13 State Machines

DRAFT WTF?

We've already demonstrated the use of state machines as abstract models of step-by-step processes. In this chapter we examine two further applications of stste machines, first as models of concurrent computational processes, and second as the basis for reasoning about programs.

13.1 The Alternating Bit Protocol

The Alternating Bit Protocol is a well-known two-process communication protocol that achieves reliable FIFO communication over unreliable channels that operate concurrently. The unreliable channels may lose or duplicate messages, but are assumed not to reorder them. We'll use the Invariant Method to verify the Protocol.

The Protocol allows a **Sender** process to send a sequence of messages from a message alphabet, M, to a **Receiver** process. It works as follows.

Sender repeatedly sends the rightmost message in its **outgoing-queue** of messages, tagged with a **tagbit** that is initially 1. When **Receiver** receives this tagged message, it sets its **ackbit** to be the message tag 1, and adds the message to the left-hand end of its **received-msgs** list. Then as an acknowledgement, **Receiver** sends back **ackbit** 1 repeatedly. When **Sender** gets this acknowledgement bit, it deletes the rightmost outgoing message in its queue, sets its **tagbit** to 0, and begins sending the new rightmost outgoing message, tagged with **tagbit**.

Receiver, having already accepted the message tagged with **ackbit** 1, ignores subsequent messages with tag 1, and waits until it sees the first message with tag 0; it adds this message to the lefthand side of its **received-msgs** list, sets **ackbit** to 0 and acknowledges repeatedly with with **ackbit** 0. **Sender** now waits till it gets acknowledgement bit 0, then goes on to send the next outgoing message with tag 1. In this way, it alternates use of the tags 1 and 0 for successive messages.

We claim that this causes **Sender** to receive *suffix* original **outgoing-msgs** queue. That is, at any stage in the process when the the **outgoing-msgs**

(The fact that **Sender** actually outputs the entire outgoing queuee is a *liveness* claim—liveness properties are a generalization of termination properties. We'll ignore this issue for now.)

We formalize the description above as a state whose states consist of:

outgoing-msgs, a finite sequence of M, whose initial value is called **all-msgs**
tagbit $\in \{0, 1\}$, initially 1

received-msgs, a finite sequence of M, initially empty
ackbit($\in \{0, 1\}$, initially 0

msg-channel, a finite sequence of $M \times \{0, 1\}$, initially empty,
ack-channel, a finite sequence of $\{0, 1\}$, initially empty

The transitions are:

SEND: (a) **action**: **send-msg**(m, b)
precondition: $m =$ **rightend**(**outgoing-msgs**) AND $b =$ **tagbit**
effect: add (m, b) to the lefthand end of **msg-channel**, any number ≥ 0 of times

(b) **action**: **send-ack**(b)
precondition: $b =$ **ackbit**
effect: add b to the righthand end of **ack-channel**, any number ≥ 0 of times

RECEIVE: (a) **action**: **receive-msg**(m, b)
precondition: $(m, b) =$ **rightend**(**msg-channel**)
effect: remove **rightend** of **msg-channel**;
if $b \neq$ **ackbit**, then [add m to the lefthand end of **receive-msgs**; **ackbit** $:= b$.]

(b) **action**: **receive-ack**(b)
precondition: $b =$ **leftend**(**ack-channel**()
effect: remove **leftend** of **ack-channel**.
if $b =$ **tagbit**, then [remove **rightend** of **outgoing-msgs** (if nonempty);
tagbit $:=$ $\overline{\textbf{tagbit}}$]

Our goal is to show that when **tagbit** $\neq$ **ackbit**, then

$$\textbf{outgoing-queue} \cdot \textbf{received-msgs} = \textbf{all-msgs}. \tag{13.1}$$

This requires three auxiliary invariants. For the first of these, we need a definition.

Let **tag-sequence** be the sequence consisting of bits in **ack-channel**, in right-to-left order, followed by **tagbit**, followed by the tag components of the elements of **msg-channel**, in left-to-right order, followed by **ackbit**.

Property 2: **tag-sequence** consists of one of the following:

1. All 0's.
2. All 1's.
3. A positive number of 0's followed by a positive number of 1's.
4. A positive number of 1's followed by a positive number of 0's.

What is being ruled out by these four cases is the situation where the sequence contains more than one switch of tag value.

The fact that Property 2 is an invariant can be proved easily by induction. We also need:

Property 3: If $(m, \textbf{tag})$ is in **msg-channel** then $m =$ **rightend(outgoing-queue)**.

Proof. (That Property 3 is an invariant)

By induction, using Property 2.

Base: Obvious, since no message is in the channel initially.

Inductive step: It is easy to see that the property is preserved by $\textbf{send}_{m,b}$, which adds new messages to $\textbf{channel}_{1,2}$. The only other case that could cause a problem is $\textbf{receive}(b)_{2,1}$, which could cause $\textbf{tag}_1$ to change when there is another message already in $\textbf{channel}_{1,2}$ with the same tag. But this can't happen, by Property 2 applied before the step—since the incoming tag g must be equal to $\textbf{tag}_1$ in this case, all the tags in **tag-sequence** must be the same. ■

Finally, we need that the following counterpart to (13.1): when **tagbit** = **ackbit**, then

$$\textbf{lefttail(outgoing-queue)} \cdot \textbf{received-msgs} = \textbf{all-msgs}, \tag{13.2}$$

where **lefttail(outgoing-queue)** all but the rightmost message, if any, in **outgoing-queue**.

Property 4, part 2, easily implies the goal Property 1. It also implies that $\textbf{work-buf}_2$ is always nonempty when $\textbf{receive}(b)_{2,1}$ occurs with equal tags; therefore, the parenthetical check in the code always works out to be true.

Proof. (That Property 4 is an invariant)

By induction. Base: In an initial state, the tags are unequal, $\textbf{work-buf}_1 = \textbf{buf}_1$ and $\textbf{buf}_2$ is empty. This suffices to show part 1. part 2 is vacuous.

Inductive step: When a **send** occurs, the tags and buffers are unchanged, so the truth of the invariants must be preserved. It remains to consider **receive** events.

$\textbf{receive}(m, b)_{1,2}$:

If $b = \textbf{tag}_2$, nothing happens, so the invariants are preserved. So suppose that $b \neq \textbf{tag}_2$. Then Property 2 implies that $b = \textbf{tag}_1$, and then Property 3 implies that

m is the first message on $\mathbf{work\text{-}buf}_1$. The effect of the transition is to change $\mathbf{tag}_2$ to make it equal to $\mathbf{tag}_1$, and to replicate the first element of $\mathbf{work\text{-}buf}22_1$ at the end of $\mathbf{buf}_2$.

The inductive hypothesis implies that, before the step, $\mathbf{buf}_2 \cdot \mathbf{work\text{-}buf}_1 = \mathbf{buf}_1$. The changes caused by the step imply that, after the step, $\mathbf{tag}_1 = \mathbf{tag}_2$, $\mathbf{work\text{-}buf}_1$ and $\mathbf{buf}_2$ are nonempty, $\mathbf{head}(\mathbf{work\text{-}buf}_1) = \mathbf{last}(\mathbf{buf}_2)$, and $\mathbf{buf}_2 \cdot \mathbf{tail}(\mathbf{work\text{-}buf}_1) = \mathbf{buf}_1$. This is as needed.

$\mathbf{receive}(b)_{2,1}$:

The argument is similar to the one for $\mathbf{receive}(m, b)_{1,2}$. If $b \neq \mathbf{tag}_1$, nothing happens so the invariants are preserved. So suppose that $b = \mathbf{tag}_1$. Then Property 2 implies that $b = \mathbf{tag}_2$, and the step changes $\mathbf{tag}_1$ to make it unequal to $\mathbf{tag}44_2$. The step also removes the first element of $\mathbf{work\text{-}buf}_1$. The inductive hypothesis implies that, before the step, $\mathbf{work\text{-}buf}_1$ and $\mathbf{buf}_2$ are nonempty, $\mathbf{head}(\mathbf{work\text{-}buf}_1) = \mathbf{last}(\mathbf{buf}_2)$, and $\mathbf{buf}_2 \cdot \mathbf{tail}(\mathbf{work\text{-}buf}_1) = \mathbf{buf}_1$. The changes caused by the step imply that, after the step, $\mathbf{tag}_1 \neq \mathbf{tag}_2$ and $\mathbf{buf}_2 \cdot \mathbf{work\text{-}buf}_1 = \mathbf{buf}_1$. This is as needed. ■

"How is this SM"

13.2 Reasoning About While Programs

Real programs and programming languages are often huge and complicated, making them hard to model and even harder to reason about. Still, making programs "reasonable" is a crucial aspect of software engineering. In this section we'll illustrate what it means to have a clean mathematical model of a simple programming language and reasoning principles that go with it—if only real programming languages allowed for such simple, accurate modeling.

13.2.1 While Programs

The programs we'll study are called "***while*** *programs*." We can define them as a recursive data type:

Definition 13.2.1.

base cases:

- $x := e$ is a **while** program, called an *assignment statement*, where x is a variable and e is an expression.
- ***Done*** is a **while** program.

constructor cases: If C and D are **while** programs, and T is a test, then the following are also **while** programs:

- $C;D$—called the *sequencing* of C and D,
- **if** T **then** C **else** D—called a *conditional* with *test*, T, and *branches*, C and D,
- **while** T **do** C **od**—called a *while loop* with *test*, T, and *body*, C.

For simplicity we'll stick to **while** programs operating on integers. So by expressions we'll mean any of the familiar integer valued expressions involving integer constants and operations such as addition, multiplication, exponentiation, quotient or remainder. As *tests*, we'll allow propositional formulas built from basic formulas of the form $e \leq f$ where e and f are expressions. For example, here is the Euclidean algorithm for $\gcd(a, b)$ expressed as a **while** program.

$\mathtt{x} := a$**;**
$\mathtt{y} := b$**;**
while $\mathtt{y} \neq 0$ **do**
    $\mathtt{t} := \mathtt{y}$**;**
    $\mathtt{y} := \mathrm{rem}(\mathtt{x}, \mathtt{y})$**;**
    $\mathtt{x} := \mathtt{t}$ **od**

13.2.2 The While Program State Machine

A **while** program acts as a pure command; it is run solely for its side effects on stored data and it doesn't return a value. The data consists of integers stored as the values of variables, namely environments:

Definition 13.2.2. An *environment* is a total function from variables to integers. Let Env be set of all environments.

So if ρ is an environment and x is a variable, then $\rho(x)$ is an integer. More generally, the environment determines the integer value of each expression, e, and the truth value of each test, T. We can think of an expression, e as defining a function $[\![e]\!] : \mathrm{Env} \to \mathbb{Z}$, and refer to this function, $[\![e]\!]$ as the *meaning* of e, and likewise for tests.

It's standard in programming language theory to write $[\![e]\!]\rho$ as shorthand for $[\![e]\!](\rho)$, that is, applying the *meaning*, $[\![e]\!]$, of e to ρ. For example, if $\rho(\mathtt{x}) = 4$, and $\rho(\mathtt{y}) = -2$, then

$$[\![\mathtt{x}^2 + \mathtt{y} - 3]\!]\rho = \rho(\mathtt{x})^2 + \rho(y) - 3 = 11. \tag{13.3}$$

Executing a program causes a succession of changes to the environment[1] which may continue until the program halts. Actually the only command which immediately alters the environment is an assignment command. Namely, the effect of the command

$$x := e$$

on an environment is that the value assigned to the variable x is changed to the value of e in the original environment. We can say this precisely and concisely using the following notation: $f[a \leftarrow b]$ is a function that is the same as the function, f, except that when applied to element a its value is b. Namely,

Definition 13.2.3. If $f : A \to B$ is a function and a, b are arbitrary elements, define

$$f[a \leftarrow b]$$

to be the function g such that

$$g(u) = \begin{cases} b & \text{if } u = a. \\ f(u) & \text{otherwise.} \end{cases}$$

Now we can specify the step-by-step execution of a **while** program as a state machine, where the states of the machine consist of a **while** program paired with an environment. The transitions of this state machine are defined recursively on the definition of **while** programs.

Definition 13.2.4. The transitions $\langle C, \rho\rangle \longrightarrow \langle D, \rho'\rangle$ of the **while** program state machine are defined as follows:

base cases:

$$\langle x := e, \rho\rangle \longrightarrow \langle \textbf{Done}, \rho[x \leftarrow [\![e]\!]\rho]\rangle$$

constructor cases: If C and D are **while** programs, and T is a test, then:

- if $\langle C, \rho\rangle \longrightarrow \langle C', \rho'\rangle$, then

$$\langle C;D, \rho\rangle \longrightarrow \langle C';D, \rho'\rangle.$$

Also,

$$\langle \textbf{Done};D, \rho\rangle \longrightarrow \langle D, \rho\rangle.$$

[1] More sophisticated programming models distinguish the environment from a *store* which is affected by commands, but this distinction is unnecessary for our purposes.

- if $[\![T]\!]\rho = \mathbf{T}$, then

$$\langle \textbf{if } T \textbf{ then } C \textbf{ else } D,\ \rho\rangle \longrightarrow \langle C,\ \rho\rangle\,,$$

or if $[\![T]\!]\rho = \mathbf{F}$, then

$$\langle \textbf{if } T \textbf{ then } C \textbf{ else } D,\ \rho\rangle \longrightarrow \langle D,\ \rho\rangle\,.$$

- if $[\![T]\!]\rho = \mathbf{T}$, then

$$\langle \textbf{while } T \textbf{ do } C \textbf{ od},\ \rho\rangle \longrightarrow \langle C;\textbf{while } T \textbf{ do } C \textbf{ od},\ \rho\rangle$$

or if $[\![T]\!]\rho = \mathbf{F}$, then

$$\langle \textbf{while } T \textbf{ do } C \textbf{ od},\ \rho\rangle \longrightarrow \langle \textbf{Done},\ \rho\rangle\,.$$

Now **while** programs are probably going to be the simplest kind of programs you will ever see, but being condescending about them would be a mistake. It turns that *every function on nonnegative integers that can be computed by any program* on any machine whatsoever can also be computed by **while** programs (maybe more slowly). We can't take the time to explain how such a sweeping claim can be justified, but you can find out by taking a course in computability theory such as 6.045 or 6.840.

13.2.3 Denotational Semantics

The net effect of starting a **while** program in some environment is reflected in the final environment when the program halts. So we can think of a **while** program, C, as defining a function, $[\![C]\!] : \text{Env} \to \text{Env}$, from initial environments to environments at halting. The function $[\![C]\!]$ is called the *meaning* of C.

$[\![C]\!]$ of a **while** program, C to be a partial function from Env to Env mapping an initial environment to the final halting environment.

We'll need one bit of notation first. For any function $f : S \to S$, let $f^{(n)}$ be the composition of f with itself n times where $n \in \mathbb{N}$. Namely,

$$f^{(0)} ::= \mathrm{Id}_S$$
$$f^{(n+1)} ::= f \circ f^{(n)},$$

where "$\circ$" denotes functional composition.

The recursive definition of the meaning of a program follows the definition of the **while** program recursive data type.

Definition 13.2.5. base cases:

- $[\![x := e]\!]$ is the function from Env to Env defined by the rule:

$$[\![x := e]\!]\rho ::= \rho[x \leftarrow [\![e]\!]\rho].$$

- $$[\![\textbf{Done}]\!] ::= \mathrm{Id}_{\mathrm{Env}}$$

  where $\mathrm{Id}_{\mathrm{Env}}$ is the identity function on Env. In other words, $[\![\textbf{Done}]\!]\rho ::= \rho$.

constructor cases: If C and D are **while** programs, and T is a test, then:

- $$[\![C;D]\!] ::= [\![D]\!] \circ [\![C]\!]$$

  That is,

$$[\![C;D]\!]\rho ::= [\![D]\!]([\![C]\!]\rho).$$

- $$[\![\textbf{if } T \textbf{ then } C \textbf{ else } D]\!]\rho ::= \begin{cases} [\![C]\!]\rho & \text{if } [\![T]\!]\rho = \mathbf{T} \\ [\![D]\!]\rho & \text{if } [\![T]\!]\rho = \mathbf{F}. \end{cases}$$

- $$[\![\textbf{while } T \textbf{ do } C \textbf{ od}]\!]\rho ::= [\![C]\!]^{(n)}\rho$$

  where n is the least nonnegative integer such that $[\![T]\!]([\![C]\!]^{(n)}\rho) = \mathbf{F}$. (If there is no such n, then $[\![\textbf{while } T \textbf{ do } C \textbf{ od}]\!]\rho$ is undefined.)

We can use the denotational semantics of **while** programs to reason about **while** programs using structural induction on programs, and this is often much simpler than reasoning about them using induction on the number of steps in an execution. This is OK as long as the denotational semantics accurately captures the state machine behavior. In particular, using the notation $\longrightarrow^*$ for the transitive closure of the transition relation:

Theorem 13.2.6.

$$\langle C, \rho\rangle \longrightarrow^* \langle \textit{Done}, \rho'\rangle \quad \textit{iff} \quad [\![C]\!]\rho = \rho'$$

Theorem 13.2.6 can be proved easily by induction; it appear in Problem13.1.

