1z$ and $y = 0z$ for some length-$(n-1)$ binary string z. Strings x and y therefore differ in precisely one bit-position, namely, bit-position 0.
- Precisely the same argument shows that when $i = 2^{n-1} - 1$, strings x and y again differ precisely in bit-position 0.
- In all other cases, namely, when $i \in \{0, 1, \ldots, 2^n - 1\} \setminus \{2^{n-1} - 1, 2^n - 1\}$, strings x and y share the same first bit-position. In fact, for some bit $\beta \in \{0, 1\}$, $x = \beta u$ and $y = \beta v$ for length-$(n-1)$ binary strings u and v which are cyclically adjacent in the order-$(n-1)$ Gray code. By an inductive argument, u and v differ in precisely one bit-position—which means that x and y also differ in precisely one bit-position.

(e) By Proposition 10.10, each de Bruijn network $\mathscr{D}_n$ is the line-digraph of the next bigger de Bruijn network, $\mathscr{D}_{n+1}$. Therefore, by definition of "line (di)graph", the fact that $\mathscr{D}_n$ is (directed)-Eulerian (Corollary 10.1) means that $\mathscr{D}_{n+1}$ is (directed)-Hamiltonian. □

10.2.2.3 Testing general graphs for Hamiltonianicity

The techniques we used in Subsection B to investigate the Hamiltonianicity of our "named" graphs exploit the detailed structures of the individual graphs. Thus, we cannot expect the proof of Proposition 10.14 to suggest avenues for determining whether an arbitrary given graph is Hamiltonian. In fact, quite sophisticated results proved in the early 1970s make a strong mathematical argument that no set of case studies is likely to have a major impact on the problem of testing general graphs for Hamiltonianicity.

The backstory of the preceding assertion began in 1971 when Stephen A. Cook was able to adapt, in [27], arguments[8] underlying the mathematical foundations of the theory of computing to the study of specific computational problems; the

[8] These arguments were the legacy of intellectual giants such as Kurt Gödel (in [38]) and Alan M. Turing (in [83]).

new theory was soon named *(the theory of)* NP-*completeness* for reasons that are explained in sources such as and developed in texts such as . Within a year, Richard M. Karp and colleagues had greatly lengthened the list of computational problems that Cook's nascent theory encompassed; several variants of the Hamiltonianicity-detection problem were on this list. The details of the theory of NP-completeness are beyond the scope of this text, but we do want the reader to recognize the following.

> *The problem of deciding, given a graph $\mathscr{G}$ that is presented via a list of nodes and a list of edges, whether $\mathscr{G}$ admits a Hamiltonian path or a Hamiltonian cycle is* NP-*complete.*

It remains mathematically *possible* that NP-complete problems can be solved efficiently—meaning "in time polynomial in the size of the description of the problem" (e.g., the number of nodes and edges in a graph). However it is mathematically *certain* that if any one problem in this class has a polynomial-time solution, then every problem in this class does.

Given the half-century that has passed since the work of Cook and Karp, and given the practical importance of many of the problems that are known to be NP-complete, it is widely *suspected* that no NP-complete problem can be solved via a polynomial-time algorithm.

There is a large, vibrant literature concerning approaches for lessening the negative computational impacts of NP-completeness—by considering significant special cases of NP-complete problems, by developing approximate solutions to such problems, and by developing heuristics fr such problems, whose behavior cannot be guaranteed but which seem to work well "in practice".

10.3 Graph Coloring and Chromatic number

This section introduces the notion of *graph coloring*, and its associated notion of the *chromatic number* of a graph.

A *node-coloring* of a graph $\mathscr{G}$ is an assignment of labels to $\mathscr{G}$'s nodes, in such a way that all of a node v's neighbors get a different label than v. Traditionally, the labels are called *colors*, for that term's evocative power. The *chromatic number* of a graph $\mathscr{G}$ is the smallest number of colors that can be used in a legal node-coloring of $\mathscr{G}$. In traditional parlance, the assertions

"$\mathscr{G}$ has chromatic number c" and "$\mathscr{G}$ is c-*colorable*"

are considered to be synonymous.

The notion of graph coloring can be used to computational advantage to model a broad variety of situations. An extremely important, and illustrative, use of graph coloring is to model *distributed* computing. In this setting, the nodes of a computation-graph $\mathscr{G}$ represent *agents*, such as, e.g., processing elements in a computer; and g's edges represent *communication links* that enable each node u to check its neighbors' states before any action and to inform its neighbors of state-changes occasioned by an action of u. The prohibition against "monochrome" edges—i.e.,

edges both of whose incident nodes have the same color—guarantees that node *u* and all of its like-colored nodes can act at the same instant with no fear of missing an important input to those actions. Indeed, one often encounters programs for distributed computing that look something like

1. All *red* nodes perform an action
2. All *green* nodes perform an action
3. All *blue* nodes perform an action

$\vdots$

10.3.1 Graphs with Small Chromatic Numbers

Most of the "named" graphs in Section 10.1.3 have very small chromatic numbers. This is no accident: These graphs were invented (or, at least, placed in the spotlight) because of their importance to the topic of parallel and distributed computing—and we suggested only a few paragraphs ago how to use graph node-colors to orchestrate some such computations. Accordingly, we devote this section to studying graphs with very small chromatic numbers.

10.3.1.1 Graphs with chromatic number 2

We begin to garner intuition about graph coloring by exposing a large set of graphs with chromatic number 2. In fact, we can completely characterize these graphs structurally.

A graph $\mathscr{G}$ is *leveled* if there exists an assignment of *level numbers* $\{1,2,\ldots,\lambda\}$ to the nodes of $\mathscr{G}$ in such a way that every neighbor of a level-ℓ node *u* resides either on level $\ell+1$ or on level $\ell-1$.

Proposition 10.15 *A graph $\mathscr{G}$ has chromatic number* 2 *if, and only if, it is leveled.*

Proof. Say first that $\mathscr{G}$ is a leveled graph. Then labeling each node of $\mathscr{G}$ with the (odd-even) parity of its level provides a valid 2-coloring of $\mathscr{G}$.

Say next that $\mathscr{G}$ is 2-colorable. Pick any node *v* of $\mathscr{G}$ and assign it to be the unique node on level 1. Let all neighbors of *v* be assigned to level 2. Continuing iteratively, say that the largest level-number that we have employed—i.e., assigned nodes to—is ℓ. Then we now assign to level $\ell+1$ all neighbors of level-ℓ nodes which have not yet been assigned to a level of $\mathscr{G}$. Because $\mathscr{G}$ is 2-colorable, each of the levels we have specified is monochromatic, so that each edge of $\mathscr{G}$ connects a node of one color with a node of the other color. □

We can now show that the following named graphs are 2-colorable.

Corollary 10.2 *The following graphs are leveled, hence have chromatic number* 2.
(a) *any tree (which includes any path-graph $\mathscr{P}_n$)*

(b) *any cycle-graph $\mathscr{C}_n$ that has an even number n of nodes*
(c) *any mesh-graph $\mathscr{M}_{m,n}$*
(d) *any torus-graph $\widetilde{\mathscr{M}}_{m,n}$ that has an even number of diagonals, i.e., even $m+n$*
(e) *any hypercube $\mathscr{Q}_n$*

Proof. We provide a detailed sketch for each of the five graph families in turn.
(a) We follow the procedure from the second half of the proof of Proposition 10.15 to expose a level structure in any tree $\mathscr{T}$, as follows. Pick any node v of $\mathscr{T}$ and make it the unique node on level 1. Let all neighbors of v be assigned to level 2. Continuing iteratively, say that the largest level-number that we have employed is ℓ. Then we now assign to level $\ell+1$ all neighbors of level-ℓ nodes which have not yet been assigned to a level of $\mathscr{T}$.

Of course this process can be simplified when $\mathscr{T}$ is a path-graph $\mathscr{P}$, by choosing one of $\mathscr{P}$'s end-nodes as node v. We thereby have precisely one node on each level, whereas the general procedure can have levels with 2 nodes. A lesson here is that *a graph may admit many distinct level structures*.

(b) When we apply the procedure of part (a) to an even-length cycle $\mathscr{C}_{2n}$, we produce a level structure in which levels 1 and $n+1$ have one node apiece, while all other levels have two nodes apiece.