13.2.4 Logic of Programs

A typical program specification describes the kind of inputs and environments the program should handle, and then describes what should result from an execution. The specification of the inputs or initial environment is called the *precondition* for program execution, and the prescription of what the result of execution should be is called the *postcondition*. So if P is a logical formula expressing the precondition for a program, C, and likewise Q expresses the postcondition, the specification requires that

If P holds when C is started, then Q will hold if and when C halts.

We'll express this requirement as a formula

$$P\,\{C\}\,Q$$

called a *partial correctness assertion*.

For example, if E is the **while** program above for the Euclidean algorithm, then the partial correctness of E can be expressed as

$$(a, b \in \mathbb{N} \text{ AND } \mathrm{a} \neq 0)\ \{E\}\ (\mathrm{x} = \gcd(a, b))\,. \tag{13.4}$$

More precisely, notice that just as the value of an expression in an environment is an integer, the value of a logical formula in an environment is a truth value. For example, if $\rho(\mathrm{x}) = 4$, and $\rho(\mathrm{y}) = -2$, then by (13.3), $[\![\mathrm{x}^2 + \mathrm{y} - 3]\!]\rho = 11$, so

$$[\![\exists \mathrm{z}.\ \mathrm{z} > 4 \text{ AND } \mathrm{x}^2 + \mathrm{y} - 3 = \mathrm{z}]\!]\rho = \mathbf{T},$$
$$[\![\exists \mathrm{z}.\ \mathrm{z} > 13 \text{ AND } \mathrm{x}^2 + \mathrm{y} - 3 = \mathrm{z}]\!]\rho = \mathbf{F}.$$

Definition 13.2.7. For logical formulas P and Q, and **while** program, C, the partial correctness assertion

$$P\,\{C\}\,Q$$

is true proving that for all environments, rho, if $[\![P]\!]\rho$ is true, and $\langle C, \rho\rangle \longrightarrow^* \langle \mathbf{Done}, \rho'\rangle$ for some ρ', then $[\![Q]\!]\rho'$ is true.

In the 1970', Univ. Dublin formulated a set of inference rules for proving partial correctness formulas. These rules are known as *Hoare Logic*.

The first rule captures the fact that strengthening the preconditions and weakening the postconditions makes a partial correctness specification easier to satisfy:

$$\frac{P \text{ IMPLIES } R,\quad R\,\{C\}\,S,\quad S \text{ IMPLIES } Q}{P\,\{C\}\,Q}$$

The rest of the logical rules follow the recursive definition of **while** programs. There are axioms for the base case commands:

$$P(x)\ \{x := e\}\ P(e)$$
$$P\ \{\textbf{Done}\}\ P,$$

and proof rules for the constructor cases:

- $$\frac{P\ \{C\}\ Q \ \text{ AND } \ Q\ \{D\}\ R}{P\ \{C;D\}\ R}$$

- $$\frac{P \ \text{ AND } \ T\ \{C\}\ Q}{P \ \text{ AND } \ T\ \{\textbf{if } T \textbf{ then } C \textbf{ else } D\}\ Q \ \text{ AND } \ T}$$

- $$\frac{P \ \text{ AND } \ T\ \{C\}\ P}{P\ \{\textbf{while } T \textbf{ do } C \textbf{ od}\}\ P \ \text{ AND } \ \text{NOT}(T)}$$

Example 13.2.8. TBA - Formal correctness proof of (13.4) for the Euclidean algorithm.

Problems for Section 13.2

Homework Problems

Problem 13.1.
Prove Theorem 13.2.6:

Theorem.

$$\langle C, \rho\rangle \longrightarrow^* \langle \textit{Done}, \rho'\rangle \quad \textit{iff} \quad [\![C]\!]\rho = \rho'$$

Hint: Prove the left to right direction by induction on the number of steps C needs to halt starting in environment ρ. Prove the right to left direction by structural induction on the definition of **while** programs. Both proofs follow almost mechanically from the definitions.

14 Sums and Asymptotics

Sums and products arise regularly in the analysis of algorithms, financial applications, physical problems, and probabilistic systems. For example, according to Theorem 2.3.1,

$$1+2+3+\cdots+n=\frac{n(n+1)}{2}. \tag{14.1}$$

Of course the lefthand sum could be expressed concisely as a subscripted summation

$$\sum_{i=1}^{n} i,$$

but the right hand expression $n(n+1)/2$ is not only concise, it is also easier to evaluate, and it more clearly reveals properties such as the growth rate of the sum. Expressions like $n(n+1)/2$ that do not make use of subscripted summations or products —or those handy but sometimes troublesome dots —are called *closed forms*.

Another example is the closed from for a *geometric sum*

$$1+x+x^2+x^3+\cdots+x^n=\frac{1-x^{n+1}}{1-x} \tag{14.2}$$

given in Problem 6.2. The sum as described on the left hand side of (14.2) involves n additions and $1+2+\cdots+(n-1)=(n-1)n/2$ multiplications, but its closed form on the right hand side can be evaluated using fast exponentiation with at most $2\log n$ multiplications and a couple of substractions. Also, the closed form makes the growth and limiting behavior of the sum much more apparent.

Equations (14.1) and (14.2) were easy to verify by induction, but, as is often the case, the proofs by induction gave no hint about how these formulas were found in the first place. Finding them is part math and part art, which we'll start examining in this chapter.

A first motivating example will be figuring out the value of the annuity. The value will be a large and nasty-looking sum. We will then describe several methods for finding closed forms for several sorts of sums, including the annuity sums. In some cases, a closed form for a sum may not exist and so we will provide a general method for finding closed forms for good upper and lower bounds on the sum.

The methods we develop for sums will also work for products since any product can be converted into a sum by taking a logarithm of the product. As an example,

we will use this approach to find a good closed-form approximation to the *factorial function*

$$n! ::= 1 \cdot 2 \cdot 3 \cdots n.$$

We conclude the chapter with a discussion of asymptotic notation. Asymptotic notation is often used to bound the error terms when there is no exact closed form expression for a sum or product. It also provides a convenient way to express the growth rate or order of magnitude of a sum or product.

14.1 The Value of an Annuity

Would you prefer a million dollars today or $50,000 a year for the rest of your life? On the one hand, instant gratification is nice. On the other hand, the *total dollars* received at $50K per year is much larger if you live long enough.

Formally, this is a question about the value of an annuity. An *annuity* is a financial instrument that pays out a fixed amount of money at the beginning of every year for some specified number of years. In particular, an n-year, m-payment annuity pays m dollars at the start of each year for n years. In some cases, n is finite, but not always. Examples include lottery payouts, student loans, and home mortgages. There are even Wall Street people who specialize in trading annuities.[1]

A key question is, "What is an annuity worth?" For example, lotteries often pay out jackpots over many years. Intuitively, $50,000 a year for 20 years ought to be worth less than a million dollars right now. If you had all the cash right away, you could invest it and begin collecting interest. But what if the choice were between $50,000 a year for 20 years and a *half* million dollars today? Now it is not clear which option is better.

14.1.1 The Future Value of Money

In order to answer such questions, we need to know what a dollar paid out in the future is worth today. To model this, let's assume that money can be invested at a fixed annual interest rate p. We'll assume an 8% rate[2] for the rest of the discussion.

Here is why the interest rate p matters. Ten dollars invested today at interest rate p will become $(1+p)\cdot 10 = 10.80$ dollars in a year, $(1+p)^2 \cdot 10 \approx 11.66$ dollars

[1] Such trading ultimately led to the subprime mortgage disaster in 2008–2009. We'll talk more about that in a later chapter.

[2] U.S. interest rates have dropped steadily for several years, and ordinary bank deposits now earn around 1.0%. But just a few years ago the rate was 8%; this rate makes some of our examples a little more dramatic. The rate has been as high as 17% in the past thirty years.

in two years, and so forth. Looked at another way, ten dollars paid out a year from now is only really worth $1/(1+p)\cdot 10 \approx 9.26$ dollars today. The reason is that if we had the \$9.26 today, we could invest it and would have \$10.00 in a year anyway. Therefore, p determines the value of money paid out in the future.

So for an n-year, m-payment annuity, the first payment of m dollars is truly worth m dollars. But the second payment a year later is worth only $m/(1+p)$ dollars. Similarly, the third payment is worth $m/(1+p)^2$, and the n-th payment is worth only $m/(1+p)^{n-1}$. The total value, V, of the annuity is equal to the sum of the payment values. This gives:

$$\begin{aligned} V &= \sum_{i=1}^{n} \frac{m}{(1+p)^{i-1}} \\ &= m \cdot \sum_{j=0}^{n-1} \left(\frac{1}{1+p}\right)^{j} && \text{(substitute } j = i-1\text{)} \\ &= m \cdot \sum_{j=0}^{n-1} x^{j} && \text{(substitute } x = 1/(1+p)\text{).} \end{aligned} \tag{14.3}$$

The goal of the preceding substitutions was to get the summation into the form of a simple geometric sum. This leads us to an explanation of a way you could have discovered the closed form (14.2) in the first place using the *Perturbation Method*.

14.1.2 The Perturbation Method

Given a sum that has a nice structure, it is often useful to "perturb" the sum so that we can somehow combine the sum with the perturbation to get something much simpler. For example, suppose

$$S = 1 + x + x^2 + \cdots + x^n.$$

An example of a perturbation would be

$$xS = x + x^2 + \cdots + x^{n+1}.$$

The difference between S and xS is not so great, and so if we were to subtract xS from S, there would be massive cancellation:

$$\begin{aligned} S &= 1 + x + x^2 + x^3 + \cdots + x^{n-1} \\ -xS &= \quad -x - x^2 - x^3 - \cdots - x^{n-1} - x^n. \end{aligned}$$

The result of the subtraction is

$$S - xS = 1 - x^n.$$

Solving for S gives the desired closed-form expression in equation 14.2, namely,

$$S = \frac{1 - x^{n+1}}{1 - x}.$$

We'll see more examples of this method when we introduce *generating functions* in a later Chapter.

oh that was not that hard!

14.1.3 A Closed Form for the Annuity Value

Using equation 14.2, we can derive a simple formula for V, the value of an annuity that pays m dollars at the start of each year for n years.

$$V = m\left(\frac{1 - x^n}{1 - x}\right) \qquad \text{(by equations 14.3 and 14.2)} \qquad (14.4)$$

$$= m\left(\frac{1 + p - (1/(1+p))^{n-1}}{p}\right) \qquad \text{(substituting } x = 1/(1+p)\text{)}. \qquad (14.5)$$

Equation 14.5 is much easier to use than a summation with dozens of terms. For example, what is the real value of a winning lottery ticket that pays \$50,000 per year for 20 years? Plugging in $m = \$50{,}000$, $n = 20$, and $p = 0.08$ gives $V \approx \$530{,}180$. So because payments are deferred, the million dollar lottery is really only worth about a half million dollars! This is a good trick for the lottery advertisers.

at 8% = r

14.1.4 Infinite Geometric Series

The question we began with was whether you would prefer a million dollars today or \$50,000 a year for the rest of your life. Of course, this depends on how long you live, so optimistically assume that the second option is to receive \$50,000 a year *forever*. This sounds like infinite money! But we can compute the value of an annuity with an infinite number of payments by taking the limit of our geometric sum in equation 14.2 as n tends to infinity.

Theorem 14.1.1. *If* $|x| < 1$, *then*

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1 - x}.$$

1/r

Proof.

$$\begin{aligned}\sum_{i=0}^{\infty} x^i &::= \lim_{n\to\infty} \sum_{i=0}^{n} x^i \\ &= \lim_{n\to\infty} \frac{1-x^{n+1}}{1-x} && \text{(by equation 14.2)} \\ &= \frac{1}{1-x}.\end{aligned}$$

The final line follows from that fact that $\lim_{n\to\infty} x^{n+1} = 0$ when $|x| < 1$. ■

In our annuity problem, $x = 1/(1+p) < 1$, so Theorem 14.1.1 applies, and we get

$$\begin{aligned}V &= m \cdot \sum_{j=0}^{\infty} x^j && \text{(by equation 14.3)} \\ &= m \cdot \frac{1}{1-x} && \text{(by Theorem 14.1.1)} \\ &= m \cdot \frac{1+p}{p} && (x = 1/(1+p)).\end{aligned}$$

Plugging in $m = \$50{,}000$ and $p = 0.08$, we see that the value V is only \$675,000. Amazingly, a million dollars today is worth much more than \$50,000 paid every year forever! Then again, if we had a million dollars today in the bank earning 8% interest, we could take out and spend \$80,000 a year forever. So on second thought, this answer really isn't so amazing.

14.1.5 Examples

Equation 14.2 and Theorem 14.1.1 are incredibly useful in computer science. Here are some other common sums that can be put into closed form using equa-

tion 14.2 and Theorem 14.1.1:

$$1 + 1/2 + 1/4 + \cdots = \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i = \frac{1}{1-(1/2)} = 2 \tag{14.6}$$

$$0.99999\cdots = 0.9\sum_{i=0}^{\infty} \left(\frac{1}{10}\right)^i = 0.9\left(\frac{1}{1-1/10}\right) = 0.9\left(\frac{10}{9}\right) = 1 \tag{14.7}$$

$$1 - 1/2 + 1/4 - \cdots = \sum_{i=0}^{\infty} \left(\frac{-1}{2}\right)^i = \frac{1}{1-(-1/2)} = \frac{2}{3} \tag{14.8}$$

$$1 + 2 + 4 + \cdots + 2^{n-1} = \sum_{i=0}^{n-1} 2^i = \frac{1-2^n}{1-2} = 2^n - 1 \tag{14.9}$$

$$1 + 3 + 9 + \cdots + 3^{n-1} = \sum_{i=0}^{n-1} 3^i = \frac{1-3^n}{1-3} = \frac{3^n-1}{2} \tag{14.10}$$

If the terms in a geometric sum grow smaller, as in equation 14.6, then the sum is said to be *geometrically decreasing*. If the terms in a geometric sum grow progressively larger, as in equations 14.9 and 14.10, then the sum is said to be *geometrically increasing*. In either case, the sum is usually approximately equal to the term in the sum with the greatest absolute value. For example, in equations 14.6 and 14.8, the largest term is equal to 1 and the sums are 2 and 2/3, both relatively close to 1. In equation 14.9, the sum is about twice the largest term. In equation 14.10, the largest term is 3^{n-1} and the sum is $(3^n - 1)/2$, which is only about a factor of 1.5 greater. You can see why this rule of thumb works by looking carefully at equation 14.2 and Theorem 14.1.1.

14.1.6 Variations of Geometric Sums

We now know all about geometric sums —if you have one, life is easy. But in practice one often encounters sums that cannot be transformed by simple variable substitutions to the form $\sum x^i$.

A non-obvious, but useful way to obtain new summation formulas from old is by differentiating or integrating with respect to x. As an example, consider the following sum:

$$\sum_{i=1}^{n-1} i x^i = x + 2x^2 + 3x^3 + \cdots + (n-1)x^{n-1}$$

This is not a geometric sum, since the ratio between successive terms is not fixed, and so our formula for the sum of a geometric sum cannot be directly applied. But

differentiating equation 14.2 leads to:

$$\frac{d}{dx}\left(\sum_{i=0}^{n-1} x^i\right) = \frac{d}{dx}\left(\frac{1-x^n}{1-x}\right). \tag{14.11}$$

The left-hand side of equation 14.11 is simply

$$\sum_{i=0}^{n-1} \frac{d}{dx}(x^i) = \sum_{i=0}^{n-1} i x^{i-1}.$$

The right-hand side of equation 14.11 is

$$\begin{aligned}\frac{-nx^{n-1}(1-x)-(-1)(1-x^n)}{(1-x)^2} &= \frac{-nx^{n-1}+nx^n+1-x^n}{(1-x)^2}\\ &= \frac{1-nx^{n-1}+(n-1)x^n}{(1-x)^2}.\end{aligned}$$

Hence, equation 14.11 means that

$$\sum_{i=0}^{n-1} i x^{i-1} = \frac{1-nx^{n-1}+(n-1)x^n}{(1-x)^2}.$$

Incidentally, Problem 14.2 shows how the perturbation method could also be applied to derive this formula.

Often, differentiating or integrating messes up the exponent of x in every term. In this case, we now have a formula for a sum of the form $\sum i x^{i-1}$, but we want a formula for the series $\sum i x^i$. The solution is simple: multiply by x. This gives:

$$\sum_{i=1}^{n-1} i x^i = \frac{x-nx^n+(n-1)x^{n+1}}{(1-x)^2} \tag{14.12}$$

and we have the desired closed-form expression for our sum[3]. It's a little complicated looking, but it's easier to work with than the sum.

Notice that if $|x| < 1$, then this series converges to a finite value even if there are infinitely many terms. Taking the limit of equation 14.12 as n tends infinity gives the following theorem:

[3]Since we could easily have made a mistake in the calculation, it is always a good idea to go back and validate a formula obtained this way with a proof by induction.

Theorem 14.1.2. *If* $|x| < 1$, *then*

$$\sum_{i=1}^{\infty} ix^i = \frac{x}{(1-x)^2}.$$

As a consequence, suppose that there is an annuity that pays im dollars at the *end* of each year i forever. For example, if $m = \$50{,}000$, then the payouts are \$50,000 and then \$100,000 and then \$150,000 and so on. It is hard to believe that the value of this annuity is finite! But we can use Theorem 14.1.2 to compute the value:

annuity w/ growth

$$\begin{aligned} V &= \sum_{i=1}^{\infty} \frac{im}{(1+p)^i} \\ &= m \cdot \frac{1/(1+p)}{(1-\frac{1}{1+p})^2} \\ &= m \cdot \frac{1+p}{p^2}. \end{aligned}$$

The second line follows by an application of Theorem 14.1.2. The third line is obtained by multiplying the numerator and denominator by $(1+p)^2$.

For example, if $m = \$50{,}000$, and $p = 0.08$ as usual, then the value of the annuity is $V = \$8{,}437{,}500$. Even though the payments increase every year, the increase is only additive with time; by contrast, dollars paid out in the future decrease in value exponentially with time. The geometric decrease swamps out the additive increase. Payments in the distant future are almost worthless, so the value of the annuity is finite.

The important thing to remember is the trick of taking the derivative (or integral) of a summation formula. Of course, this technique requires one to compute nasty derivatives correctly, but this is at least theoretically possible!

14.2 Sums of Powers

In Chapter 6, we verified the formula (14.1), but the source of this formula is still a mystery. Sure, we can prove it is true using well ordering or induction, but where did the expression on the right come from in the first place? Even more inexplicable is the closed form expression for the sum of consecutive squares:

$$\sum_{i=1}^{n} i^2 = \frac{(2n+1)(n+1)n}{6}. \tag{14.13}$$

It turns out that there is a way to derive these expressions, but before we explain it, we thought it would be fun[4] to show you how Gauss is supposed to have proved equation 14.1 when he was a young boy.

Gauss's idea is related to the perturbation method we used in Section 14.1.2. Let

$$S = \sum_{i=1}^{n} i.$$

Then we can write the sum in two orders:

$$\begin{aligned} S &= 1 + \quad 2 \quad + \ldots + (n-1) + n, \\ S &= n + (n-1) + \ldots + \quad 2 \quad + 1. \end{aligned}$$

Adding these two equations gives

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

Hence,

$$S = \frac{n(n+1)}{2}.$$

Not bad for a young child —Gauss showed some potential....

Unfortunately, the same trick does not work for summing consecutive squares. However, we can observe that the result might be a third-degree polynomial in n, since the sum contains n terms that average out to a value that grows quadratically in n. So we might guess that

$$\sum_{i=1}^{n} i^2 = an^3 + bn^2 + cn + d.$$

If the guess is correct, then we can determine the parameters a, b, c, and d by plugging in a few values for n. Each such value gives a linear equation in a, b, c, and d. If we plug in enough values, we may get a linear system with a unique solution. Applying this method to our example gives:

$$\begin{aligned} n = 0 \quad &\text{implies} \quad 0 = d \\ n = 1 \quad &\text{implies} \quad 1 = a + b + c + d \\ n = 2 \quad &\text{implies} \quad 5 = 8a + 4b + 2c + d \\ n = 3 \quad &\text{implies} \quad 14 = 27a + 9b + 3c + d. \end{aligned}$$

[4] OK, our definition of "fun" may be different than yours.

Solving this system gives the solution $a = 1/3$, $b = 1/2$, $c = 1/6$, $d = 0$. Therefore, *if* our initial guess at the form of the solution was correct, then the summation is equal to $n^3/3 + n^2/2 + n/6$, which matches equation 14.13.

The point is that if the desired formula turns out to be a polynomial, then once you get an estimate of the *degree* of the polynomial, all the coefficients of the polynomial can be found automatically.

Be careful! This method let's you discover formulas, but it doesn't guarantee they are right! After obtaining a formula by this method, it's important to go back and *prove* it using induction or some other method, because if the initial guess at the solution was not of the right form, then the resulting formula will be completely wrong! A later chapter will describe a method based on generating functions that does not require any guessing at all.

14.3 Approximating Sums

Unfortunately, it is not always possible to find a closed-form expression for a sum. For example, consider the sum

$$S = \sum_{i=1}^{n} \sqrt{i}.$$

No closed form expression is known for S.

In such cases, we need to resort to approximations for S if we want to have a closed form. The good news is that there is a general method to find closed-form upper and lower bounds that work for most any sum. Even better, the method is simple and easy to remember. It works by replacing the sum by an integral and then adding either the first or last term in the sum.

Definition 14.3.1. A function $f : \mathbb{R}^+ \to \mathbb{R}^+$ is *strictly increasing* when

$$x < y \text{ IMPLIES } f(x) < f(y),$$

and it is *weakly increasing*[5] when

$$x < y \text{ IMPLIES } f(x) \leq f(y).$$

[5]Weakly increasing functions are usually called *nondecreasing* functions. We will avoid this terminology to prevent confusion between being a nondecreasing function and the much weaker property of *not* being a decreasing function.

Similarly, f is *strictly decreasing* when

$$x < y \text{ IMPLIES } f(x) > f(y),$$

and it is *weakly decreasing*[6] when

$$x < y \text{ IMPLIES } f(x) \geq f(y).$$

For example, 2^x and $\sqrt{x}$ are strictly increasing functions, while $\max x, 2$ and $\lceil x \rceil$ are weakly increasing functions. The functions $1/x$ and 2^{-x} are strictly decreasing, while $\min 1/x, 1/2$ and $\lfloor 1/x \rfloor$ are weakly decreasing.

Theorem 14.3.2. *Let $f : \mathbb{R}^+ \to \mathbb{R}^+$ be a weakly increasing function. Define*

$$S ::= \sum_{i=1}^{n} f(i) \tag{14.14}$$

and

$$I ::= \int_1^n f(x)\,dx.$$

Then

$$I + f(1) \leq S \leq I + f(n). \tag{14.15}$$

Similarly, if f is weakly decreasing, then

$$I + f(n) \leq S \leq I + f(1).$$

Proof. Suppose $f : \mathbb{R}^+ \to \mathbb{R}^+$ is weakly increasing. The value of the sum S in (14.14) is the sum of the areas of n unit-width rectangles of heights $f(0), f(1), \ldots, f(n)$. This area of these rectangles is shown shaded in Figure 14.1.

The value of

$$I = \int_1^n f(x)\,dx$$

is the shaded area under the curve of $f(x)$ from 1 to n shown in Figure 14.2.

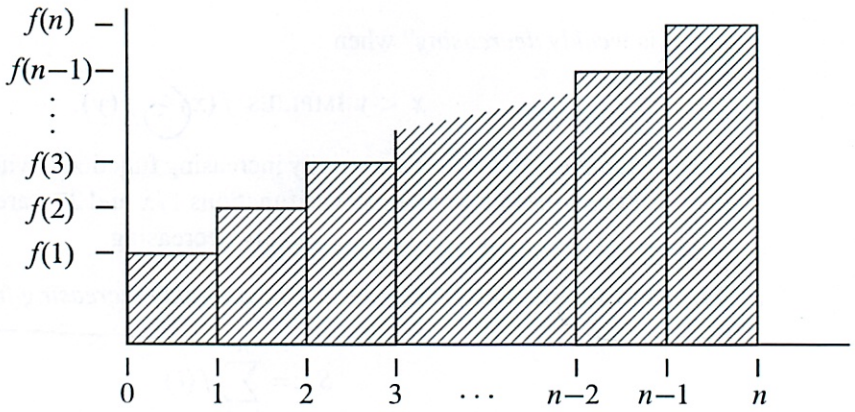
Comparing the shaded regions in Figures 14.1 and 14.2 shows that S is at least I plus the area of the leftmost rectangle. Hence,

$$S \geq I + f(1) \tag{14.16}$$

This is the lower bound for S given in (14.15).

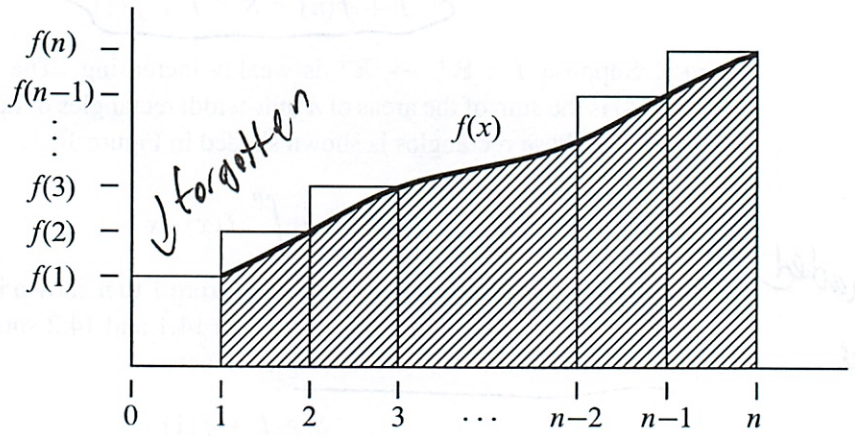
To derive the upper bound for S given in (14.15), we shift the curve of $f(x)$ from 1 to n one unit to the left as shown in Figure 14.3.

[6] Weakly decreasing functions are usually called *nonincreasing*.



Shaded = sum

Figure 14.1 The area of the ith rectangle is $f(i)$. The shaded region has area $\sum_{i=1}^{n} f(i)$.



Shaded = integral

Figure 14.2 The shaded area under the curve of $f(x)$ from 1 to n (shown in bold) is $I = \int_1^n f(x)\,dx$.

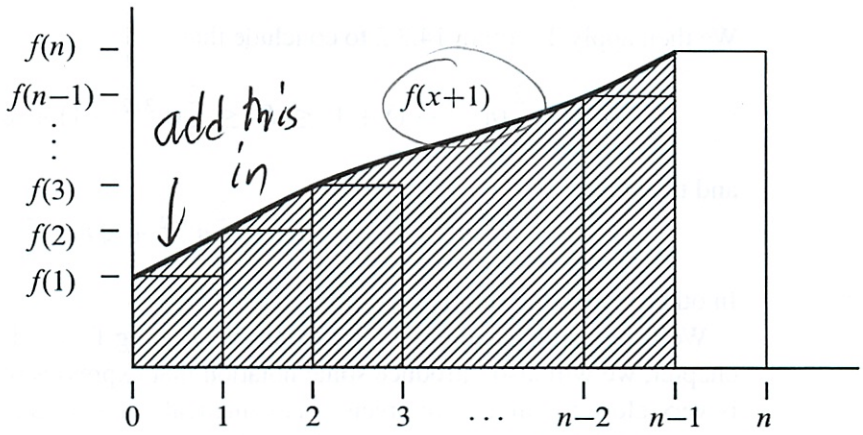


Figure 14.3 This curve is the same as the curve in Figure 14.2 shifted left by 1.

Comparing the shaded regions in Figures 14.1 and 14.3 shows that S is at most I plus the area of the rightmost rectangle. That is,

$$S \leq I + f(n),$$

which is the upper bound for S given in (14.15).

The very similar argument for the weakly decreasing case is left to Problem 14.6. ■

Theorem 14.3.2 provides good bounds for most sums. At worst, the bounds will be off by the largest term in the sum. For example, we can use Theorem 14.3.2 to bound the sum

$$S = \sum_{i=1}^{n} \sqrt{i}$$

as follows.

We begin by computing

$$\begin{aligned} I &= \int_1^n \sqrt{x}\, dx \\ &= \frac{x^{3/2}}{3/2}\Bigg|_1^n \\ &= \frac{2}{3}(n^{3/2} - 1). \end{aligned}$$

We then apply Theorem 14.3.2 to conclude that

$$\frac{2}{3}(n^{3/2}-1)+1 \le S \le \frac{2}{3}(n^{3/2}-1)+\sqrt{n}$$

and thus that

$$\frac{2}{3}n^{3/2}+\frac{1}{3} \le S \le \frac{2}{3}n^{3/2}+\sqrt{n}-\frac{2}{3}.$$

In other words, the sum is very close to $\frac{2}{3}n^{3/2}$.

We'll be using Theorem 14.3.2 extensively going forward. At the end of this chapter, we will also introduce some notation that expresses phrases like "the sum is very close to" in a more precise mathematical manner. But first, we'll see how Theorem 14.3.2 can be used to resolve a classic paradox in structural engineering.

14.4 Hanging Out Over the Edge

Suppose we have n identical unit length rectangular blocks that are uniformly weighted. We want to stack them one on top of the next on a table as shown in Figure 14.4. Is there some value of n for which it is possible to arrange the stack so that one of the blocks hangs out completely over the edge of the table without having the stack fall over? (You are not allowed to use glue or otherwise hold the stack in position.)

Most people's first response to this question—sometimes also their second and third responses—is "No. No block will ever get completely past the edge of the table." But in fact, if n is large enough, you can get the top block to stick out as far as you want: one block-length, two block-lengths, any number of block-lengths!

14.4.1 Stability

A stack of blocks is said to be *stable* if it will not fall over of its own accord. For example, the stack illustrated in Figure 14.4 is not stable because the top block is sure to fall over. This is because the center or mass of the top block is hanging out over air.

In general, a stack of n blocks will be stable if and only if the center of mass of the top i blocks sits over the $(i+1)$st block for $i = 1, 2, \ldots, n-1$, and over the table for $i = n$.

We define the *overhang* of a stable stack to be the distance between the edge of the table and the rightmost end of the rightmost block in the stack. Our goal is thus to maximize the overhang of a stable stack.

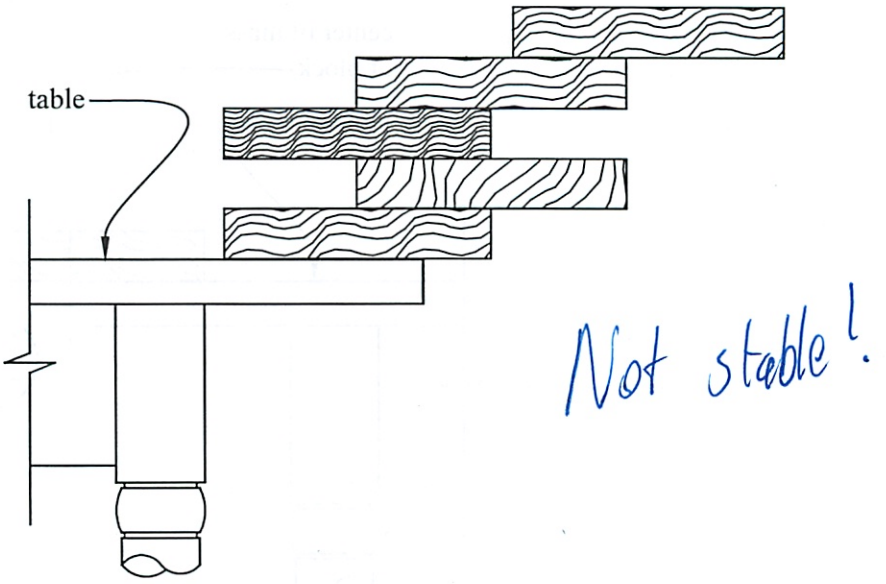


Figure 14.4 A stack of 5 identical blocks on a table. The top block is hanging out over the edge of the table, but if you try stacking the blocks this way, the stack will fall over.

For example, the maximum possible overhang for a single block is $1/2$. That is because the center of mass of a single block is in the middle of the block (which is distance $1/2$ from the right edge of the block). If we were to place the block so that its right edge is more than $1/2$ from the edge of the table, the center of mass would be over air and the block would tip over. But we can place the block so the center of mass is at the edge of the table, thereby achieving overhang $1/2$. This position is illustrated in Figure 14.5.

In general, the overhang of a stack of blocks is maximized by sliding the entire stack rightward until its center of mass is at the edge of the table. The overhang will then be equal to the distance between the center of mass of the stack and the rightmost edge of the rightmost block. We call this distance the *spread* of the stack. Note that the spread does not depend on the location of the stack on the table—it is purely a property of the blocks in the stack. Of course, as we just observed, the maximum possible overhang is equal to the maximum possible spread. This relationship is illustrated in Figure 14.6.

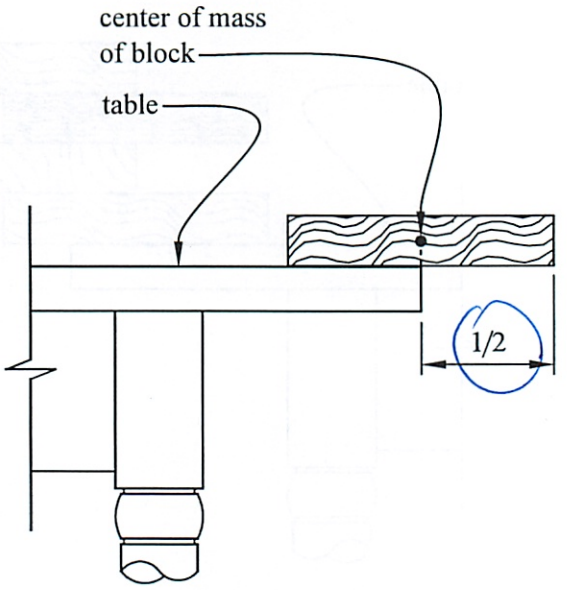


Figure 14.5 One block can overhang half a block length.