(c) The edge-structure of mesh-graphs ensures that the labeling of each node $\langle i,j\rangle$ of $\mathscr{M}_{m,n}$ with the odd-even parity of the number $i+j$ is a 2-coloring of $\mathscr{M}_{m,n}$.

(d) The labeling in part (d) provides a 2-coloring of any torus-graph $\widetilde{\mathscr{M}}_{m,n}$ with even $m+n$.

(e) Each edge of a hypercube $\mathscr{Q}_n$ connects a node $v=\beta_1\beta_2\cdots\beta_n$, where each $\beta_i\in\{0,1\}$, to a node $v'=\beta_1'\beta_2'\cdots\beta_n'$ where precisely one β_j differs from β_j'. Therefore, the following aggregation of nodes of $\mathscr{Q}_n$ into sets $S_0,S_1,\ldots,S_n$ provides a valid leveling of $\mathscr{Q}_n$.

Assign node $v=\beta_1\beta_2\cdots\beta_n$ to set S_k precisely if k of the bits β_i equal 1. □

The reader can show that no odd-length cycle $\mathscr{C}_{2n+1}$ with $n\geq 1$ is 2-colorable. In like fashion, no graph $\mathscr{G}$ that *contains* an odd-length cycle can be 2-colored. This is the problem that plagues the torus $\widetilde{\mathscr{M}}_{m,n}$ when $m+n$ is odd. Put the odd cycle and torus as Exercises

The existence of odd-length cycles prevents *every* de Bruijn network $\mathscr{D}_n$ from being 2-colorable. As small examples, one observes the 3-cycle

$$00 \;\to\; 01 \;\to\; 10 \to\; 00$$

in $\mathscr{D}_2$ in Fig. 10.8 and the 3-cycle

$$001 \;\to\; 010 \;\to\; 100 \;\to\; 001$$

in $\mathscr{D}_3$ in Fig. 10.9. In fact, these small odd-length cycles are only the proverbial tip of the iceberg for de Bruijn networks. The following result asserts that de Bruijn

networks are *directed-pancyclic*, meaning that they contain (even directed!) cycles of *all* possible lengths, both odd and even. The proof of this result is outside the scope of this text, but it should be accessible to the motivated reader.

Proposition 10.16 *[85] For all n, the order-n de Bruijn network $\mathscr{D}_n$ is directed-pancylic, i.e., it contains directed cycles of all possible lengths* $1, 2, \ldots, 2^n$.

10.3.1.2 Planar and Outerplanar Graphs

In this section, we focus on two graph families that are defined in terms of the way they can be drawn (on a two-dimensional medium, such as a piece of paper).

> The reader should not view this attention to how a graph can be drawn as just an abstract game. The process of designing and implementing circuits within the constraints of *VLSI*, *Very Large Scale Integrated Circuit* technology, are very similar to drawing a circuit on a two-dimensional medium. We refer the reader to the revolutionary 1979 text [57] for an introduction to this fascinating technology, which requires technical literacy but little specialized knowledge.

A graph is *planar* precisely if it can be drawn *without any crossing edges*. A graph $\mathscr{G}$ is *outerplanar* precisely if it can drawn by *placing its nodes along a circle in such a way that its edges can be drawn as noncrossing chords of the circle*. The latter condition is equivalent to demanding that $\mathscr{G}$'s edges can be drawn within the circle without any crossings.

We urge the reader to garner intuition about the graphs in these families by experimenting with drawing some specific, rather complex graphs.

- The first set of graphs to draw are cliques, as defined in Section 10.1.3.2. The cliques $\mathscr{K}_3$, $\mathscr{K}_4$, and $\mathscr{K}_5$ will help expose the nature of the planar and outerplanar graphs, because:
  - $\mathscr{K}_3$ is outerplanar;
  - $\mathscr{K}_4$ is planar but not outerplanar;
  - $\mathscr{K}_5$ is not planar.
- The "bipartite" cousins of the cliques will also yield valuable insights. For postive integers m and n, the $m \times n$ bipartite clique $\mathscr{K}_{m,n}$ is the graph whose node-set comprises the ordered pairs of integers:

$$\mathscr{N}_{\mathscr{K}_{m,n}} = \{\langle i, j\rangle \mid 1 \le i \le m;\ 1 \le j \le n\}$$

  and whose edges connect each node $\langle i, j\rangle$ to every node $\langle i, k\rangle$ with $1 \le k \le n$ and to every node $\langle h, j\rangle$ with $1 \le h \le m$.
  The second set of graphs to draw are the bipartite cliques $\mathscr{K}_{1,3}$, $\mathscr{K}_{2,3}$, and $\mathscr{K}_{3,3}$. These graphs will also help expose the nature of the planar and outerplanar graphs, because:

- $\mathscr{K}_{1,3}$ is outerplanar;
- $\mathscr{K}_{2,3}$ is planar but not outerplanar;
- $\mathscr{K}_{3,3}$ is not planar.

We selected the preceding cliques and bipartite cliques to "play with" very carefully. Using arguments that go beyond the scope of this text, one can prove the following *characterization via exclusion* result, which characterizes each of our graph families by identifying *forbidden subgraphs*. The notion of *graph homeomorphism* plays a fundamental role in the characterization. This is a dauntingly named technical term that is easily understood informally. A *homeomorph* of a graph $\mathscr{G}$ is obtained by adding (degree-2) nodes along one or more edges of $\mathscr{G}$. The characterization of planar graphs by exclusion resides in a celebrated theorem by the Polish mathematician and logician Kazimierz Kuratowski; the analogous result for outerplanar graphs was derived by the French mathematician Gary Chartrand and the American mathematician Frank Harary.

Theorem 10.1. (a) [23] *A graph is outerplanar if, and only if, it does not have a subgraph that is homeomorphic to either $\mathscr{K}_4$ or $\mathscr{K}_{2,3}$.*

(b) [49] *A graph is planar if, and only if, it does not have a subgraph that is homeomorphic to either $\mathscr{K}_5$ or $\mathscr{K}_{3,3}$.*

A. Outerplanar graphs

We look first at the smaller of this section's graph families, namely, the *outerplanar graphs*. (We shall remark in subsection B why these graphs are called *outer*-planar.)

We begin our discussion of outerplanar graphs with some basic facts about the family.

Proposition 10.17 *Every tree is outerplanar.*

We leave to the reader the challenge of drawing a tree in a way that exposes its outerplanarity. Another exercise

Proposition 10.18 *Let $\mathscr{G}$ be an outerplanar graph. Then:*

(a) *$\mathscr{G}$ is planar.*

(b) *$\mathscr{G}$ is a subgraph of a Hamiltonian graph.*

(c) *Every subgraph of $\mathscr{G}$ is outerplanar.*

(d) *At least one of $\mathscr{G}$'s nodes has degree ≤ 2.*

Proof. **(a)** $\mathscr{G}$'s planarity can be inferred from the ability to draw $\mathscr{G}$'s edges as noncrossing chords of the circle.

(b) $\mathscr{G}$'s "sub-Hamiltonianicity" can be inferred from the ability to draw $\mathscr{G}$ with its nodes along a circle.

(c) We can produce an outerplanarity-witnessing drawing of any subgraph of $\mathscr{G}$ by erasing some nodes and/or some edges from our outerplanarity-witnessing drawing of $\mathscr{G}$.

(d) One verifies easily that this result holds for all outerplanar graphs having 3 or fewer nodes. Focus, therefore, on an arbitrary outerplanar graph $\mathscr{G}$ that has more than 3 nodes. Since adding more edges to a graph cannot decrease the degree of any node, we lose no generality by focusing on a $\mathscr{G}$ that is *maximal* outerplanar, in the sense that adding any new edge to $\mathscr{G}$ would destroy its outerplanarity.

Because $\mathscr{G}$ has more than 3 nodes, and because all of its nodes lie on a circle (in the drawing that witnesses its outerplanarity), there must be pairs of nodes of $\mathscr{G}$ that are not adjacent along the circle. Let u and v be nonadjacent nodes such that the distance between u and v (measured in edges that must be traversed to reach one from the other) is minimal among pairs of nodes nonadjacent nodes. We consider two cases.

- If the distance between u and v were 2, then the unique node that lies between u and v along the circle would have degree 2.
- If, on the other hand, the distance between u and v *exceeded* 2, then there would be at least *two* nodes that lie between u and v in either direction around the circle. But in this case, there would be two nonadjacent nodes that were closer to one another than u and v—which contradicts our choice of u and v as a pair of closest nonadjacent nodes.

We conclude that $\mathscr{G}$ must have a node of degree ≤ 2, completing the proof. □

Our primary concern in this section is, of course, on graph coloring. We return to this topic now. The 3-node cycle $\mathscr{C}_3$ witnesses the fact that not every outerplanar graph is 2-colorable; 3 colors is the best that we can hope for. We now show that this hope can be realized. The inductive proof of the following result can easily be turned into an efficient node-coloring algorithm for outerplanar graphs.

Proposition 10.19 (The 3-Color Theorem for Outerplanar Graphs) *Every outerplanar graph is* 3*-colorable.*

Proof. We proceed by induction on the number of nodes in the outerplanar graph to be colored.

The base cases of the result are provided by small outerplanar graphs—say those having 3 nodes or fewer.

We assume, for induction, that every outerplanar graph having fewer than n nodes is 3-colorable.

Let us now focus on an arbitrary n-node outerplanar graph $\mathscr{G}$. By Proposition 10.18(d), $\mathscr{G}$ has a node v of degree ≤ 2. Let us remove node v from $\mathscr{G}$, along with the edge(s) that connect v to the rest of $\mathscr{G}$; call the resulting graph $\mathscr{G}'$. Now, $\mathscr{G}'$ is clearly outerplanar, once we "stitch" together the circle we "damaged" by removing v, and $\mathscr{G}'$ has fewer than n nodes. By induction, therefore, $\mathscr{G}$ is 3-colorable. But now we can reattach node v to $\mathscr{G}'$ by replacing the edges that attach v to $\mathscr{G}$. Moreover, we can now color v using whichever of the 3 colors on $\mathscr{G}$ that is *not* used for v's neighbors in $\mathscr{G}$. We have, thus, specified a 3-coloring of $\mathscr{G}$, which extends the induction and completes the proof. □

B. Planar graphs

The larger of this section's two graph families comprises *planar graphs* i.e., graphs that can be drawn (on a two-dimensional medium, such as a piece of paper) with no crossing edges. The original focus on planar graphs stemmed from viewing them as abstractions of geographical maps.

The 4-node clique $\mathscr{K}_4$ witnesses the fact that not every planar graph is 3-colorable; 4 colors is the best that we can hope for. A century-plus attempt to prove that 4 colors suffice for planar graphs culminated in one of the most fascinating dramas in modern mathematics, as American mathematicians Kenneth Appel and Wolfgang Haken, —with the help of their families and their computer!—announced their two-article-long proof in 1974 of their renowned 4-*Color Theorem for Planar Graphs*.

Theorem 10.2 (The 4-Color Theorem for Planar Graphs [4, 5]). *Every planar graph is* 4*-colorable.*

Beginning with a failed attempt, in 1875, to prove that every planar map can be colored with four colors with no abutting countries getting the same color, the so-called 4-*Color Problem* held the world of discrete mathematics in thrall for roughly a century before But the drama surrounding the 4-Color Problem persisted, because of the Appel-Haken proof's reliance, in a fundamental way, on a compute program that checked more than a thousand essential assertions (about forbidden subgraphs). It took the mathematics community years before this proof, with its massive complexity and unprecedented employment of "collaboration" by computer, was generally accepted. In addition to the primary references [4, 5] that accompany our statement of the Theorem, we recommend to the interested reader the articles [6, 7] that summarize and, to some extent, popularize this marvelous mathematical tale.

Not surprisingly, we are not going to present a proof of Theorem 10.2 in an introductory text, but we shall now present the six-color and five-color analogues of the theorem. We present the weaker result (six colors) as well as the stronger one (five colors) because the six-color result is so close to its outerplanar-graph cousin, Proposition 10.19—a fact that we hope the reader will find thought-provoking.

The first step in proving the "6-Color Theorem for Planar Graphs" will be to derive the following analogue for planar graphs of Proposition 10.18(d), which asserts that every outerplanar graph has a node of degree 2.

Lemma 10.1. *Every planar graph has a node of degree* ≤ 5.

Proof. (Lemma 10.1) Let us focus on a planar drawing of a (perforce) planar graph $\mathscr{G}$, which has n nodes, e edges, and f *faces*. A *face* in a drawing of $\mathscr{G}$ is a polygon whose sides are edges of $\mathscr{G}$, whose points are nodes of $\mathscr{G}$, and whose interiors are empty, in that no edge of $\mathscr{G}$ crosses through the interior.

Now that we know about faces, we can finally describe the origin of the term *outerplanar*. A graph $\mathscr{G}$ is outerplanar if it can be drawn with all of its nodes around a single "outer" face in such a way that all of its edges are drawn in a noncrossing manner within the "outer" face. In fact, we usually draw $\mathscr{G}$ so that the "outer" face is literally outside the "outer" face.

The following auxiliary result derives a celebrated "formula" of Euler.

Proposition 10.20 (Euler's Formula for Planar Graphs) *Given the indicated drawing of $\mathscr{G}$, we have*

$$n - e + f = 2 \tag{10.9}$$

Proof (Proposition 10.20). We proceed by structural induction, growing the graph $\mathscr{G}$ edge by edge.

The result clearly holds for the smallest planar graphs, including the smallest interesting one, namely, $\mathscr{C}_3$, which has $n = e = 3$ and $f = 2$ (the inner face of the triangle and the outer face).

Say that we have a current version of $\mathscr{G}$, and we proceed to grow it by adding a new edge. Two cases arise.

- The new edge connects existing nodes. In this case, the augmentation of $\mathscr{G}$ increases the number of edges (e) and the number of faces (f) by 1 each, while keeping the number of nodes (n) unchanged. Euler's formula (10.9) thus continues to hold.
- The new edge adds a new node, which is appended to some preexisting node. In this case, the augmentation of $\mathscr{G}$ increases the number of edges (e) and the number of nodes (n) by 1 each, while keeping the number of faces (f) unchanged. Euler's formula (10.9) thus continues to hold.

This augmentation extends the induction, whence the result. □-Proposition 10.20

As we approach the next step of the proof, we simplify the setting by assuming henceforth that $\mathscr{G}$ is connected and that it is *maximal*, in the sense that one cannot add any new edge to the drawing without crossing an existing edge. This step only strengthen's the Lemma's conclusion by apparently making it more difficult to find a small-degree node.

With this assumption in place, we now adapt a pedagaogical tool from [11], in order to make the following counting argument easier to follow. We construct a *directed bipartite* graph $\mathbf{G}$ which exposes certain features of $\mathscr{G}$'s structure. On one side of $\mathbf{G}$ are the f faces of $\mathscr{G}$; on the other side are $\mathscr{G}$'s e edges. $\mathbf{G}$ contains an arc from each face φ of $\mathscr{G}$ to each edge ε of $\mathscr{G}$ that forms a "side" of the polygonal drawing of φ. Because $\mathscr{G}$ is a maximal planar graph, we have:

- Each face of $\mathscr{G}$ is a 3-cycle, hence involves three nodes.
- Each edge of $\mathscr{G}$ touches two faces.
- Each edge of $\mathscr{G}$ touches two nodes.

Let us now put these facts together, and assume, for contradiction, that every node of $\mathcal{G}$ had degree ≥ 6. We would then find that

$$\left[f \ \leq \ \frac{2}{3}e \right] \quad \text{and} \quad \left[e \ \geq \ 3n \right]$$

Incorporating these two bounds into Euler's formula (10.9), we arrive at the following contradiction.

$$2 \ = \ n - e + f \ \leq \ \frac{1}{3}e - e + \frac{2}{3}e \ = \ 0$$

This contradiction proves that every planar graph must have a node of degree ≤ 5. □-Lemma 10.1

We finally have the tools to color planar graphs using 6 colors.