14.4.2 A Recursive Solution

Our goal is to find a formula for the maximum possible spread S_n that is achievable with a stable stack of n blocks.

We already know that $S_1 = 1/2$ since the right edge of a single block with length 1 is always distance $1/2$ from its center of mass. Let's see if we can use a recursive approach to determine S_n for all n. This means that we need to find a formula for S_n in terms of S_i where $i < n$.

Suppose we have a stable stack $\mathcal{S}$ of n blocks with maximum possible spread S_n. There are two cases to consider depending on where the rightmost block is in the stack.

Case 1: *The rightmost block in $\mathcal{S}$ is the bottom block.* Since the center of mass of the top $n-1$ blocks must be over the bottom block for stability, the spread is maximized by having the center of mass of the top $n-1$ blocks be directly over the *left* edge of the bottom block. In this case the center of mass of $\mathcal{S}$ is[7]

$$\frac{(n-1)\cdot 1 + (1)\cdot \frac{1}{2}}{n} = 1 - \frac{1}{2n}$$

[7]The center of mass of a stack of blocks is the average of the centers of mass of the individual blocks.

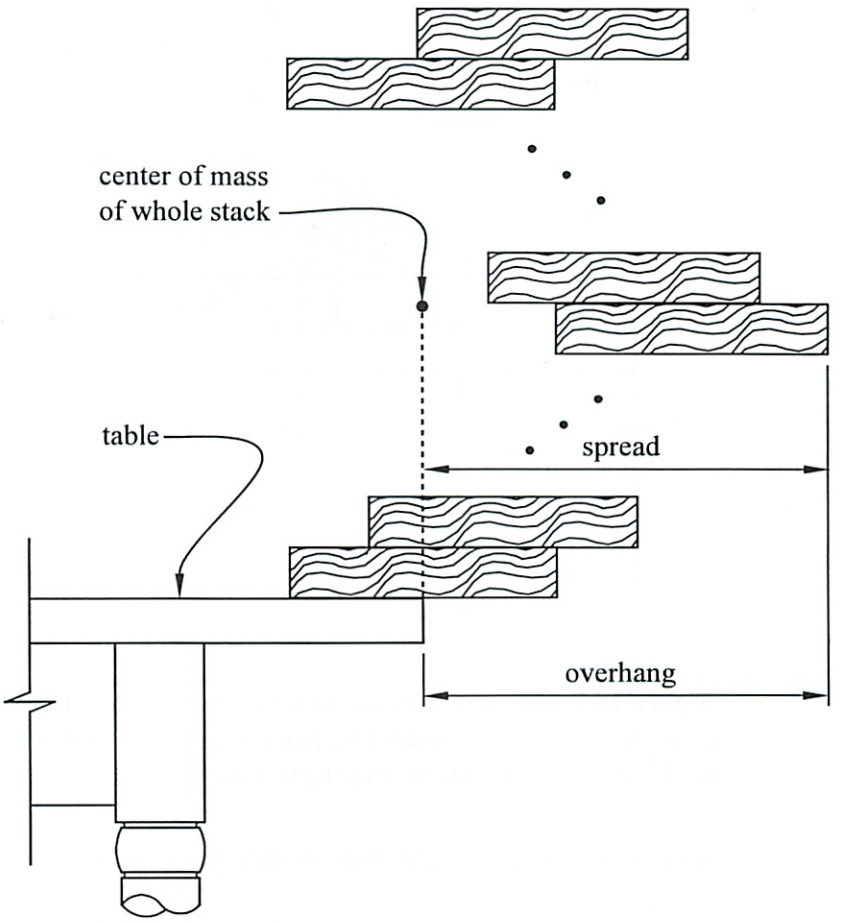


Figure 14.6 The overhang is maximized by maximizing the spread and then placing the stack so that the center of mass is at the edge of the table.

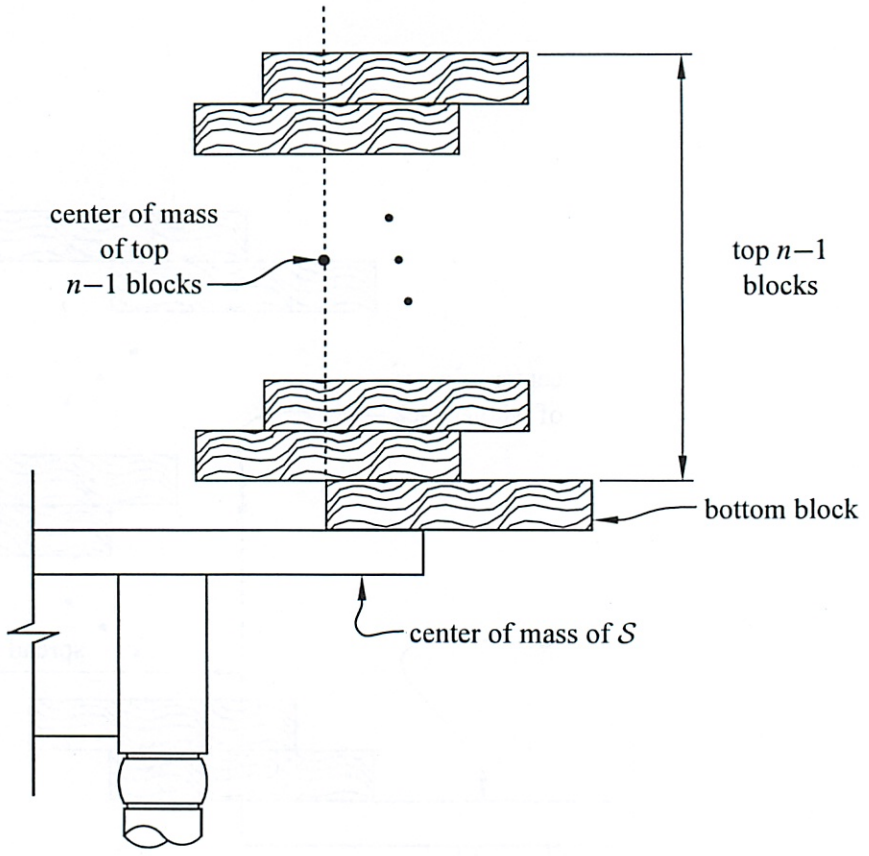


Figure 14.7 The scenario where the bottom block is the rightmost block. In this case, the spread is maximized by having the center of mass of the top $n-1$ blocks be directly over the left edge of the bottom block.

to the left of the right edge of the bottom block and so the spread for S is

$$1 - \frac{1}{2n}. \tag{14.17}$$

For example, see Figure 14.7.

In fact, the scenario just described is easily achieved by arranging the blocks as shown in Figure 14.8, in which case we have the spread given by equation 14.17. For example, the spread is $3/4$ for 2 blocks, $5/6$ for 3 blocks, $7/8$ for 4 blocks, etc.

Can we do any better? The best spread in Case 1 is always less than 1, which means that we cannot get a block fully out over the edge of the table in this scenario. Maybe our intuition was right that we can't do better. Before we jump to any false conclusions, however, let's see what happens in the other case.

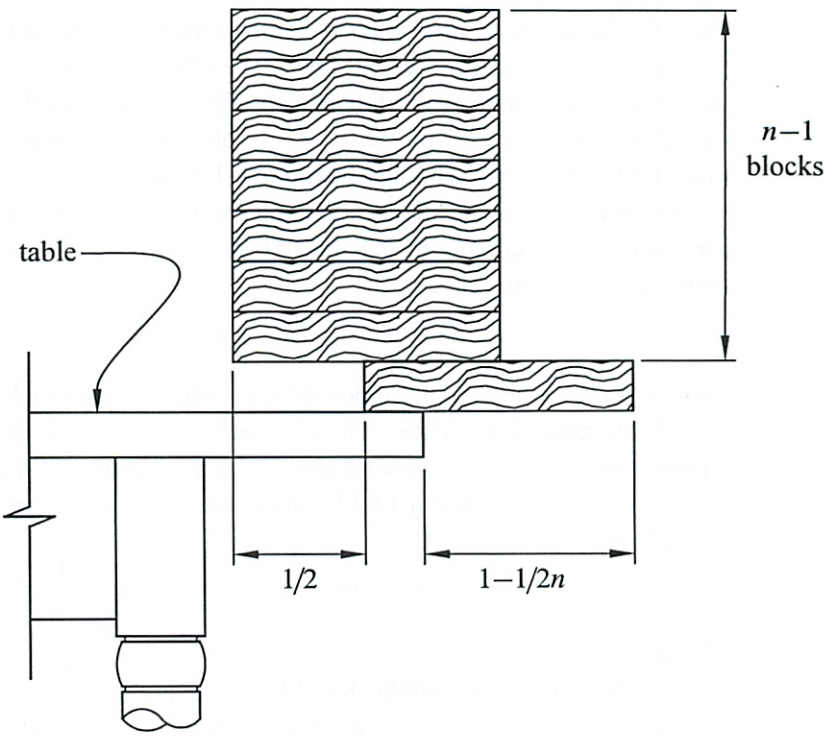


Figure 14.8 A method for achieving spread (and hence overhang) $1 - 1/2n$ with n blocks, where the bottom block is the rightmost block.

Case 2: *The rightmost block in $\mathcal{S}$ is among the top $n-1$ blocks.* In this case, the spread is maximized by placing the top $n-1$ blocks so that their center of mass is directly over the *right* end of the bottom block. This means that the center of mass for $\mathcal{S}$ is at location

$$\frac{(n-1)\cdot C + 1\cdot\left(C-\frac{1}{2}\right)}{n} = C - \frac{1}{2n}$$

where C is the location of the center of mass of the top $n-1$ blocks. In other words, the center of mass of $\mathcal{S}$ is $1/2n$ to the left of the center of mass of the top $n-1$ blocks. (The difference is due to the effect of the bottom block, whose center of mass is $1/2$ unit to the left of C.) This means that the spread of $\mathcal{S}$ is $1/2n$ greater than the spread of the top $n-1$ blocks (because we are in the case where the rightmost block is among the top $n-1$ blocks.)

Since the rightmost block is among the top $n-1$ blocks, the spread for $\mathcal{S}$ is maximized by maximizing the spread for the top $n-1$ blocks. Hence the maximum spread for $\mathcal{S}$ in this case is

$$S_{n-1} + \frac{1}{2n} \tag{14.18}$$

where S_{n-1} is the maximum possible spread for $n-1$ blocks (using any strategy).

We are now almost done. There are only two cases to consider when designing a stack with maximum spread and we have analyzed both of them. This means that we can combine equation 14.17 from Case 1 with equation 14.18 from Case 2 to conclude that

$$S_n = \max\left\{1 - \frac{1}{2n},\ S_{n-1} + \frac{1}{2n}\right\} \tag{14.19}$$

for any $n > 1$.

Uh-oh. This looks complicated. Maybe we are not almost done after all!

Equation 14.19 is an example of a *recurrence*. We will describe numerous techniques for solving recurrences in a later Chapter, but, fortunately, equation 14.19 is simple enough that we can solve it directly.

One of the first things to do when you have a recurrence is to get a feel for it by computing the first few terms. This often gives clues about a way to solve the recurrence, as it will in this case.

We already know that $S_1 = 1/2$. What about S_2? From equation 14.19, we find that

$$\begin{aligned} S_2 &= \max\left\{1 - \frac{1}{4},\ \frac{1}{2} + \frac{1}{4}\right\} \\ &= 3/4. \end{aligned}$$

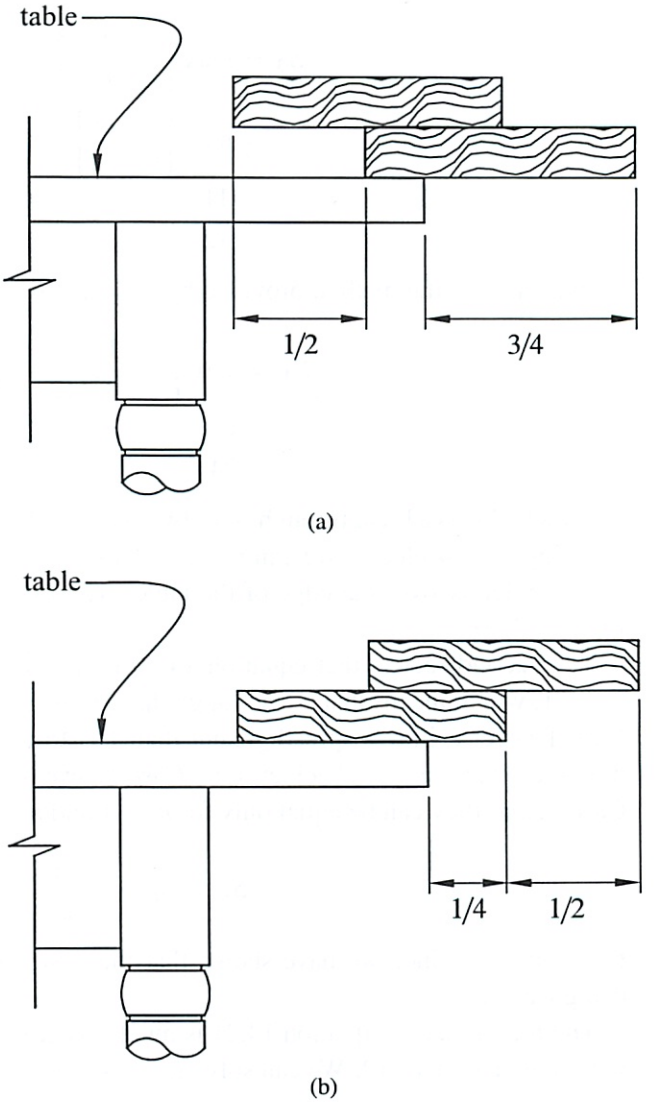


Figure 14.9 Two ways to achieve spread (and hence overhang) $3/4$ with $n = 2$ blocks. The first way (a) is from Case 1 and the second (b) is from Case 2.

Both cases give the same spread, albeit by different approaches. For example, see Figure 14.9.

That was easy enough. What about S_3?

$$\begin{aligned} S_3 &= \max\left\{1-\frac{1}{6}, \frac{3}{4}+\frac{1}{6}\right\} \\ &= \max\left\{\frac{5}{6}, \frac{11}{12}\right\} \\ &= \frac{11}{12}. \end{aligned}$$

As we can see, the method provided by Case 2 is the best. Let's check $n = 4$.

$$\begin{aligned} S_4 &= \max\left\{1-\frac{1}{8}, \frac{11}{12}+\frac{1}{8}\right\} \\ &= \frac{25}{24}. \end{aligned} \tag{14.20}$$

Wow! This is a breakthrough—for two reasons. First, equation 14.20 tells us that by using only 4 blocks, we can make a stack so that one of the blocks is hanging out completely over the edge of the table. The two ways to do this are shown in Figure 14.10.

The second reason that equation 14.20 is important is that we now know that $S_4 > 1$, which means that we no longer have to worry about Case 1 for $n > 4$ since Case 1 never achieves spread greater than 1. Moreover, even for $n \leq 4$, we have now seen that the spread achieved by Case 1 never exceeds the spread achieved by Case 2, and they can be equal only for $n = 1$ and $n = 2$. This means that

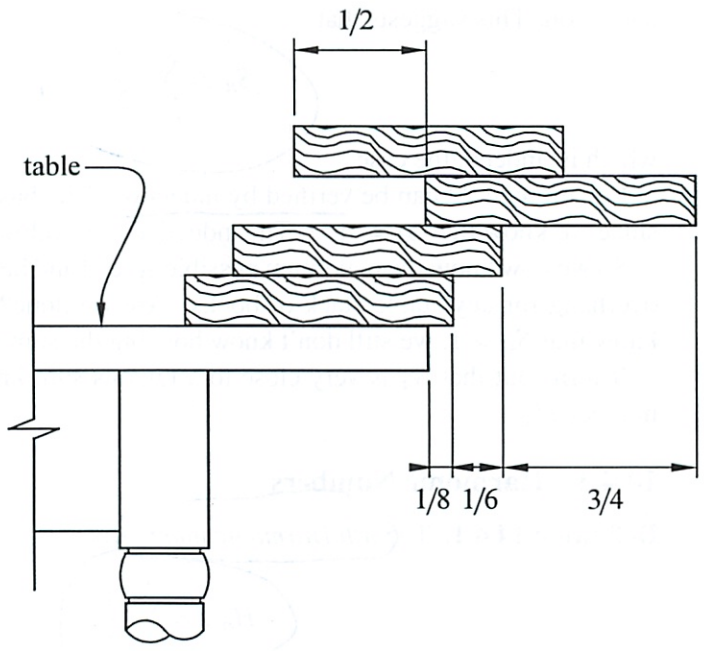
$$S_n = S_{n-1} + \frac{1}{2n} \tag{14.21}$$

for all $n > 1$ since we have shown that the best spread can always be achieved using Case 2.