Proposition 10.21 (The 6-Color Theorem for Planar Graphs) *Every planar graph is 6-colorable.*

Proof. The 2-Color Theorem for Outerplanar Graphs (Proposition 10.19) and this result follows via almost-identical inductions on the number of nodes in the graph $\mathcal{G}$ that is being colored. Both arguments:

1. remark that the coloring goal can be met for small graphs
   For outerplanar graphs, "small" means "3 or fewer nodes". For planar graphs, it means "4 or fewer nodes".
2. remove from $\mathcal{G}$ a node v of smallest degree d_v, together with all its incident edges
   For outerplanar graphs, we guarantee that $d_v \leq 2$ (Proposition 10.18(d)). For planar graphs, we guarantee that$d_v \leq 5$ (Lemma 10.1).
3. inductively color the nodes of the graph $\mathcal{G}'$ left after the removal of v
   For outerplanar graphs, we color $\mathcal{G}'$ with ≤ 3 colors (Proposition 10.19). For planar graphs, we use an inductive assumption that $\mathcal{G}'$ can be colored with ≤ 6 colors.
4. reattach v via its d_v edges and then color v.
   Note that the coloring guarantee in both results—Proposition 10.19 for outerplanar graphs and the current result for planar graphs—allows us to use $d_v + 1$ colors to color $\mathcal{G}$. Because v has degree d_v, it is a neighbor of no more than d_v nodes of $\mathcal{G}'$, so our access to $d_v + 1$ colors guarantees that we can successfully color v.

The proofs of the 3-colorability of outerplanar graphs and the 6-colorability of planar graphs thus differ only in the value of d_v. □

**HERE
We now show how to color planar graphs using 5 colors.
**FIND REFERENCE

Proposition 10.22 (The 5-Color Theorem for Planar Graphs) *Every planar graph is 5-colorable.*

Proof. This completes the proof. □

10.3.2 Computing the Chromatic Number of an Arbitrary Graph

**FIND REFERENCES

It is an NP-complete problem to decide, given any fixed $k \geq 3$, whether a given graph $\mathscr{G}$ is k-colorable.

A simple argument shows that it is NP-hard to find smallest number of colors that provide a valid node-coloring of $\mathscr{G}$.

But one can exhibit "greedy" algorithms that give good results.

10.4 ⊕ Pointers to Advanced Topics

**PROVIDE SOME REFERENCES

We conclude this chapter by mentioning a variety of topics that are typically not covered—at least in depth—early in the curriculum, but that are important enough that the reader should at least be aware of them. The topics we mention are motivated by virtually every computational area that benefits from graph-theoretic models. We have tried to present each topic we touch on at a level of discourse that will prepare the interested reader to delve more deeply into the material, yet at a level of informality that will make the material accessible to the more casual reader. We thus strive for an intuitive presentation that will not lead any reader astray.

The two problems we discuss in Section 10.4.1 illustrate how the *dynamic* models in the field of algorithmics—"dynamic" in the sense that they *do* something—and the *structural* models provided by graph theory can often provide beneficial illumination of one another.

Section 10.4.2 focuses on the myriad computations on graph that can be accomplished efficiently via recursive algorithms that decompose, then reassemble, the graphs that they work on.

Section 10.4.3 introduces the increasingly important topic of graphs whose structure changes dynamically over time. One timely instance of this dynamic evolution is the connectivity graph of the Internet.

10.4.1 Relating Computational and Mathematical Problems

This

10.4.1.1 The Route Inspection/ Chinese Postman Problem

**HERE

After solving the preceding "pure" version of the Eulerian-tour problem—which seeks a tour of a graph which crosses each edge precisely once—we discuss an extension of this problem which allows us to augment a graph $\mathscr{G}$ by adding multiple edges between the same two nodes. (Terminologically, we thereby convert $\mathscr{G}$ to a *multi-graph* consisting of nodes and *multi-edges*.) Not obviously, adding multi-edges can often convert a graph $\mathscr{G}$ that does not admit an Eulerian cycle into a multi-graph that does admit an Eulerian cycle. The problem of finding the *smallest* such augmentation of $\mathscr{G}$—i.e., of adding the fewest multi-edges—is called the *Route Inspection Problem*; it is also often called the *Chinese Postman Problem*, in honor of its inventor, the Chinese mathematician Kwan Mei-Ko [50].

This section is devoted to studying the *Route Inspection Problem*, also known as the *Chinese Postman Problem*, in honor of its inventor, the Chinese mathematician Kwan Mei-Ko [50]. This problem seeks to add as few multi-edges as possible to a graph $\mathscr{G}$ in order to render $\mathscr{G}$ Eulerian. (A *multi-graph* is "almost" an undirected graph. It differs from a true graph because of the possible presence of multiple multi-edges that connect the same two nodes.)

**HERE

We know from Proposition 10.11 that if all of $\mathscr{G}$'s nodes have even node-degrees, then—*and only then*—$\mathscr{G}$ admits an Eulerian cycle. Therefore, in this case, *zero* multi-edges need be added to $\mathscr{G}$ to render it Eulerian.

Let us now present the more general problem of determining a cycle that contains all the edges in any graph, in particular when there exist some odd vertices. From the previous section, we know that there is no Eulerian cycle in this case and thus, any feasible solution should duplicate some edges. The problem is to duplicate the minimum. This problem is known as the *chinese postman* and it is described below (in a french equivalent version).

A postman moved recently from Grenoble to a small village in the country side. He asked himself how to organize his daily tour by bike for distributing the letters in the shortest possible time. The director of the post office gives him the map and fortunately, the postman had some old souvenir of previous lectures in Graph Theory. The tour starts from the post office and of course, the postman has to go through every roads for distributing the letters before coming back to his office. The underlying graph is $G = (V, E)$ where V is the (finite) set of cross points and E is the set of the links between the cross roads weighted by the distances.

Fig. 10.11 presents an example of the chinese postman problem.

This problem can be formulated mathematically in term of Eulerian cycles. Intuitively, the basic idea is to duplicate some edges that are carefully chosen in order to use the previous construction of an Eulerian tour of Section 10.2.1 that will help the postman to determine the optimal tour (of minimal length) using some simple mathematical properties.

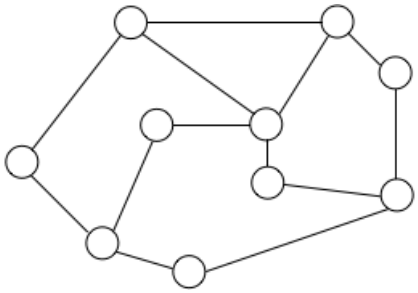


Fig. 10.11 An instance of the Chinese postman with 10 cross-nodes.

First, we know that there is an even number of odd vertices. Considering the previous instance of the postman problem, there are 4 such vertices (represented in grey in Fig. 10.12).

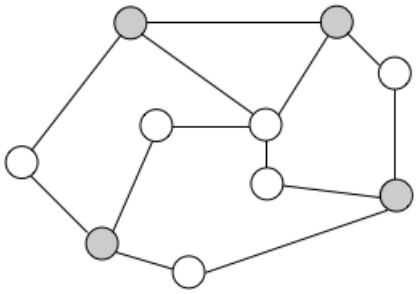


Fig. 10.12 The 4 vertices with an odd degree in the previous instance.

As there exists a path between any pair of vertices of odd degree in V_{odd}, we consider the complete graph whose vertices are the odd degree vertices weighting the edges with the shortest paths (denoted by K_{odd}). As we mentioned in the preliminary properties, computing the shortest paths is a classical problem, which can be solved in polynomial time.

Then, it is possible to do the correspondence between the optimal solution of the postman problem and a perfect matching of minimal weight in K_{odd} by duplicating the edges of the minimal perfect matching.

The main steps of the algorithm for determining the optimal tour are the following:

- Consider the complete graph with the odd vertices and compute its weight by the shortest paths. Compute a perfect matching of minimal weight between these vertices.

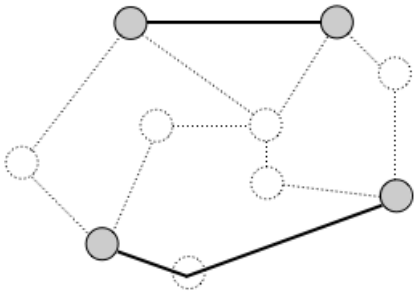


Fig. 10.13 The minimum weight perfect matching labelled by the shortest distances between the vertices of V_{odd}.