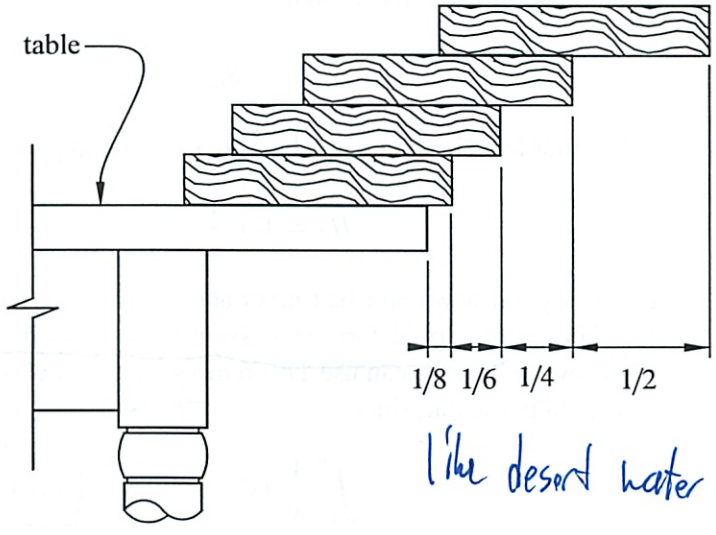
The recurrence in equation 14.21 is much easier to solve than the one we started with in equation 14.19. We can solve it by expanding the equation as follows:

$$\begin{aligned} S_n &= S_{n-1} + \frac{1}{2n} \\ &= S_{n-2} + \frac{1}{2(n-1)} + \frac{1}{2n} \\ &= S_{n-3} + \frac{1}{2(n-2)} + \frac{1}{2(n-1)} + \frac{1}{2n} \end{aligned}$$

(This is not what we did in class



(a)



(b)

Figure 14.10 The two ways to achieve spread (and overhang) 25/24. The method in (a) uses Case 1 for the top 2 blocks and Case 2 for the others. The method in (b) uses Case 2 for every block that is added to the stack.

and so on. This suggests that

$$S_n = \sum_{i=1}^{n} \frac{1}{2i}, \tag{14.22}$$

which is, indeed, the case.

Equation 14.22 can be verified by induction. The base case when $n = 1$ is true since we know that $S_1 = 1/2$. The inductive step follows from equation 14.21.

So we now know the maximum possible spread and hence the maximum possible overhang for any stable stack of books. Are we done? Not quite. Although we know that $S_4 > 1$, we still don't know how big the sum $\sum_{i=1}^{n} \frac{1}{2i}$ can get.

It turns out that S_n is very close to a famous sum known as the nth Harmonic number H_n.

14.4.3 Harmonic Numbers

Definition 14.4.1. The *nth Harmonic number* is

$$H_n ::= \sum_{i=1}^{n} \frac{1}{i}.$$

So equation 14.22 means that

$$S_n = \frac{H_n}{2}. \tag{14.23}$$

The first few Harmonic numbers are easy to compute. For example,

$$H_4 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{25}{12}.$$

There is good news and bad news about Harmonic numbers. The bad news is that there is no closed-form expression known for the Harmonic numbers. The good news is that we can use Theorem 14.3.2 to get close upper and lower bounds on H_n. In particular, since

$$\int_1^n \frac{1}{x}\,dx = \ln(x)\Big|_1^n = \ln(n),$$

Theorem 14.3.2 means that

$$\ln(n) + \frac{1}{n} \leq H_n \leq \ln(n) + 1. \tag{14.24}$$

In other words, the nth Harmonic number is very close to $\ln(n)$.

Because the Harmonic numbers frequently arise in practice, mathematicians have worked hard to get even better approximations for them. In fact, it is now known that

$$H_n = \ln(n) + \gamma + \frac{1}{2n} + \frac{1}{12n^2} + \frac{\epsilon(n)}{120n^4} \tag{14.25}$$

Here γ is a value $0.577215664\ldots$ called *Euler's constant*, and $\epsilon(n)$ is between 0 and 1 for all n. We will not prove this formula.

We are now finally done with our analysis of the block stacking problem. Plugging the value of H_n into equation 14.23, we find that the maximum overhang for n blocks is very close to $\frac{1}{2}\ln(n)$. Since $\ln(n)$ grows to infinity as n increases, this means that if we are given enough blocks (in theory anyway), we can get a block to hang out arbitrarily far over the edge of the table. Of course, the number of blocks we need will grow as an exponential function of the overhang, so it will probably take you a long time to achieve an overhang of 2 or 3, never mind an overhang of 100.

14.4.4 Asymptotic Equality

For cases like equation 14.25 where we understand the growth of a function like H_n up to some (unimportant) error terms, we use a special notation, $\sim$, to denote the leading term of the function. For example, we say that $H_n \sim \ln(n)$ to indicate that the leading term of H_n is $\ln(n)$. More precisely:

Definition 14.4.2. For functions $f, g : \mathbb{R} \to \mathbb{R}$, we say f is *asymptotically equal* to g, in symbols,

$$f(x) \sim g(x)$$

iff

$$\lim_{x\to\infty} f(x)/g(x) = 1.$$

Although it is tempting to write $H_n \sim \ln(n) + \gamma$ to indicate the two leading terms, this is not really right. According to Definition 14.4.2, $H_n \sim \ln(n) + c$ where c is *any constant*. The correct way to indicate that γ is the second-largest term is $H_n - \ln(n) \sim \gamma$.

The reason that the $\sim$ notation is useful is that often we do not care about lower order terms. For example, if $n = 100$, then we can compute $H(n)$ to great precision using only the two leading terms:

$$|H_n - \ln(n) - \gamma| \leq \left|\frac{1}{200} - \frac{1}{120000} + \frac{1}{120 \cdot 100^4}\right| < \frac{1}{200}.$$

We will spend a lot more time talking about asymptotic notation at the end of the chapter. But for now, let's get back to sums.

14.5 Products

We've covered several techniques for finding closed forms for sums but no methods for dealing with products. Fortunately, we do not need to develop an entirely new set of tools when we encounter a product such as

$$n! ::= \prod_{i=1}^{n} i. \tag{14.26}$$

That's because we can convert any product into a sum by taking a logarithm. For example, if

$$P = \prod_{i=1}^{n} f(i),$$

then

$$\ln(P) = \sum_{i=1}^{n} \ln(f(i)).$$

We can then apply our summing tools to find a closed form (or approximate closed form) for $\ln(P)$ and then exponentiate at the end to undo the logarithm.

For example, let's see how this works for the factorial function $n!$ We start by taking the logarithm:

$$\begin{aligned} \ln(n!) &= \ln(1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n) \\ &= \ln(1) + \ln(2) + \ln(3) + \cdots + \ln(n-1) + \ln(n) \\ &= \sum_{i=1}^{n} \ln(i). \end{aligned}$$

Unfortunately, no closed form for this sum is known. However, we can apply Theorem 14.3.2 to find good closed-form bounds on the sum. To do this, we first compute

$$\begin{aligned} \int_1^n \ln(x)\,dx &= x\ln(x) - x\Big|_1^n \\ &= n\ln(n) - n + 1. \end{aligned}$$

Plugging into Theorem 14.3.2, this means that

$$n\ln(n) - n + 1 \le \sum_{i=1}^{n} \ln(i) \le n\ln(n) - n + 1 + \ln(n).$$

Exponentiating then gives

$$\frac{n^n}{e^{n-1}} \leq n! \leq \frac{n^{n+1}}{e^{n-1}}. \tag{14.27}$$

This means that $n!$ is within a factor of n of n^n/e^{n-1}.

14.5.1 Stirling's Formula

$n!$ is probably the most commonly used product in discrete mathematics, and so mathematicians have put in the effort to find much better closed-form bounds on its value. The most useful bounds are given in Theorem 14.5.1.

Theorem 14.5.1 (Stirling's Formula). *For all $n \geq 1$,*

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\epsilon(n)}$$

where

$$\frac{1}{12n+1} \leq \epsilon(n) \leq \frac{1}{12n}.$$

Theorem 14.5.1 can be proved by induction on n, but the details are a bit painful (even for us) and so we will not go through them here.

There are several important things to notice about Stirling's Formula. First, $\epsilon(n)$ is always positive. This means that

$$n! > \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{14.28}$$

for all $n \in \mathbb{N}^+$.

Second, $\epsilon(n)$ tends to zero as n gets large. This means that

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \tag{14.29}$$

which is rather surprising. After all, who would expect both π and e to show up in a closed-form expression that is asymptotically equal to $n!$?

Third, $\epsilon(n)$ is small even for small values of n. This means that Stirling's Formula provides good approximations for $n!$ for most all values of n. For example, if we use

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

as the approximation for $n!$, as many people do, we are guaranteed to be within a factor of

$$e^{\epsilon(n)} \leq e^{\frac{1}{12n}}$$

Approximation	$n \geq 1$	$n \geq 10$	$n \geq 100$	$n \geq 1000$
$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$	$< 10\%$	$< 1\%$	$< 0.1\%$	$< 0.01\%$
$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{1/12n}$	$< 1\%$	$< 0.01\%$	$< 0.0001\%$	$< 0.000001\%$

Table 14.1 Error bounds on common approximations for $n!$ from Theorem 14.5.1. For example, if $n \geq 100$, then $\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ approximates $n!$ to within 0.1%.

of the correct value. For $n \geq 10$, this means we will be within 1% of the correct value. For $n \geq 100$, the error will be less than 0.1%.

If we need an even closer approximation for $n!$, then we could use either

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{1/12n}$$

or

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{1/(12n+1)}$$

depending on whether we want an upper bound or a lower bound, respectively. By Theorem 14.5.1, we know that both bounds will be within a factor of

$$e^{\frac{1}{12n}-\frac{1}{12n+1}} = e^{\frac{1}{144n^2+12n}}$$

of the correct value. For $n \geq 10$, this means that either bound will be within 0.01% of the correct value. For $n \geq 100$, the error will be less than 0.0001%.

For quick future reference, these facts are summarized in Corollary 14.5.2 and Table 14.1.

Corollary 14.5.2. *For $n \geq 1$,*

$$n! < 1.09\sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

For $n \geq 10$,

$$n! < 1.009\sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

For $n \geq 100$,

$$n! < 1.0009\sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

14.6 Double Trouble

Sometimes we have to evaluate sums of sums, otherwise known as *double summations*. This sounds hairy, and sometimes it is. But usually, it is straightforward—you just evaluate the inner sum, replace it with a closed form, and then evaluate the outer sum (which no longer has a summation inside it). For example,[8]

$$\sum_{n=0}^{\infty}\left(y^n\sum_{i=0}^{n}x^i\right)=\sum_{n=0}^{\infty}\left(y^n\frac{1-x^{n+1}}{1-x}\right) \qquad \text{equation 14.2}$$

$$=\left(\frac{1}{1-x}\right)\sum_{n=0}^{\infty}y^n-\left(\frac{1}{1-x}\right)\sum_{n=0}^{\infty}y^nx^{n+1}$$

$$=\frac{1}{(1-x)(1-y)}-\left(\frac{x}{1-x}\right)\sum_{n=0}^{\infty}(xy)^n \qquad \text{Theorem 14.1.1}$$

$$=\frac{1}{(1-x)(1-y)}-\frac{x}{(1-x)(1-xy)} \qquad \text{Theorem 14.1.1}$$

$$=\frac{(1-xy)-x(1-y)}{(1-x)(1-y)(1-xy)}$$

$$=\frac{1-x}{(1-x)(1-y)(1-xy)}$$

$$=\frac{1}{(1-y)(1-xy)}.$$

When there's no obvious closed form for the inner sum, a special trick that is often useful is to try *exchanging the order of summation*. For example, suppose we want to compute the sum of the first n Harmonic numbers

$$\sum_{k=1}^{n}H_k=\sum_{k=1}^{n}\sum_{j=1}^{k}\frac{1}{j} \qquad (14.30)$$

For intuition about this sum, we can apply Theorem 14.3.2 to equation 14.24 to conclude that the sum is close to

$$\int_1^n \ln(x)\,dx = x\ln(x)-x\Big|_1^n = n\ln(n)-n+1.$$

[8] Ok, so maybe this one is a little hairy, but it is also fairly straightforward. Wait till you see the next one!

Now let's look for an exact answer. If we think about the pairs (k, j) over which we are summing, they form a triangle:

		j						
		1	2	3	4	5	...	n
k	1	1						
	2	1	1/2					
	3	1	1/2	1/3				
	4	1	1/2	1/3	1/4			
		...						
	n	1	1/2		...			$1/n$

The summation in equation 14.30 is summing each row and then adding the row sums. Instead, we can sum the columns and then add the column sums. Inspecting the table we see that this double sum can be written as

$$\begin{aligned}
\sum_{k=1}^{n} H_k &= \sum_{k=1}^{n}\sum_{j=1}^{k} \frac{1}{j} \\
&= \sum_{j=1}^{n}\sum_{k=j}^{n} \frac{1}{j} \\
&= \sum_{j=1}^{n} \frac{1}{j}\sum_{k=j}^{n} 1 \\
&= \sum_{j=1}^{n} \frac{1}{j}(n-j+1) \\
&= \sum_{j=1}^{n} \frac{n+1}{j} - \sum_{j=1}^{n} \frac{j}{j} \\
&= (n+1)\sum_{j=1}^{n} \frac{1}{j} - \sum_{j=1}^{n} 1 \\
&= (n+1)H_n - n. \qquad (14.31)
\end{aligned}$$

14.7 Asymptotic Notation

Asymptotic notation is a shorthand used to give a quick measure of the behavior of a function $f(n)$ as n grows large. For example, the asymptotic notation $\sim$ of Definition 14.4.2 is a binary relation indicating that two functions grow at the *same* rate. There is also a binary relation indicating that one function grows at a significantly *slower* rate than another.

14.7.1 Little Oh

Definition 14.7.1. For functions $f, g : \mathbb{R} \to \mathbb{R}$, with g nonnegative, we say f is *asymptotically smaller* than g, in symbols,

$$f(x) = o(g(x)),$$

iff

$$\lim_{x\to\infty} f(x)/g(x) = 0.$$

For example, $1000x^{1.9} = o(x^2)$, because $1000x^{1.9}/x^2 = 1000/x^{0.1}$ and since $x^{0.1}$ goes to infinity with x and 1000 is constant, we have $\lim_{x\to\infty} 1000x^{1.9}/x^2 = 0$. This argument generalizes directly to yield

Lemma 14.7.2. *$x^a = o(x^b)$ for all nonnegative constants $a < b$.*

Using the familiar fact that $\log x < x$ for all $x > 1$, we can prove

Lemma 14.7.3. *$\log x = o(x^\epsilon)$ for all $\epsilon > 0$.*

Proof. Choose $\epsilon > \delta > 0$ and let $x = z^\delta$ in the inequality $\log x < x$. This implies

$$\log z < z^\delta/\delta = o(z^\epsilon) \qquad \text{by Lemma 14.7.2.} \tag{14.32}$$

■

Corollary 14.7.4. *$x^b = o(a^x)$ for any $a, b \in \mathbb{R}$ with $a > 1$.*

Lemma 14.7.3 and Corollary 14.7.4 can also be proved using l'Hôpital's Rule or the McLaurin Series for $\log x$ and e^x. Proofs can be found in most calculus texts.

14.7.2 Big Oh

Big Oh is the most frequently used asymptotic notation. It is used to give an upper bound on the growth of a function, such as the running time of an algorithm.

Definition 14.7.5. Given nonnegative functions $f, g : \mathbb{R} \to \mathbb{R}$, we say that

$$f = O(g)$$

iff

$$\limsup_{x\to\infty} f(x)/g(x) < \infty.$$

This definition[9] makes it clear that

Lemma 14.7.6. *If $f = o(g)$ or $f \sim g$, then $f = O(g)$.*

Proof. $\lim f/g = 0$ or $\lim f/g = 1$ implies $\lim f/g < \infty$. ∎

It is easy to see that the converse of Lemma 14.7.6 is not true. For example, $2x = O(x)$, but $2x \not\sim x$ and $2x \neq o(x)$.

The usual formulation of Big Oh spells out the definition of lim sup without mentioning it. Namely, here is an equivalent definition:

Definition 14.7.7. Given functions $f, g : \mathbb{R} \to \mathbb{R}$, we say that

$$f = O(g)$$

iff there exists a constant $c \geq 0$ and an x_0 such that for all $x \geq x_0$, $|f(x)| \leq cg(x)$.

This definition is rather complicated, but the idea is simple: $f(x) = O(g(x))$ means $f(x)$ is less than or equal to $g(x)$, except that we're willing to ignore a constant factor, namely, c, and to allow exceptions for small x, namely, $x < x_0$.

We observe,

Lemma 14.7.8. *If $f = o(g)$, then it is* not *true that $g = O(f)$.*

[9] We can't simply use the limit as $x \to \infty$ in the definition of $O()$, because if $f(x)/g(x)$ oscillates between, say, 3 and 5 as x grows, then $f = O(g)$ because $f \leq 5g$, but $\lim_{x\to\infty} f(x)/g(x)$ does not exist. So instead of limit, we use the technical notion of lim sup. In this oscillating case, $\limsup_{x\to\infty} f(x)/g(x) = 5$.

The precise definition of lim sup is

$$\limsup_{x\to\infty} h(x) ::= \lim_{x\to\infty} \text{lub}_{y\geq x} h(y),$$

where "lub" abbreviates "least upper bound."