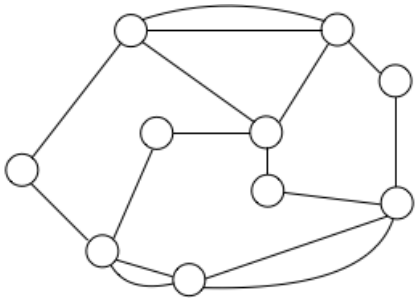


Fig. 10.14 Final step: adding the edges of the minimum perfect matching.

- Duplicate all the edges along the paths of this matching.
- Determine an Eulerian tour in this new graph with even degrees.

The optimality of this algorithm comes from the fact that the duplicated edges are the minimum possible ones. Finally, all the vertices of the new graph are even since the degree of the odd vertices in G is augmented by 1 (extremities of the paths) and the other even vertices which are intermediate vertices of the paths remain even.

This short discussion is a good segué to the material in the next section, which leds valuable perspective on our brief study of Hamiltonian paths and cycles—and, indeed, on the computational implications of that work.

10.4.1.2 Hamiltonianicity in weighted graphs and the Traveling Salesman Problem

In Section **??**.C, we suggested how daunting it is computationally to determine whether an unweighted graph admits a Hamiltonian cycle. There is an important, practically significant, analogue of this problem for edge-weighted graphs. This analogue has the traditional familiar name, the *Traveling Salesman Problem*, of-

ten abbreviated *TSP*. The TSP is a classical problem in Operations Research. Its familiar name arises from the following story.

We consider a saleswoman who wants to make a call on all of her n clients, who live in the n cities, $C_1, C_2, \ldots, C_n$. In order to minimize the *cost* of her tour, she studies the $\binom{n}{2}$ real numbers $\{c_{i,j} \mid 1 \leq i, j \leq n\}$, where each $c_{i,j}$ is the *cost* of traveling between cities C_i and C_j (in either direction).

> We are purposely vague about the meaning of the word "cost" here. The costs represented by the unknowns $c_{i,j}$ could be intercity driving distances or intercity travel times or intercity airfares. Instances of the TSP can be formulated with *any* notion of cost that can be represented by positive real numbers.
> The importance of being vague is that, in particular, *intercity costs are not assumed to obey any of the laws that one commonly associates with the distances we encounter in our daily lives*. The *triangle inequality* is the prime example of such a law. Intuitively, this law insists that the distance between any two cities is never decreased by placing an intermediate stopover city between them. In a common formulation: *A straight line is the shortest distance between two points.* Cost measures that obey the triangle inequality are termed *Euclidean* because the distances studied in Euclidean geometry are assumed to obey this law. (Some people like to think of intercity costs as airfares—which clearly have no logical basis.)

The saleswoman's objective is to schedule the order of visiting her clients' n cities in the most economical way. Formally, the challenge of the TSP is to discover a minimum-cost tour of all n cities. Such a tour would be a cycle of the form

$$C_{i_1} - C_{i_2} - \cdots - C_{i_n} - C_{i_1}$$

such that:

- all n cities appear in the tour precisely once;
- no tour has cost smaller than the cost

  $$c_{i_1,i_2} + c_{i_2,i_3} + \cdots + c_{i_{n-1},i_n} + c_{i_n,i_1}$$

  of the indicated tour.

The main connection of the TSP to this chapter is twofold.

1. The TSP admits a graceful representation as a weighted analogue of the problem of detecting Hamiltonian cycles. Within the representation, the TSP's n cities are depicted as an instance of the complete graph $\mathscr{K}_n$, and the TSP's intercity costs are depicted as real weights on the edges of $\mathscr{K}_n$. We describe this representation via the following informal, but precise, encoding of a graph $\mathscr{G}$ as an instance of the TSP.

   - Each node of $\mathscr{G}$ becomes a "city" that must be visited.

- For each pair of cities/nodes A and B:

$$\text{DISTANCE}(A,B) \;=\; \begin{cases} 1 & \text{if there is an edge between } A \text{ and } B \text{ in } \mathscr{G} \\ \infty & \text{if there is no edge between } A \text{ and } B \text{ in } \mathscr{G} \\ & \text{(if the idea of "infinite" distance bothers you,} \\ & \text{then make this some ridiculousy large number)} \end{cases}$$

2. The TSP is computationally "no easier" than the Hamiltonianicity-detection problem, in a very strong sense. By definition, the target of the TSP is a tour of minimum *overall cost*, as measured by the sum of the costs of the edges traversed in the tour. But, of course, *every* tour has some cost, according to this measure. The computational difficulty of the TSP is suggested by the following result, which we state without proof.

Proposition 10.23 *For any fixed positive constant κ, the problem of finding a tour for an instance of the TSP that is within a factor of κ of minimal is an* NP*-complete problem.*

The fact that TSP is a computationally complex problem is not very surprising, because the TSP can be framed in a way that encompasses a form of the Hamiltonianicity-detection problem. What *is* surprising, though, is that if we restrict attention to *Euclidean* instances of the TSP—i.e., instances whose "costs" measure Euclidean distances—say, driving distances—then there exists a rather efficient algorithm that solves such instances of the TSP *approximately* optimally, in the sense of Proposition 10.23. That is, there exists a fixed constant $\kappa > 0$ such that, for each Euclidean instance $\mathscr{I}$ of the TSP, if an *optimal*—i.e., *cost-minimal*—tour for instance $\mathscr{I}$ has cost $c^\star$, then the cost of the tour discovered by the algorithm is no larger than $\kappa c^\star$.

Proposition 10.24 *[26] There exists an algorithm for the Euclidean TSP that produces approximately optimal tours, specifically, tours whose costs are no greater than $\frac{3}{2}$ times the cost of an optimal tour.*

Proof. We construct an efficient solution for the Euclidean TSP, which is known as *the Christofides algorithm*, after its inventor, Nicos Christofides. [26]

The following table, and Fig. 10.18, depict an instance of the Euclidian TSP with $n = 7$ cities.

City	**Inter-City Costs**						
C_1	0	$c_{1,2}$	$c_{1,3}$	$c_{1,4}$	$c_{1,5}$	$c_{1,6}$	$c_{1,7}$
C_2	$c_{2,1}$	0	$c_{2,3}$	$c_{2,4}$	$c_{2,5}$	$c_{2,6}$	$c_{2,7}$
C_3	$c_{3,1}$	$c_{3,2}$	0	$c_{3,4}$	$c_{3,5}$	$c_{3,6}$	$c_{3,7}$
C_4	$c_{4,1}$	$c_{4,2}$	$c_{4,3}$	0	$c_{4,5}$	$c_{4,6}$	$c_{4,7}$
C_5	$c_{5,1}$	$c_{5,2}$	$c_{5,3}$	$c_{5,4}$	0	$c_{5,6}$	$c_{5,7}$
C_6	$c_{6,1}$	$c_{6,2}$	$c_{6,3}$	$c_{6,4}$	$c_{6,5}$	0	$c_{6,7}$
C_7	$c_{7,1}$	$c_{7,2}$	$c_{7,3}$	$c_{7,4}$	$c_{7,5}$	$c_{7,6}$	0

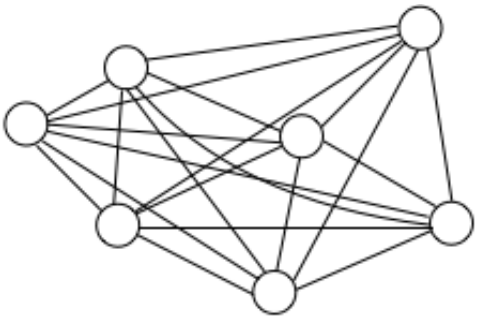


Fig. 10.15 An instance of the Euclidean TSP with 7 cities, $C_1, C_2, C_3, C_4, C_5, C_6, C_7$, and intercity costs $\{c_{i,j} \mid 1 \leq i, j \leq 7\}$.