Proof.

$$\lim_{x\to\infty} \frac{g(x)}{f(x)} = \frac{1}{\lim_{x\to\infty} f(x)/g(x)} = \frac{1}{0} = \infty,$$

so $g \neq O(f)$. ■

Proposition 14.7.9. $100x^2 = O(x^2)$.

Proof. Choose $c = 100$ and $x_0 = 1$. Then the proposition holds, since for all $x \geq 1$, $|100x^2| \leq 100x^2$. ■

Proposition 14.7.10. $x^2 + 100x + 10 = O(x^2)$.

Proof. $(x^2+100x+10)/x^2 = 1+100/x+10/x^2$ and so its limit as x approaches infinity is $1+0+0 = 1$. So in fact, $x^2+100x+10 \sim x^2$, and therefore $x^2+100x+10 = O(x^2)$. Indeed, it's conversely true that $x^2 = O(x^2 + 100x + 10)$. ■

Proposition 14.7.10 generalizes to an arbitrary polynomial:

Proposition 14.7.11. $a_k x^k + a_{k-1}x^{k-1} + \cdots + a_1 x + a_0 = O(x^k)$.

We'll omit the routine proof.

Big Oh notation is especially useful when describing the running time of an algorithm. For example, the usual algorithm for multiplying $n \times n$ matrices uses a number of operations proportional to n^3 in the worst case. This fact can be expressed concisely by saying that the running time is $O(n^3)$. So this asymptotic notation allows the speed of the algorithm to be discussed without reference to constant factors or lower-order terms that might be machine specific. It turns out that there is another, ingenious matrix multiplication procedure that uses $O(n^{2.55})$ operations. This procedure will therefore be much more efficient on large enough matrices. Unfortunately, the $O(n^{2.55})$-operation multiplication procedure is almost never used in practice because it happens to be less efficient than the usual $O(n^3)$ procedure on matrices of practical size.[10]

I thought that was its only use!

14.7.3 Theta

Sometimes we want to specify that a running time $T(n)$ is precisely quadratic up to constant factors (both upper bound *and* lower bound). We could do this by saying that $T(n) = O(n^2)$ and $n^2 = O(T(n))$, but rather than say both, mathematicians have devised yet another symbol, Θ, to do the job.

what does this mean?

See next pg

[10] It is even conceivable that there is an $O(n^2)$ matrix multiplication procedure, but none is known.

Definition 14.7.12.

$$f = \Theta(g) \quad \text{iff} \quad f = O(g) \text{ and } g = O(f).$$

how is that diff from when it just applies One way

The statement $f = \Theta(g)$ can be paraphrased intuitively as "f and g are equal to within a constant factor."

The Theta notation allows us to highlight growth rates and allow suppression of distracting factors and low-order terms. For example, if the running time of an algorithm is

$$T(n) = 10n^3 - 20n^2 + 1,$$

then we can more simply write

$$T(n) = \Theta(n^3).$$

In this case, we would say that T *is of order* n^3 or that $T(n)$ *grows cubically*, which is probably what we really want to know. Another such example is

$$\pi^2 3^{x-7} + \frac{(2.7x^{113} + x^9 - 86)^4}{\sqrt{x}} - 1.08^{3x} = \Theta(3^x).$$

Just knowing that the running time of an algorithm is $\Theta(n^3)$, for example, is useful, because if n doubles we can predict that the running time will *by and large*[11] increase by a factor of at most 8 for large n. In this way, Theta notation preserves information about the scalability of an algorithm or system. Scalability is, of course, a big issue in the design of algorithms and systems.

14.7.4 Pitfalls with Asymptotic Notation

There is a long list of ways to make mistakes with asymptotic notation. This section presents some of the ways that Big Oh notation can lead to ruin and despair. With minimal effort, you can cause just as much chaos with the other symbols.

The Exponential Fiasco

Sometimes relationships involving Big Oh are not so obvious. For example, one might guess that $4^x = O(2^x)$ since 4 is only a constant factor larger than 2. This reasoning is incorrect, however; 4^x actually grows as the square of 2^x.

[11] Since $\Theta(n^3)$ only implies that the running time, $T(n)$, is between cn^3 and dn^3 for constants $0 < c < d$, the time $T(2n)$ could regularly exceed $T(n)$ by a factor as large as $8d/c$. The factor is sure to be close to 8 for all large n only if $T(n) \sim n^3$.

Constant Confusion

Every constant is $O(1)$. For example, $17 = O(1)$. This is true because if we let $f(x) = 17$ and $g(x) = 1$, then there exists a $c > 0$ and an x_0 such that $|f(x)| \leq cg(x)$. In particular, we could choose $c = 17$ and $x_0 = 1$, since $|17| \leq 17 \cdot 1$ for all $x \geq 1$. We can construct a false theorem that exploits this fact.

False Theorem 14.7.13.

$$\sum_{i=1}^{n} i = O(n)$$

Bogus proof. Define $f(n) = \sum_{i=1}^{n} i = 1+2+3+\cdots+n$. Since we have shown that every constant i is $O(1)$, $f(n) = O(1) + O(1) + \cdots + O(1) = O(n)$. ■

Of course in reality $\sum_{i=1}^{n} i = n(n+1)/2 \neq O(n)$.

The error stems from confusion over what is meant in the statement $i = O(1)$. For any *constant* $i \in \mathbb{N}$ it is true that $i = O(1)$. More precisely, if f is any constant function, then $f = O(1)$. But in this False Theorem, i is not constant—it ranges over a set of values $0, 1, \ldots, n$ that depends on n.

And anyway, we should not be adding $O(1)$'s as though they were numbers. We never even defined what $O(g)$ means by itself; it should only be used in the context "$f = O(g)$" to describe a relation between functions f and g.

Lower Bound Blunder

Sometimes people incorrectly use Big Oh in the context of a lower bound. For example, they might say, "The running time, $T(n)$, is at least $O(n^2)$," when they probably mean "$n^2 = O(T(n))$." [12]

Equality Blunder

The notation $f = O(g)$ is too firmly entrenched to avoid, but the use of "=" is really regrettable. For example, if $f = O(g)$, it seems quite reasonable to write $O(g) = f$. But doing so might tempt us to the following blunder: because $2n = O(n)$, we can say $O(n) = 2n$. But $n = O(n)$, so we conclude that $n = O(n) = 2n$, and therefore $n = 2n$. To avoid such nonsense, we will never write "$O(f) = g$."

Similarly, you will often see statements like

$$H_n = \ln(n) + \gamma + O\left(\frac{1}{n}\right)$$

[12] This would more usually be expressed as "$T(n) = \Omega(n^2)$."

or

$$n! = (1 + o(1))\sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

In such cases, the true meaning is

$$H_n = \ln(n) + \gamma + f(n)$$

for some $f(n)$ where $f(n) = O(1/n)$, and

$$n! = (1 + g(n))\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

where $g(n) = o(1)$. These transgressions are OK as long as you (and your reader) know what you mean.

14.7.5 Omega

Suppose you want to make a statement of the form "the running time of the algorithm is a least...". Can you say it is "at least $O(n^2)$"? No! This statement is meaningless since big-oh can only be used for *upper* bounds. For lower bounds, we use a different symbol, called "big-Omega."

here is lower bounds

Definition 14.7.14. Given functions $f, g : \mathbb{R} \to \mathbb{R}$, define

$$f = \Omega(g)$$

to mean

$$g = O(f).$$

For example, $x^2 = \Omega(x)$, $2^x = \Omega(x^2)$, and $x/100 = \Omega(100x + \sqrt{x})$.

So if the running time of your algorithm on inputs of size n is $T(n)$, and you want to say it is at least quadratic, say

$$T(n) = \Omega(n^2).$$

Likewise, there is also a symbol called little-omega, analogous to little-oh, to denote that one function grows strictly faster than another function.

Definition 14.7.15. For functions $f, g : \mathbb{R} \to \mathbb{R}$ with f nonnegative, define

$$f = \omega(g)$$

to mean

$$g = o(f).$$

For example, $x^{1.5} = \omega(x)$ and $\sqrt{x} = \omega(\ln^2(x))$.

The little-omega symbol is not as widely used as the other asymptotic symbols we defined.

Problems for Section 14.1

Class Problems

Problem 14.1.
We begin with two large glasses. The first glass contains a pint of water, and the second contains a pint of wine. We pour 1/3 of a pint from the first glass into the second, stir up the wine/water mixture in the second glass, and then pour 1/3 of a pint of the mix back into the first glass and repeat this pouring back-and-forth process a total of n times.

(a) Describe a closed form formula for the amount of wine in the first glass after n back-and-forth pourings.

(b) What is the limit of the amount of wine in each glass as n approaches infinity?

Problem 14.2.
You've seen this neat trick for evaluating a geometric sum:

$$\begin{aligned} S &= 1 + z + z^2 + \ldots + z^n \\ zS &= z + z^2 + \ldots + z^n + z^{n+1} \\ S - zS &= 1 - z^{n+1} \\ S &= \frac{1 - z^{n+1}}{1 - z} \end{aligned}$$

Use the same approach to find a closed-form expression for this sum:

$$T = 1z + 2z^2 + 3z^3 + \ldots + nz^n$$

Homework Problems

Problem 14.3.
Is a Harvard degree really worth more than an MIT degree?! Let us say that a person with a Harvard degree starts with \$40,000 and gets a \$20,000 raise every year after graduation, whereas a person with an MIT degree starts with \$30,000, but gets a 20% raise every year. Assume inflation is a fixed 8% every year. That is, \$1.08 a year from now is worth \$1.00 today.

(a) How much is a Harvard degree worth today if the holder will work for n years following graduation?

(b) How much is an MIT degree worth in this case?

(c) If you plan to retire after twenty years, which degree would be worth more?

Problem 14.4.
Suppose you deposit \$100 into your MIT Credit Union account today, \$99 in one month from now, \$98 in two months from now, and so on. Given that the interest rate is constantly 0.3% per month, how long will it take to save \$5,000?

Problems for Section 14.3

Exam Problems

Problem 14.5.
Assume n is an integer larger than 1. Circle all the correct inequalities below.

Explanations are not required, but partial credit for wrong answers will not be given without them. *Hint:* You may find the graphs helpful.

- $\sum_{i=1}^{n} \ln(i+1) \leq \ln 2 + \int_1^n \ln(x+1)dx$
- $\sum_{i=1}^{n} \ln(i+1) \leq \int_0^n \ln(x+2)dx$
- $\sum_{i=1}^{n} \frac{1}{i} \geq \int_0^n \frac{1}{x+1}dx$

Homework Problems

Problem 14.6.
Let $f : \mathbb{R}^+ \to \mathbb{R}^+$ be a weakly decreasing function. Define

$$S ::= \sum_{i=1}^{n} f(i)$$

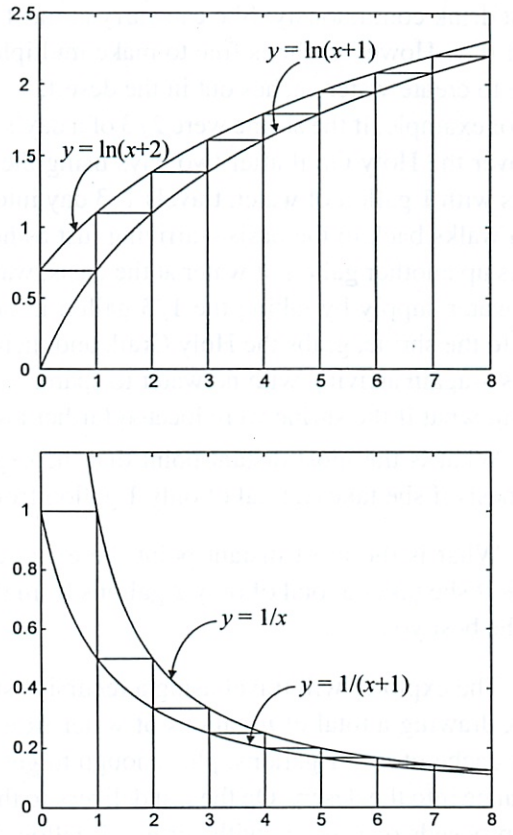
and

$$I ::= \int_1^n f(x)\,dx.$$

Prove that

$$I + f(n) \leq S \leq I + f(1).$$

Hint: See Theorem 14.6.



Problem 14.7.
Use integration to find upper and lower bounds that differ by at most 0.1 for the following sum. (You may need to add the first few terms explicitly and then use integrals to bound the sum of the remaining terms.)

$$\sum_{i=1}^{\infty} \frac{1}{(2i+1)^2}$$

Problems for Section 14.4

Class Problems

Problem 14.8.
An explorer is trying to reach the Holy Grail, which she believes is located in a desert shrine d days walk from the nearest oasis. In the desert heat, the explorer

must drink continuously. She can carry at most 1 gallon of water, which is enough for 1 day. However, she is free to make multiple trips carrying up to a gallon each time to create water caches out in the desert.

For example, if the shrine were 2/3 of a day's walk into the desert, then she could recover the Holy Grail after two days using the following strategy. She leaves the oasis with 1 gallon of water, travels 1/3 day into the desert, caches 1/3 gallon, and then walks back to the oasis—arriving just as her water supply runs out. Then she picks up another gallon of water at the oasis, walks 1/3 day into the desert, tops off her water supply by taking the 1/3 gallon in her cache, walks the remaining 1/3 day to the shrine, grabs the Holy Grail, and then walks for 2/3 of a day back to the oasis—again arriving with no water to spare.

But what if the shrine were located farther away?

(a) What is the most distant point that the explorer can reach and then return to the oasis if she takes a total of only 1 gallon from the oasis?

(b) What is the most distant point the explorer can reach and still return to the oasis if she takes a total of only 2 gallons from the oasis? No proof is required; just do the best you can.

(c) The explorer will travel using a recursive strategy to go far into the desert and back drawing a total of n gallons of water from the oasis. Her strategy is to build up a cache of $n-1$ gallons, plus enough to get home, a certain fraction of a day's distance into the desert. On the last delivery to the cache, instead of returning home, she proceeds recursively with her $n-1$ gallon strategy to go farther into the desert and return to the cache. At this point, the cache has just enough water left to get her home.

Prove that with n gallons of water, this strategy will get her $H_n/2$ days into the desert and back, where H_n is the nth Harmonic number:

$$H_n ::= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}.$$

Conclude that she can reach the shrine, however far it is from the oasis.

(d) Suppose that the shrine is $d = 10$ days walk into the desert. Use the asymptotic approximation $H_n \sim \ln n$ to show that it will take more than a million years for the explorer to recover the Holy Grail.

Problem 14.9.

There is a number a such that $\sum_{i=1}^{\infty} i^p$ converges iff $p < a$. What is the value of a? Prove it.

Homework Problems

Problem 14.10.
There is a bug on the edge of a 1-meter rug. The bug wants to cross to the other side of the rug. It crawls at 1 cm per second. However, at the end of each second, a malicious first-grader named Mildred Anderson *stretches* the rug by 1 meter. Assume that her action is instantaneous and the rug stretches uniformly. Thus, here's what happens in the first few seconds:

- The bug walks 1 cm in the first second, so 99 cm remain ahead.
- Mildred stretches the rug by 1 meter, which doubles its length. So now there are 2 cm behind the bug and 198 cm ahead.
- The bug walks another 1 cm in the next second, leaving 3 cm behind and 197 cm ahead.
- Then Mildred strikes, stretching the rug from 2 meters to 3 meters. So there are now $3 \cdot (3/2) = 4.5$ cm behind the bug and $197 \cdot (3/2) = 295.5$ cm ahead.
- The bug walks another 1 cm in the third second, and so on.

Your job is to determine this poor bug's fate.

(a) During second i, what *fraction* of the rug does the bug cross?

(b) Over the first n seconds, what fraction of the rug does the bug cross altogether? Express your answer in terms of the Harmonic number H_n.

(c) The known universe is thought to be about $3 \cdot 10^{10}$ light years in diameter. How many universe diameters must the bug travel to get to the end of the rug?

Problems for Section 14.7

Practice Problems

Problem 14.11.
Let $f(n) = n^3$. For each function $g(n)$ in the table below, indicate which of the indicated asymptotic relations hold.

$g(n)$	$f = O(g)$	$f = o(g)$	$g = O(f)$	$g = o(f)$
$6 - 5n - 4n^2 + 3n^3$				
$n^3 \log n$				
$(\sin(\pi n/2) + 2)\, n^3$				
$n^{\sin(\pi n/2)+2}$				
$\log n!$				
$e^{0.2n} - 100n^3$				

Problem 14.12.
Circle each of the true statements below.

Explanations are not required, but partial credit for wrong answers will not be given without them.

- $n^2 \sim n^2 + n$
- $3^n = O\left(2^n\right)$
- $n^{\sin(n\pi/2)+1} = o\left(n^2\right)$
- $n = \Theta\left(\dfrac{3n^3}{(n+1)(n-1)}\right)$

Problem 14.13.
Show that

$$\ln(n^2!) = \Theta(n^2 \ln n)$$

Homework Problems

Problem 14.14. **(a)** Prove that $\log x < x$ for all $x > 1$ (requires elementary calculus).

(b) Prove that the relation, R, on functions such that $f \; R \; g$ iff $f = o(g)$ is a strict partial order.

(c) Prove that $f \sim g$ iff $f = g + h$ for some function $h = o(g)$.

Problem 14.15.
Indicate which of the following holds for each pair of functions $(f(n), g(n))$ in the table below. Assume $k \geq 1$, $\epsilon > 0$, and $c > 1$ are constants. Pick the four table entries you consider to be the most challenging or interesting and justify your answers to these.

$f(n)$	$g(n)$	$f = O(g)$	$f = o(g)$	$g = O(f)$	$g = o(f)$	$f = \Theta(g)$	$f \sim g$
2^n	$2^{n/2}$						
$\sqrt{n}$	$n^{\sin n\pi/2}$						
$\log(n!)$	$\log(n^n)$						
n^k	c^n						
$\log^k n$	n^ϵ						

Problem 14.16.
Let f, g be nonnegative real-valued functions such that $\lim_{x\to\infty} f(x) = \infty$ and $f \sim g$.

(a) Give an example of f, g such that NOT$(2^f \sim 2^g)$.

(b) Prove that $\log f \sim \log g$.

(c) Use Stirling's formula to prove that in fact

$$\log(n!) \sim n \log n$$

Problem 14.17.
Determine which of these choices

$$\Theta(n), \quad \Theta(n^2 \log n), \quad \Theta(n^2), \quad \Theta(1), \quad \Theta(2^n), \quad \Theta(2^{n \ln n}), \quad \text{none of these}$$

describes each function's asymptotic behavior. Full proofs are not required, but briefly explain your answers.

(a)

$$n + \ln n + (\ln n)^2$$

(b)

$$\frac{n^2 + 2n - 3}{n^2 - 7}$$

(c)

$$\sum_{i=0}^{n} 2^{2i+1}$$

(d)

$$\ln(n^2!)$$

(e)

$$\sum_{k=1}^{n} k\left(1-\frac{1}{2^k}\right)$$

Problem 14.18. **(a)** Either prove or disprove each of the following statements.

- $n! = O((n+1)!)$
- $(n+1)! = O(n!)$
- $n! = \Theta((n+1)!)$
- $n! = o((n+1)!)$
- $(n+1)! = o(n!)$

(b) Show that $\left(\frac{n}{3}\right)^{n+e} = o(n!)$.

Problem 14.19.
Prove that $\sum_{k=1}^{n} k^6 = \Theta(n^7)$.

Class Problems

Problem 14.20.
Give an elementary proof (without appealing to Stirling's formula) that $\log(n!) = \Theta(n \log n)$.

Problem 14.21.
Suppose $f, g : \mathbb{N}^+ \to \mathbb{N}^+$ and $f \sim g$.

(a) Prove that $2f \sim 2g$.

(b) Prove that $f^2 \sim g^2$.

(c) Give examples of f and g such that $2^f \not\sim 2^g$.

Problem 14.22.
Recall that for functions f, g on $\mathbb{N}$, $f = O(g)$ iff

$$\exists c \in \mathbb{N}\, \exists n_0 \in \mathbb{N}\, \forall n \geq n_0 \quad c \cdot g(n) \geq |f(n)|. \tag{14.33}$$

For each pair of functions below, determine whether $f = O(g)$ and whether $g = O(f)$. In cases where one function is O() of the other, indicate the *smallest nonegative integer*, c, and for that smallest c, the *smallest corresponding nonegative integer* n_0 ensuring that condition (14.33) applies.

(a) $f(n) = n^2, g(n) = 3n$.

$f = O(g)$	YES	NO	If YES, $c =$ _____, $n_0 =$ _____
$g = O(f)$	YES	NO	If YES, $c =$ _____, $n_0 =$ _____

(b) $f(n) = (3n - 7)/(n + 4), g(n) = 4$

$f = O(g)$	YES	NO	If YES, $c =$ _____, $n_0 =$ _____
$g = O(f)$	YES	NO	If YES, $c =$ _____, $n_0 =$ _____

(c) $f(n) = 1 + (n \sin(n\pi/2))^2, g(n) = 3n$

$f = O(g)$	YES	NO	If yes, $c =$ _____ $n_0 =$ _____
$g = O(f)$	YES	NO	If yes, $c =$ _____ $n_0 =$ _____

Problem 14.23.

False Claim.

$$2^n = O(1). \tag{14.34}$$

Explain why the claim is false. Then identify and explain the mistake in the following bogus proof.

Bogus proof. The proof by induction on n where the induction hypothesis, $P(n)$, is the assertion (14.34).

base case: $P(0)$ holds trivially.

inductive step: We may assume $P(n)$, so there is a constant $c > 0$ such that $2^n \leq c \cdot 1$. Therefore,

$$2^{n+1} = 2 \cdot 2^n \leq (2c) \cdot 1,$$

which implies that $2^{n+1} = O(1)$. That is, $P(n + 1)$ holds, which completes the proof of the inductive step.

We conclude by induction that $2^n = O(1)$ for all n. That is, the exponential function is bounded by a constant.

■

Problem 14.24. **(a)** Prove that the relation, R, on functions such that $f \; R \; g$ iff $f = o(g)$ is a strict partial order.

(b) Describe two functions f, g that are incomparable under big Oh:

$$f \neq O(g) \text{ AND } g \neq O(f).$$

Conclude that R is not a path-total order.

Exam Problems

Problem 14.25. **(a)** Show that

$$(an)^{b/n} \sim 1.$$

where a, b are positive constants and $\sim$ denotes asymptotic equality. *Hint:* $an = a2^{\log_2 n}$.

(b) You may assume that if $f(n) \geq 1$ and $g(n) \geq 1$ for all n, then $f \sim g \longrightarrow f^{\frac{1}{n}} \sim g^{\frac{1}{n}}$. Show that

$$\sqrt[n]{n!} = \Theta(n).$$

Problem 14.26.

(a) Define a function $f(n)$ such that $f = \Theta(n^2)$ and NOT$(f \sim n^2)$.

(b) Define a function $g(n)$ such that $g = O(n^2)$, $g \neq \Theta(n^2)$ and $g \neq o(n^2)$.

Problem 14.27. **(a)** Show that

$$(an)^{b/n} \sim 1.$$

where a, b are positive constants and $\sim$ denotes asymptotic equality. *Hint:* $an = a2^{\log_2 n}$.

(b) Show that

$$\sqrt[n]{n!} = \Theta(n).$$

Problem 14.28.

(a) Define two functions f, g that are incomparable under big Oh:

$$f \neq O(g) \text{ AND } g \neq O(f).$$

(b) For each of the asymptotic relations below on the set of nonnegative real-valued functions, indicate whether it is *transitive* but not a partial order (**Tr**), a *total order* (**Tot**), a *strict partial order* that is not total (**Str**), a *weak partial order* that is not total (**Wk**), or *none* of the above (**Non**).

- $f \sim g$, the "asymptotically Equal" relation.
- $f = o(g)$, the "little Oh" relation.
- $f = O(g)$, the "big Oh" relation.
- $f = \Theta(g)$, the "Theta" relation.

Index

−, set difference, 68
C_n, 302, **323**
$K_{3,3}$, 357
K_5, 357
big omega, 426
$\Theta()$, 423
bij, **88**
$\mathbb{C}$, **68**
$\emptyset$, **68**
::=, **7**
$\equiv \pmod{n}$, 199
$\forall$, **8**
Done, **384**
$\in$, **8**
inj, **82**, **88**
$\mathbb{Z}$, **68**
$\mathbb{Z}^-$, **68**
$\cap$, **68**
λ, **71**
$\mathbb{N}$, **8**, **68**
$\overline{A}$, **68**
$\phi(n)$, **210**
$\mathbb{Z}^+$, **8**
$\mathcal{P}(A)$, **69**
$\mathbb{Q}$, **68**
$\mathbb{R}$, **68**
$\mathbb{R}^+$, **68**
$\sim$, 421
$\sim$ (asymptotic equality), 415
strict, **88**
$\subset$, **68**
$\subseteq$, **68**
surj, **88**
$\cup$, **68**
k-edge connected, **324**
$n+1$-bit adder, **141**
icr , **332**
while programs, **384**
2-D Array, 292
2-Layer Array, **292**
2-dimensional array, **281**

adjacency matrix, **237**
adjacent, **298**
Adleman, 207
Agrawal, 183
alphabet, **158**
annuity, **392**
antecedents, **11**
antichain, **253**, 267
antisymmetric, **244**, **256**
antisymmetry, **244**
arrows, **231**
assignment statement, **132**, **384**
asymmetric, **243**
asymmetry, **243**
asymptotically equal, **415**
asymptotically smaller, **421**
asymptotic relations, 431
average degree, 300, 356
axiomatic method, **10**
Axiom of Choice, 102
axioms, **4**, **10**

Banach-Tarski, 102
base case, **116**
basis step, **116**
Beneš nets, **285**
bijective, **76**
binary predicate, 54
binary relation, **74**
Binary relations, **73**
binary trees, **174**
bipartite graph, **305**, 309, 344, 369

15 Cardinality Rules

Counting Rules

15.1 Counting One Thing by Counting Another

4/8

How do you count the number of people in a crowded room? You could count heads, since for each person there is exactly one head. Alternatively, you could count ears and divide by two. Of course, you might have to adjust the calculation if someone lost an ear in a pirate raid or someone was born with three ears. The point here is that you can often *count one thing by counting another*, though some fudge factors may be required. This is a central theme of counting, from the easiest problems to the hardest. In fact, we've already seen this technique used in Theorem 5.1.5 where the number of subsets of an n-element set was proved to the same as the number of length-n bit-strings because by describing a bijection between the subsets and the bit-strings.

The most direct way to count one thing by counting another is to find a bijection between them, since if there is a bijection between two sets, then the sets have the same size. This important fact is commonly known as the *Bijection Rule*. We've already seen it as the Mapping Rules bijective case (5.3).

15.1.1 The Bijection Rule

The Bijection Rule acts as a magnifier of counting ability; if you figure out the size of one set, then you can immediately determine the sizes of many other sets via bijections. For example, consider the two sets mentioned at the beginning of Part III:

don't remember →

A = all ways to select a dozen doughnuts when five varieties are available

B = all 16-bit sequences with exactly 4 ones

So I think I missed the problem

How can you know How many ways can you pick 12 donuts from 5 varieties?

which 5?

Let's consider a particular element of set A:

$$\underbrace{00}_{\text{chocolate}}\ \underbrace{\ }_{\text{lemon-filled}}\ \underbrace{000000}_{\text{sugar}}\ \underbrace{00}_{\text{glazed}}\ \underbrace{00}_{\text{plain}}$$

We've depicted each doughnut with a 0 and left a gap between the different varieties. Thus, the selection above contains two chocolate doughnuts, no lemon-filled, six sugar, two glazed, and two plain. Now let's put a 1 into each of the four gaps:

$$\underbrace{00}_{\text{chocolate}}\ 1\ \underbrace{\ }_{\text{lemon-filled}}\ 1\ \underbrace{000000}_{\text{sugar}}\ 1\ \underbrace{00}_{\text{glazed}}\ 1\ \underbrace{00}_{\text{plain}}$$

- must have 12!

and close up the gaps:

$$0011000000100100\,.$$

We've just formed a 16-bit number with exactly 4 ones —an element of B!

This example suggests a bijection from set A to set B: map a dozen doughnuts consisting of:

$$c \text{ chocolate, } l \text{ lemon-filled, } s \text{ sugar, } g \text{ glazed, and } p \text{ plain}$$

to the sequence:

$$\underbrace{0\ldots0}_{c}\ 1\ \underbrace{0\ldots0}_{l}\ 1\ \underbrace{0\ldots0}_{s}\ 1\ \underbrace{0\ldots0}_{g}\ 1\ \underbrace{0\ldots0}_{p}$$

The resulting sequence always has 16 bits and exactly 4 ones, and thus is an element of B. Moreover, the mapping is a bijection; every such bit sequence is mapped to by exactly one order of a dozen doughnuts. Therefore, $|A| = |B|$ by the Bijection Rule!

This example demonstrates the magnifying power of the bijection rule. We managed to prove that two very different sets are actually the same size —even though we don't know exactly how big either one is. But as soon as we figure out the size of one set, we'll immediately know the size of the other.

This particular bijection might seem frighteningly ingenious if you've not seen it before. But you'll use essentially this same argument over and over, and soon you'll consider it routine.

15.2 Counting Sequences

The Bijection Rule lets us count one thing by counting another. This suggests a general strategy: get really good at counting just a *few* things and then use bijections to count *everything else*. This is the strategy we'll follow. In particular, we'll get really good at counting *sequences*. When we want to determine the size of some other set T, we'll find a bijection from T to a set of sequences S. Then we'll use our super-ninja sequence-counting skills to determine $|S|$, which immediately gives us $|T|$. We'll need to hone this idea somewhat as we go along, but that's pretty much the plan!

15.2.1 The Product Rule

The *Product Rule* gives the size of a product of sets. Recall that if $P_1, P_2, \ldots, P_n$ are sets, then

$$P_1 \times P_2 \times \ldots \times P_n$$

is the set of all sequences whose first term is drawn from P_1, second term is drawn from P_2 and so forth.

Rule 15.2.1 (Product Rule). *If $P_1, P_2, \ldots P_n$ are finite sets, then:*

$$|P_1 \times P_2 \times \ldots \times P_n| = |P_1| \cdot |P_2| \cdots |P_n|$$

For example, suppose a *daily diet* consists of a breakfast selected from set B, a lunch from set L, and a dinner from set D where:

$$B = \{\text{pancakes, bacon and eggs, bagel, Doritos}\}$$
$$L = \{\text{burger and fries, garden salad, Doritos}\}$$
$$D = \{\text{macaroni, pizza, frozen burrito, pasta, Doritos}\}$$

Then $B \times L \times D$ is the set of all possible daily diets. Here are some sample elements:

(pancakes, burger and fries, pizza)
(bacon and eggs, garden salad, pasta)
(Doritos, Doritos, frozen burrito)

The Product Rule tells us how many different daily diets are possible:

$$\begin{aligned} |B \times L \times D| &= |B| \cdot |L| \cdot |D| \\ &= 4 \cdot 3 \cdot 5 \\ &= 60. \end{aligned}$$

15.2.2 Subsets of an n-element Set

The fact that there are 2^n subsets of an n-element set was proved in Theorem 5.1.5 by setting up a bijection between the subsets and the length-n bit-strings. So the original problem about subsets was tranformed into a question about sequences — *exactly according to plan!*. Now we can fill in the missing explanation of why there are 2^n length-n bit-strings: we can write the set of all n-bit sequences as a product of sets:

$$\{0,1\}^n ::= \underbrace{\{0,1\} \times \{0,1\} \times \ldots \times \{0,1\}}_{n \text{ terms}}.$$

Then Product Rule gives the answer:

$$|\{0,1\}^n| = |\{0,1\}|^n = 2^n.$$

15.2.3 The Sum Rule

Linus allocates his big sister Lucy a quota of 20 crabby days, 40 irritable days, and 60 generally surly days. On how many days can Lucy be out-of-sorts one way or another? Let set C be her crabby days, I be her irritable days, and S be the generally surly. In these terms, the answer to the question is $|C \cup I \cup S|$. Now assuming that she is permitted at most one bad quality each day, the size of this union of sets is given by the *Sum Rule*:

Rule 15.2.2 (Sum Rule). *If $A_1, A_2, \ldots, A_n$ are disjoint sets, then:*

$$|A_1 \cup A_2 \cup \ldots \cup A_n| = |A_1| + |A_2| + \ldots + |A_n|$$

Thus, according to Linus' budget, Lucy can be out-of-sorts for:

$$\begin{aligned} |C \cup I \cup S| &= |C| + |I| + |S| \\ &= 20 + 40 + 60 \\ &= 120 \text{ days} \end{aligned}$$

Notice that the Sum Rule holds only for a union of *disjoint* sets. Finding the size of a union of overlapping sets is a more complicated problem that we'll take up later.

15.2.4 Counting Passwords

Few counting problems can be solved with a single rule. More often, a solution is a flurry of sums, products, bijections, and other methods.

For solving problems involving passwords, telephone numbers, and license plates, the sum and product rules are useful together. For example, on a certain computer system, a valid password is a sequence of between six and eight symbols. The first symbol must be a letter (which can be lowercase or uppercase), and the remaining symbols must be either letters or digits. How many different passwords are possible?

Let's define two sets, corresponding to valid symbols in the first and subsequent positions in the password.

$$\begin{aligned} F &= \{a, b, \ldots, z, A, B, \ldots, Z\} \\ S &= \{a, b, \ldots, z, A, B, \ldots, Z, 0, 1, \ldots, 9\} \end{aligned}$$

In these terms, the set of all possible passwords is:[1]

$$(F \times S^5) \cup (F \times S^6) \cup (F \times S^7)$$

[1] The notation S^5 means $S \times S \times S \times S \times S$.