Let us construct a good solution (not *too far* from the optimal) in polynomial time. Let us denote by ω_G the weight of graph G (i.e. the sum of the weights on its edges). The Chritofides algorithm proceeds in three steps.

Step 1. Determine a minimal weight spanning tree T^*. As we recalled in the preliminaries, a minimal weight spanning tree can be determined in polynomial time.

ω_{T^*} is a lower bound of the value of the optimal tour ω_{H^*}. Indeed, H^* is a cycle, then, removing any edge in H^* leads to a chain, which is a particular spanning tree. As T^* is the minimal spanning tree, we have: $\omega_{T^*} \leq \omega_{H^*}$.

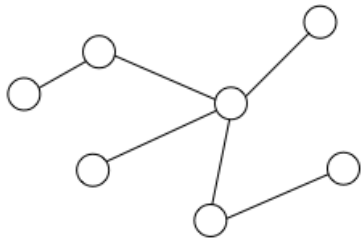


Fig. 10.16 Construction of an optimal spanning Tree T^*.

Step 2. Consider now the set V_{odd} of the vertices of T^* whose degrees are odd.

We proved in the preliminary properties that the cardinality of V_{odd} is even.

Let us construct the perfect matching C^* of minimum weight between the vertices in V_{odd}. Fig. 10.4 shows all possible perfect matchings on the previous example, the optimal one (with minimal weight) is represented in bold.

Fig. 10.17 illustrates the graph obtained by considering the edges of both T^* and C^*.

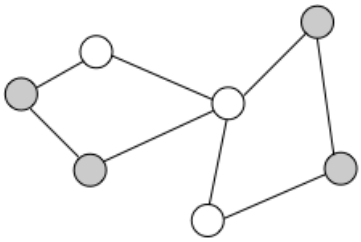


Fig. 10.17 Adding the optimal perfect matching C^* to the minimal spanning tree T^*.

Let us now determine a lower bound of the optimal tour H^* (represented in Fig. 10.18).

$2\omega_{C^*}$ is a lower bound of the value of the optimal tour ($\omega_{C^*} \leq \frac{1}{2}\omega_{H^*}$). Indeed, consider first the perfect matching C^*. As its vertices belong to H^*, ω_{C_*} is lower than the piece of Hamiltonian tour contained between these vertices because of the euclidian property (see Fig. 10.19). Similarly for the *complementary* perfect matching C (Fig. 10.20). Thus, the weight of the cycle formed by the concatenation of both perfect matchings is lower than the Hamiltonian tour $\omega_{C^* \cup C} \leq \omega_{H^*}$. Moreover, as C^* is the minimum perfect matching, we have $\omega_{C^*} \leq \omega_C$, this concludes the proof.

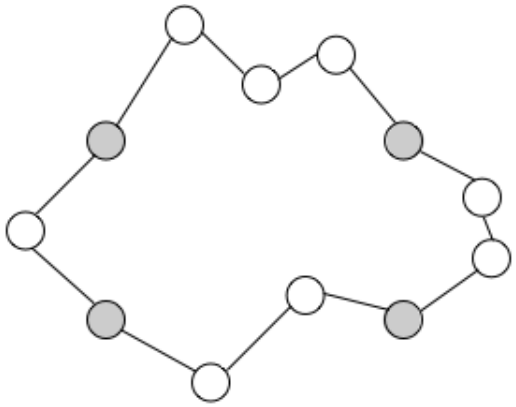


Fig. 10.18 An optimal Hamiltonian cycle H^*.

Step 3. All the vertices of $T^* \cup C^*$ have an even degree since we added an edge of C^* to every odd degree vertices of T^*. We are now going to transform this graph by replacing iteratively the high degree vertices by shortcuts, which decreases the degree until reaching 2.

While it exists a vertex of degree greater than 4, we remove two of these consecutive edges and replace them by the opposite edge of this triangle without dis-

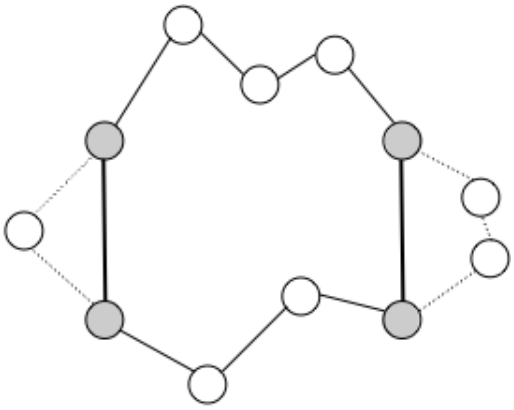


Fig. 10.19 Perfect matching C^* between the vertices of odd degrees.

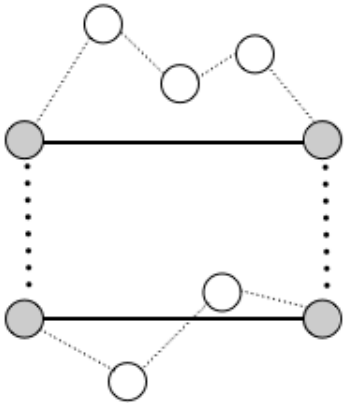


Fig. 10.20 Cycle $C^* \bigcup C$ (in dashed and bold).

connecting the graph. There are $2k$ ways to remove 2 edges and replace them by the triangle edge. Some of them disconnect the graph and thus, must be avoided. Fig. 10.21 shows such a transformation on the previous example, Fig. 10.22 shows a valid transformation.

This process leads to a feasible tour. Such transformations do not increase the total weight.

Finally, as $\omega_{T^*} \leq \omega_{H^*}$ and $\omega_{C^*} \leq 1/2\omega_{H^*}$, we deduce that the value of such a tour is lower than $3/2\omega_{H^*}$. □

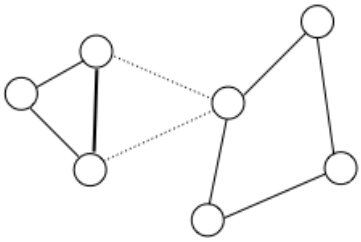


Fig. 10.21 Reduction of the degree in $T^* \bigcup C^*$, disconnected solution.

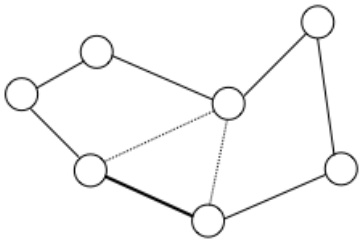


Fig. 10.22 Reduction of the degree in $T^* \bigcup C^*$, connected solution.

10.4.2 Graph Decomposition

A fundamental variety of relevant notions within the study of graphs reside in the notions known in various contexts via terms such as *graph separators* or *graph bisection*. The key idea that underlies these notions is that certain graph-theoretic structures interconnect their graphs' constituent nodes more or less densely — and the type and level of interconnectivity has important algorithmic consequences. In such situations, the student must understand how the phenomenon/a modeled by the graphs of interest are elucidated by the way a graph can be broken into subgraphs by excising nodes or edges. When discussing communication-related structures, for instance, graph are often used to model the individual pairwise communications that must occur in order to accomplish the desired overall communication (such as a broadcast). There is often a provable tradeoff between the number of such pairwise communications and the overall time for the completion of the overall task. As another example, when discussing social networks, one can pose questions such as: which node in a network is best to connect to (when joining the network) in order to best facilitate one's interactions or influence within the community. The latter topic leads, e.g., to the study of *power-law* networks, a topic that would not be studied in depth in any early course; indeed, the structure of these networks is not yet well understood even in advanced settings.

[71]

10.4.3 Graphs with Evolving Structure

Classical problems in the area of graph algorithms will discuss graphs, especially trees, whose structures evolve over time. Such evolution is observed, e.g., in the study of "classical" algorithmic problems such as *Minimum Spanning Tree* and *Branch and Bound*; see, e.g., [28]. What is certain to be more exciting to the reader, though, are the "modern" topics where one encounters graphs with evolving structure, such as *social networks* and *inter-networks* (e.g., the *Internet of Things*).

For "classical" topics, as exemplified by the two we have mentioned, the mathematics covered in this chapter will provide the reader with the background necessary to deal with graph evolution. Indeed, this evolution emerges as an inevitable concomitant of the algorithmics that is superimposed upon the traditional structures of graph theory: the challenge to the reader is to assimilate new algorithmic notions, not new mathematics.

In contrast, the "modern" topics we have mentioned do require the reader's assimilating new mathematics. Dealing successfully with the algorithmic issues that arise with social networks and inter-networks requires the reader to understand the structures of the evolving graph-oriented systems and how evolution changes these structures. A number of competing, rather sophisticated, abstract models have been developed—see, e.g., [1, 8, 16, 25]—and numerous studies have attempted to understand the specific situations wherein the abstract models reflect reality more or less faithfully—see, e.g., [18, 33, 42, 82, 86].

10.4.4 Hypergraphs

A large variety of modern computing-related topics benefit from the structure inherent in graph-theoretic models but do not comfortably conform to the *binary* relationships imposed by graphs' having *two* nodes per edge. A model that retains the structure of graph-theoretic models without the binary constraint is the generalization of graphs called *hypergraphs*. A hypergraph has nodes that play exactly the same role as with graphs, but in place of a graph's binary edges, a hypergraph has *hyperedges*, each being a set of nodes whose size is not restricted to 2. A rather general treatment of hypergraphs can be found in the comprehensive graph-theory text [11]; a specialized article that focuses on some of the topics of this chapter, such as node-coloring, is [56]. Because of their inherent complexity, hypergraphs as graph-theoretic objects are usually relegated to advanced courses. However, the literature contains many studies of hypergraphs that are "fine-tuned" for specific computing-related application areas, and many of these should be accessible without extensive mathematical background. Sample computing-related application areas that benefit from hypergraph-oriented models include the following.

- Bus-connected parallel communication has been part of digital computer design since its earliest days. The informal picture of such a system is that there are

communication channels that multiple agents can retrieve message from and post messages to. In hypergraph-oriented terms: the nodes/communicating agents aggregate into groups/hyperedges. Each group's agents share "read/write" access to a specific channel. A spcialized genre of hypergraph that was invented to study the described scenario is the *interval hypergraph* model developed in [67].

- Modern electronic circuits are implemented using integrated circuit technology, specifically, *VLSI: Very Large Scale Integrated circuitry*; see, e.g., [57]. These technologies tend to be voltage-driven, rather than current-driven. Accordingly, much of the attention when designing circuits centers on the coordination of equi-potential points in a network, rather than on point-to-point transmission of signals. Hypercubes are tailor-made for such technologies. A crucially important issue that arises because of the design strengths and weaknesses of VLSI technology is *fault tolerance*—how to cope with the inevitable faulty transistors in massive VLSI systems. Even mathematically quite-accessible ideas can provide provocative ideas about this important topics; see, e.g., [66]
- Social networks have become so prevalent in society that no one will be surprised to learn that many approaches to modeling the networks' interconnectivity have been studied. In Section 10.4.3, we discussed an interconnectivity model based on evolving graphs and clustering within such graphs. More recently, hypergraph-based models have also been proposed; see, e.g., [2, 55].

References

1. W. Aiello, F.R.K. Chung, L. Lu (2000): A random graph model for massive graphs. *32nd Ann. Symp. on the Theory of Computing*.
2. F. Amato, F. di Lillo, V. Moscato, A. Picariello, G. Sperl (2017): Influence analysis in online social networks using hypergraphs. *IEEE Int'l Conf. on Information Reuse and Integration*. DOI: 10.1109/IRI.2017.72
3. F. Annexstein, M. Baumslag, A.L. Rosenberg (1990): Group action graphs and parallel architectures. *SIAM J. Comput. 19*, 544–569.
4. K. Appel, W. Haken (1977): Every planar map is four colorable, I: Discharging. *Illinois J. Mathematics, 21*(3), 429-490.
5. K. Appel, W. Haken (1977): Every planar map is four colorable, II: Reducibility. *Illinois J. Mathematics, 21*(3), 491-567.
6. K. Appel, W. Haken (October 1977): Solution of the four color map problem. *Scientific American, 237*(4), 108-121.
7. K. Appel, W. Haken [with the collaboration of J. Koch] (1989): Every planar map is four-colorable. *Contemporary Mathematics, 98*, American Mathematical Society, Providence, RI.
8. A.L. Barabási, R. Albert (1999): Emergence of scaling in random networks. *Science (286)*, 509–512.
9. S.L. Basin (1963): The Fibonacci Sequence as it appears in nature. *Fibonacci Quart. 1*, 53–57.
10. E.T. Bell (1986): *Men of Mathematics*. Simon and Schuster, New York.
11. C. Berge (1973): *Graphs and Hypergraphs*. North-Holland, Amsterdam.
12. J.-C. Bermond, C. Peyrat (1989): The de Bruijn and Kautz networks: a competitor for the hypercube? In *Hypercube and Distributed Computers* (F. Andre and J.P. Verjus, eds.) North-Holland, Amsterdam, 279–293.
13. F. Bernstein (1905): Untersuchungen aus der Mengenlehre. *Math. Ann. 61*, 117–155.
14. G. Birkhoff and S. Mac Lane (1953): *A Survey of Modern Algebra*, Macmillan, New York.
15. E. Bishop (1967): *Foundations of Constructive Analysis*, McGraw Hill, New York.
16. B. Bollobas (1985): *Random Graphs*. Academic Press, N.Y.
17. W.W. Boone, F.B. Cannonito, R.C. Lyndon (1973): *Word Problems: Decision Problem in Group Theory*, North-Holland, Amsterdam.
18. T. Bu, D. Towsley (2002): On distiniguishing between internet power-law generators. *IEEE INFOCOM'02*.
19. G. Cantor (1874): Über eine Eigenschaft des Inbegriffes aller reellen algebraischen Zahlen. *J. Reine und Angew. Math. 77*, 258–262.
20. G. Cantor (1878): Ein Beitrag zur Begründung der transfiniter Mengenlehre. *J. Reine Angew. Math. 84*, 242–258.
21. G. Cantor (1887): Mitteilungen zur Lehre vom Transfiniten. *Zeitschrift fü Philosophie und Philosophische Kritik 91* 81-125.

22. A.L. Cauchy (1821): *Cours d'analyse de l'École Royale Polytechnique, 1ère partie: Analyse algébrique*. l'Imprimerie Royale, Paris. Reprinted: Wissenschaftliche Buchgesellschaft, Darmstadt, 1968.
23. G. Chartrand, F. Harary (1967): Planar permutation graphs. *Annales de l'Institut Henri Poincaré B, 3*(4), 433-438.
24. G. Chartrand, S.F. Kapoor (1969): The cube of every connected graph is 1-hamiltonian. *J. Research of the National Bureau of Standards, 73B(1)*. DOI: 10.6028/jres.073B.007
25. Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, W. Willinger (2002): The origin of power laws in internet topologies revisited. *IEEE INFOCOM'02*.
26. N. Christofides (1976): Worst-case analysis of a new heuristic for the travelling salesman problem. Report 388, Graduate School of Industrial Administration, Carnegie-Mellon Univ.
27. S.A. Cook (1971): The complexity of theorem-proving procedures. *ACM Symp. on Theory of Computing*, 151–158.
28. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein (2001): *Introduction to Algorithms (2nd ed.)*. MIT Press, Cambridge, MA.
29. H.B. Curry (1934): Some properties of equality and implication in combinatory logic. *Annals of Mathematics, 35*, 849–850.
30. H.B. Curry, R. Feys, W. Craig (1958): *Combinatory Logic. Studies in logic and the foundations of mathematics*. North-Holland, Amsterdam.
31. M. Davis (1958): *Computability and Unsolvability.* McGraw-Hill, New York.
32. O. Deiser (2010): Einführung in die Mengenlehre Die Mengenlehre Georg Cantors und ihre Axiomatisierung durch Ernst Zermelo (3rd, corrected ed.), Berlin/Heidelberg: Springer, pp. 71, 501, doi:10.1007/978-3-642-01445-1, ISBN 978-3-642-01444-4.
33. M. Faloutsos, P. Faloutsos, C. Faloutsos (1999): On power-law relationships of the internet topology. *ACM SIGCOMM'99*.
34. H. Fleischner (1974): The square of every two-connected graph is hamiltonian. *J. Combinatorics Theory (B) 16*, 29–34.
35. G. Fubini (1907): Sugli integrali multipli. *Rom. Acc. L. Rend. (5), 16(1)*, pp. 608614. In *zbMATH 38.0343.02*. Reprinted in G. Fubini (1958): *Opere scelte, 2*, Cremonese, pp. 243249.
36. R. Fueter and G. Pólya (1923): Rationale Abzählung der Gitterpunkte. *Vierteljschr. Naturforsch. Ges. Zürich 58*, 380–386.
37. M.R. Garey and D.S. Johnson (1979): *Computers and Intractability*. W.H. Freeman and Co., San Francisco.
38. K. Gödel (1931): Über Formal Unentscheidbare Sätze der Principia Mathematica und Verwandter Systeme, I. *Monatshefte für Mathematik u. Physik 38*, 173–198.
39. P.R. Halmos (1960): *Naive Set Theory*. D. Van Nostrand, New York.
40. M. Hazewinkel, ed. (2001): Viète theorem. In *Encyclopedia of Mathematics*. Springer Science+Business Media B.V./Kluwer Academic Publishers, ISBN 978-1-55608-010-4.
41. W.G. Horner (1819): A new method of solving numerical equations of all orders, by continuous approximation. *Philosophical Transactions. Royal Society of London*, 308-335.
42. S. Jaiswal, A.L. Rosenberg, D. Towsley (2004): Comparing the structure of power-law graphs and the Internet AS graph. *12th IEEE Int'l Conf. on Network Protocols (ICNP'04)*.
43. S.L. Johnsson, C.T. Ho (1989): Optimum broadcasting and personalized communication in hypercubes. *IEEE Trans. Computers 38*, 1249–1268.
44. R.M. Karp (1972): Reducibility among combinatorial problems. In *Complexity of Computer Computations* (R.E. Miller and J.W. Thatcher, eds.) Plenum Press, NY, pp. 85–103.
45. S.C. Kleene (1936): General recursive functions of natural numbers. *Math. Annalen 112*, 727–742.
46. S.C. Kleene (1952): *Introduction to Metamathematics.* D. Van Nostrand, Princeton, NJ.
47. D. König (1936): *Theorie der endlichen und unendlichen Graphen.* Lipzig: Akad. Verlag.
48. H.W. Kuhn (1955): The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*, 83-97.
49. K. Kuratowski (1930): Sur le problème des courbes gauches en topologie. *Fundamenta Mathematica 15*, 271–283.

50. Kwan, Mei-Ko (1960): Graphic programming using odd or even points. *Acta Mathematica Sinica, 10* (in Chinese), 263–266. Translated into English in *Chinese Mathematics 1* (1962) 273-277.
51. G.W. Leibniz (Leibnitz) (1674-76): *Sämtliche Schriften und Briefe, Reihe VII: Mathematische Schriften, vol. 5: Infinitesimalmathematik*. Akademie Verlag, Berlin.
52. J.S. Lew and A.L. Rosenberg (1978): Polynomial indexing of integer lattices, I. *J. Number Th. 10*, 192–214.
53. J.S. Lew and A.L. Rosenberg (1978): Polynomial indexing of integer lattices, II. *J. Number Th. 10*, 215–243.
54. J.E. Littlewood (1953): *A Mathematician's Miscellany.* Methuen & Co, Ltd. Reprinted as *Littlewood's Miscellany* (B. Bollobàs, ed.), 1986, University Press, Cambridge.
55. D. Liu, N. Blenn, P. Van Mieghem (2010): Modeling social networks with overlapping communities using hypergraphs and their line graphs. Report arXiv:1012.2774, Dec. 2010, `http://cds.cern.ch/record/1314107`.
56. L. Lovasz (1973): Coverings and colorings of hypergraphs. *4th Southeast Conf. on Combinatorics, Graph Theory, and Computing*, 3–12.
57. C. Mead and L. Conway (1979): *Introduction to VLSI Systems*. Addison-Wesley, Reading, MA., (ISBN 0201043580).
58. R.E. Miller, N. Pippenger, A.L. Rosenberg, L. Snyder (1979): Optimal 2,3-trees. *SIAM J. Comput. 8*, 42–59.
59. I. Newton (1687): *Philosophi Naturalis Principia Mathematica* (known popularly as *Principia Mathematica*). Royal Society.
60. I. Niven and H.S. Zuckerman (1980): *An Introduction to the Theory of Numbers.* (4th ed.) J. Wiley & Sons, New York.
61. J.A. Paulos (1990): *Innumeracy: Mathematical Iliteracy and Its Consequences.* Vintage Books (Random House), New York.
62. W.W. Peterson, E.J. Weldon, Jr. (1981): *Error-Correcting Codes.* (2nd Ed.) MIT Press, Cambridge, Mass.
63. A.L. Rosenberg (1974): Allocating storage for extendible arrays. *J. ACM 21*, 652–670.
64. A.L. Rosenberg (1975): Managing storage for extendible arrays. *SIAM J. Comput. 4*, 287–306.
65. A.L. Rosenberg (1979): Profile numbers. *Fibonacci Quart. 17*(3), 259–264.
66. A.L. Rosenberg (1985): A hypergraph model for fault-tolerant VLSI processor arrays. *IEEE Trans. Comput. C-34*, 578–584.
67. A.L. Rosenberg (1989): Interval hypergraphs. In *Graphs and Algorithms* (R.B. Richter, ed.) *Contemporary Mathematics 89*, Amer. Math. Soc., 27–44.
68. A.L. Rosenberg (2003): Accountable Web-computing. *IEEE Trans. Parallel and Distr. Systs. 14*, 97–106.
69. A.L. Rosenberg (2003): Efficient pairing functions—and why you should care. *Intl. J. Foundations of Computer Science 14*, 3–17.
70. A.L. Rosenberg (2009): *The Pillars of Computation Theory: State, Encoding, Nondeterminism*. Universitext Series, Springer, New York
71. A.L. Rosenberg and L.S. Heath (2001): *Graph Separators, with Applications*. Kluwer Academic/Plenum Publishers, New York.
72. A.L. Rosenberg and L. Snyder (1978):Minimal-comparison 2,3-trees. *SIAM J. Comput. 7*, 465–480.
73. A.L. Rosenberg and L.J. Stockmeyer (1977): Hashing schemes for extendible arrays. *J. ACM 24*, 199–221.
74. S.M. Ross (1976): *A First Course in Probability*. Pearson Education.
75. J.B. Rosser (1953): *Logic for Mathematicians.* McGraw-Hill, New York.
76. B. Russell (1903). *Principles of Mathematics*. Cambridge University Press.
77. Y. Saad, M.H. Schultz (1989): Data communication in hypercubes. *J. Parallel and Distributed Computing 6*, 115–135.
78. M. Schönfinkel (1924): Über die Bausteine der mathematischen Logik. *Math. Annalen 92*, 305–316.

79. E. Schröder (1898): Über zwei Definitionen der Endlichkeit und G. Cantor'sche Sätze. *Nova Acta Academiae Caesareae Leopoldino-Carolinae (Halle a.d. Saale) 71*, 303–362.
80. E. Schröder (1898): Die selbständige Definition der Mächtigkeiten 0, 1, 2, 3 und die explicite Gleichzahligkeitsbedingung. *Nova Acta Academiae Caesareae Leopoldino-Carolinae (Halle a.d. Saale) 71*, 365–376.
81. J.T. Schwartz (1980): Ultracomputers. *ACM Trans. Programming Languages 2*, 484–521.
82. H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, W. Willinger (2002): Network topology generators: Degree-based vs. structural. *ACM SIGCOMM'02*.
83. A.M. Turing (1936): On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.* (ser. 2, vol. 42) 230–265; Correction *ibid.* (vol. 43) 544–546.
84. J.D. Ullman (1984): *Computational Aspects of VLSI.* Computer Science Press, Rockville, Md.
85. M. Yoeli (1962): Binary ring sequences. *Amer. Math. Monthly 69*, 852–855.
86. E. Zegura, K.L. Calvert, M.J. Donohoo (1997): A quantitative comparison of graph-based models for internetworks. *IEEE/ACM Trans. on Networking, 5*(6), 770–783.

Index