Thus, the length-six passwords are in the set $F \times S^5$, the length-seven passwords are in $F \times S^6$, and the length-eight passwords are in $F \times S^7$. Since these sets are disjoint, we can apply the Sum Rule and count the total number of possible passwords as follows:

$$\begin{aligned}
&|(F \times S^5) \cup (F \times S^6) \cup (F \times S^7)| \\
&\quad = |F \times S^5| + |F \times S^6| + |F \times S^7| && \text{Sum Rule} \\
&\quad = |F| \cdot |S|^5 + |F| \cdot |S|^6 + |F| \cdot |S|^7 && \text{Product Rule} \\
&\quad = 52 \cdot 62^5 + 52 \cdot 62^6 + 52 \cdot 62^7 \\
&\quad \approx 1.8 \cdot 10^{14} \text{ different passwords.}
\end{aligned}$$

15.3 The Generalized Product Rule

In how many ways can, say, three different prizes be awarded to n people? This is easy to answer using our strategy of translating the problem about awards into a problem about sequences. Let P be the set of n people taking the course. Then there is a bijection from ways of awarding the three prizes to the set $P^3 ::= P \times P \times P$. In particular, the assignment:

"person x wins prize #1, y wins prize #2, and z wins prize #3"

maps to the sequence (x, y, z). By the Product Rule, we have $|P^3| = |P|^3 = n^3$, so there are n^3 ways to award the prizes to a class of n people.

But what if the three prizes must be awarded to *different* students? As before, we could map the assignment

"person x wins prize #1, y wins prize #2, and z wins prize #3"

to the triple $(x, y, z) \in P^3$. But this function is *no longer a bijection*. For example, no valid assignment maps to the triple (Dave, Dave, Becky) because Dave is not allowed to receive two awards. However, there *is* a bijection from prize assignments to the set:

$$S = \{(x, y, z) \in P^3 \mid x, y, \text{ and } z \text{ are different people}\}$$

This reduces the original problem to a problem of counting sequences. Unfortunately, the Product Rule is of no help in counting sequences of this type because the entries depend on one another; in particular, they must all be different. However, a slightly sharper tool does the trick.

Prizes for *truly exceptional* Coursework

Given everyone's hard work on this material, the instructors considered awarding some prizes for truly exceptional coursework. Here are three possible prize categories:

Best Administrative Critique We asserted that the quiz was closed-book. On the cover page, one strong candidate for this award wrote, "There is no book."

Awkward Question Award "Okay, the left sock, right sock, and pants are in an antichain, but how —even with assistance —could I put on all three at once?"

Best Collaboration Statement Inspired by a student who wrote "I worked alone" on Quiz 1.

but here prize must go to each a diff person

Rule 15.3.1 (Generalized Product Rule). *Let S be a set of length-k sequences. If there are:*

- *n_1 possible first entries,*
- *n_2 possible second entries for each first entry,*
- ⋮
- *n_k possible kth entries for each sequence of first $k-1$ entries,*

then:

$$|S| = n_1 \cdot n_2 \cdot n_3 \cdots n_k$$

In the awards example, S consists of sequences (x, y, z). There are n ways to choose x, the recipient of prize #1. For each of these, there are $n-1$ ways to choose y, the recipient of prize #2, since everyone except for person x is eligible. For each combination of x and y, there are $n-2$ ways to choose z, the recipient of prize #3, because everyone except for x and y is eligible. Thus, according to the Generalized Product Rule, there are

$$|S| = n \cdot (n-1) \cdot (n-2)$$

ways to award the 3 prizes to different people.

↑ Thats what I wrote earlier w/ factorial

So if prize for each person, 1 prize per person, Possibility = $n! = n \cdot (n-1) \cdot (n-2) \cdots 1$ right?

15.3.1 Defective Dollar Bills

A dollar bill is *defective* if some digit appears more than once in the 8-digit serial number. If you check your wallet, you'll be sad to discover that defective bills are all-too-common. In fact, how common are *nondefective* bills? Assuming that the digit portions of serial numbers all occur equally often, we could answer this question by computing

$$\text{fraction of nondefective bills} = \frac{|\{\text{serial \#'s with all digits different}\}|}{|\{\text{serial numbers}\}|}. \quad (15.1)$$

Let's first consider the denominator. Here there are no restrictions; there are are 10 possible first digits, 10 possible second digits, 10 third digits, and so on. Thus, the total number of 8-digit serial numbers is 10^8 by the Product Rule.

Next, let's turn to the numerator. Now we're not permitted to use any digit twice. So there are still 10 possible first digits, but only 9 possible second digits, 8 possible third digits, and so forth. Thus, by the Generalized Product Rule, there are

$$10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 = \frac{10!}{2} = 1{,}814{,}400$$

serial numbers with all digits different. Plugging these results into Equation 15.1, we find:

$$\text{fraction of nondefective bills} = \frac{1{,}814{,}400}{100{,}000{,}000} = 1.8144\%$$

15.3.2 A Chess Problem

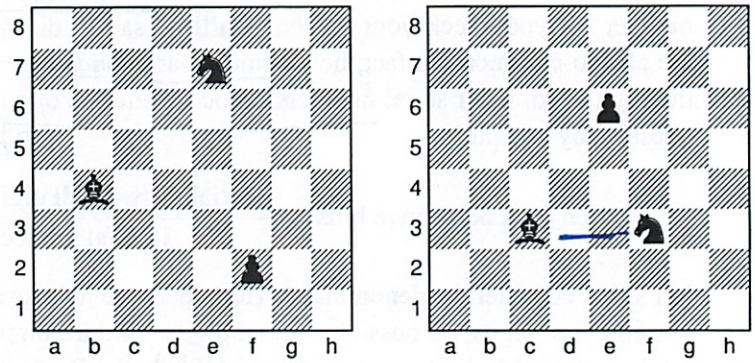
In how many different ways can we place a pawn (P), a knight (N), and a bishop (B) on a chessboard so that no two pieces share a row or a column? A valid configuration is shown in Figure 15.1(a), and an invalid configuration is shown in Figure 15.1(b).

First, we map this problem about chess pieces to a question about sequences. There is a bijection from configurations to sequences

$$(r_P, c_P, r_N, c_N, r_B, c_B)$$

where r_P, r_N, and r_B are distinct rows and c_P, c_N, and c_B are distinct columns. In particular, r_P is the pawn's row, c_P is the pawn's column, r_N is the knight's row, etc. Now we can count the number of such sequences using the Generalized Product Rule:

- r_P is one of 8 rows



(a) valid (b) invalid

Figure 15.1 Two ways of placing a pawn (♙), a knight (♘), and a bishop (♗) on a chessboard. The configuration shown in (b) is invalid because the bishop and the knight are in the same row.

- c_P is one of 8 columns
- r_N is one of 7 rows (any one but r_P)
- c_N is one of 7 columns (any one but c_P)
- r_B is one of 6 rows (any one but r_P or r_N)
- c_B is one of 6 columns (any one but c_P or c_N)

Thus, the total number of configurations is $(8 \cdot 7 \cdot 6)^2$.

15.3.3 Permutations

A *permutation* of a set S is a sequence that contains every element of S exactly once. For example, here are all the permutations of the set $\{a, b, c\}$:

$$\begin{array}{lll} (a,b,c) & (a,c,b) & (b,a,c) \\ (b,c,a) & (c,a,b) & (c,b,a) \end{array}$$

How many permutations of an n-element set are there? Well, there are n choices for the first element. For each of these, there are $n-1$ remaining choices for the second element. For every combination of the first two elements, there are $n-2$ ways to choose the third element, and so forth. Thus, there are a total of

$$n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1 = n!$$

permutations of an n-element set. In particular, this formula says that there are

$3! = 6$ permutations of the 3-element set $\{a, b, c\}$, which is the number we found above.

Permutations will come up again in this course approximately 1.6 bazillion times. In fact, permutations are the reason why factorial comes up so often and why we taught you Stirling's approximation:

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

Oh easy to estimate Permutation

15.4 The Division Rule

Counting ears and dividing by two is a silly way to count the number of people in a room, but this approach is representative of a powerful counting principle.

A *k-to-1 function* maps exactly k elements of the domain to every element of the codomain. For example, the function mapping each ear to its owner is 2-to-1. Similarly, the function mapping each finger to its owner is 10-to-1, and the function mapping each finger and toe to its owner is 20-to-1. The general rule is:

Rule 15.4.1 (Division Rule). *If $f : A \to B$ is k-to-1, then $|A| = k \cdot |B|$.*

For example, suppose A is the set of ears in the room and B is the set of people. There is a 2-to-1 mapping from ears to people, so by the Division Rule, $|A| = 2 \cdot |B|$. Equivalently, $|B| = |A|/2$, expressing what we knew all along: the number of people is half the number of ears. Unlikely as it may seem, many counting problems are made much easier by initially counting every item multiple times and then correcting the answer using the Division Rule. Let's look at some examples.

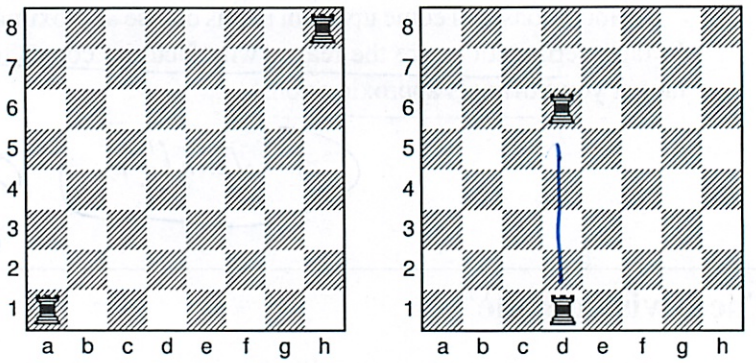
15.4.1 Another Chess Problem

In how many different ways can you place two identical rooks on a chessboard so that they do not share a row or column? A valid configuration is shown in Figure 15.2(a), and an invalid configuration is shown in Figure 15.2(b).

Let A be the set of all sequences

$$(r_1, c_1, r_2, c_2)$$

where r_1 and r_2 are distinct rows and c_1 and c_2 are distinct columns. Let B be the set of all valid rook configurations. There is a natural function f from set A to set B; in particular, f maps the sequence (r_1, c_1, r_2, c_2) to a configuration with one rook in row r_1, column c_1 and the other rook in row r_2, column c_2.



(a) valid (b) invalid

Figure 15.2 Two ways to place 2 rooks (♖) on a chessboard. The configuration in (b) is invalid because the rooks are in the same column.

But now there's a snag. Consider the sequences:

$$(1, 1, 8, 8) \qquad \text{and} \qquad (8, 8, 1, 1)$$

The first sequence maps to a configuration with a rook in the lower-left corner and a rook in the upper-right corner. The second sequence maps to a configuration with a rook in the upper-right corner and a rook in the lower-left corner. The problem is that those are two different ways of describing the *same* configuration! In fact, this arrangement is shown in Figure 15.2(a).

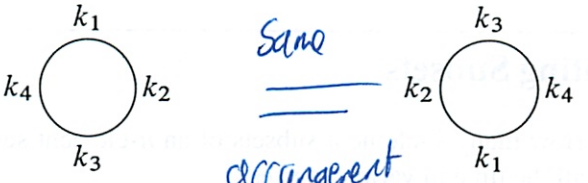
More generally, the function f maps exactly two sequences to *every* board configuration; that is f is a 2-to-1 function. Thus, by the quotient rule, $|A| = 2 \cdot |B|$. Rearranging terms gives:

$$|B| = \frac{|A|}{2} = \frac{(8 \cdot 7)^2}{2}.$$

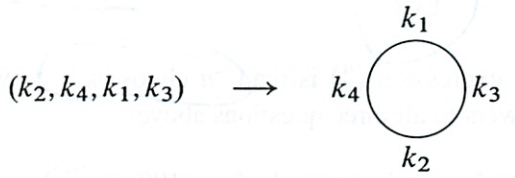
On the second line, we've computed the size of A using the General Product Rule just as in the earlier chess problem.

15.4.2 Knights of the Round Table

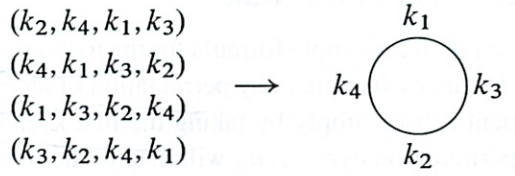
In how many ways can King Arthur arrange to seat his n different knights at his round table? Two seatings are considered to be the same *arrangement* if they yield the same sequence of knights starting at knight number 1 and going clockwise around the table. For example, the following two seatings determine the same arrangement:



So a seating is determined by the sequence of knights going clockwise around the table starting at the top seat. This means seatings are formally the same as the set, A, of all permutations of the knights. An arrangement is determined by the sequence of knights going clockwise around the table starting after knight number 1, so it is formally the same as the set, B, of all permutations of knights 2 through n. We can map each permutation in A to an arrangement in set B by seating the first knight in the permutation at the top of the table, putting the second knight to his left, the third knight to the left of the second, and so forth all the way around the table. For example:



This mapping is actually an n-to-1 function from A to B, since all n cyclic shifts of the original sequence map to the same seating arrangement. In the example, $n = 4$ different sequences map to the same seating arrangement:



Therefore, by the division rule, the number of circular seating arrangements is:

$$|B| = \frac{|A|}{n} = \frac{n!}{n} = (n-1)!$$

Note that $|A| = n!$ since there are $n!$ permutations of n knights.

15.5 Counting Subsets

How many k-element subsets of an n-element set are there? This question arises all the time in various guises:

- In how many ways can I select 5 books from my collection of 100 to bring on vacation?
- How many different 13-card Bridge hands can be dealt from a 52-card deck?
- In how many ways can I select 5 toppings for my pizza if there are 14 available toppings?

This number comes up so often that there is a special notation for it:

$$\binom{n}{k} ::= \text{the number of } k\text{-element subsets of an } n\text{-element set.}$$

The expression $\binom{n}{k}$ is read "n choose k." Now we can immediately express the answers to all three questions above:

- I can select 5 books from 100 in $\binom{100}{5}$ ways.
- There are $\binom{52}{13}$ different Bridge hands.
- There are $\binom{14}{5}$ different 5-topping pizzas, if 14 toppings are available.

15.5.1 The Subset Rule

We can derive a simple formula for the n choose k number using the Division Rule. We do this by mapping any permutation of an n-element set $\{a_1, \ldots, a_n\}$ into a k-element subset simply by taking the first k elements of the permutation. That is, the permutation $a_1 a_2 \ldots a_n$ will map to the set $\{a_1, a_2, \ldots, a_k\}$.

Notice that any other permutation with the same first k elements $a_1, \ldots, a_k$ *in any order* and the same remaining elements $n - k$ elements *in any order* will also map to this set. What's more, a permutation can only map to $\{a_1, a_2, \ldots, a_k\}$ if its first k elements are the elements $a_1, \ldots, a_k$ in some order. Since there are $k!$ possible permutations of the first k elements and $(n-k)!$ permutations of the remaining elements, we conclude from the Product Rule that exactly $k!(n-k)!$ permutations of the n-element set map to the the particular subset, S. In other words, the mapping from permutations to k-element subsets is $k!(n-k)!$-to-1.

But we know there are $n!$ permutations of an n-element set, so by the Division Rule, we conclude that

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

which proves:

Rule 15.5.1 (Subset Rule). *The number of k-element subsets of an n-element set is*

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}.$$

Notice that this works even for 0-element subsets: $n!/0!n! = 1$. Here we use the fact that 0! is a *product* of 0 terms, which by convention[2] equals 1.

15.5.2 Bit Sequences

How many n-bit sequences contain exactly k ones? We've already seen the straightforward bijection between subsets of an n-element set and n-bit sequences. For example, here is a 3-element subset of $\{x_1, x_2, \ldots, x_8\}$ and the associated 8-bit sequence:

$$\begin{array}{lllllllll} \{ & x_1, & & & x_4, & x_5 & & & \} \\ (& 1, & 0, & 0, & 1, & 1, & 0, & 0, & 0 \ ) \end{array}$$

Notice that this sequence has exactly 3 ones, each corresponding to an element of the 3-element subset. More generally, the n-bit sequences corresponding to a k-element subset will have exactly k ones. So by the Bijection Rule,

Corollary. *The number of n-bit sequences with exactly k ones is* $\binom{n}{k}$.

15.6 Sequences with Repetitions

15.6.1 Sequences of Subsets

Choosing a k-element subset of an n-element set is the same as splitting the set into a pair of subsets: the first subset of size k and the second subset consisting of the remaining $n-k$ elements. So the Subset Rule can be understood as a rule for counting the number of such splits into pairs of subsets.

[2] We don't use it here, but a *sum* of zero terms equals 0.

We can generalize this to splits into more than two subsets. Namely, let A be an n-element set and $k_1, k_2, \ldots, k_m$ be nonnegative integers whose sum is n. A $(k_1, k_2, \ldots, k_m)$-*split of* A is a sequence

$$(A_1, A_2, \ldots, A_m)$$

where the A_i are disjoint subsets of A and $|A_i| = k_i$ for $i = 1, \ldots, m$.

To count the number of splits we take the same approach as for the Subset Rule. Namely, we map any permutation $a_1 a_2 \ldots a_n$ of an n-element set A into a $(k_1, k_2, \ldots, k_m)$-split by letting the 1st subset in the split be the first k_1 elements of the permutation, the 2nd subset of the split be the next k_2 elements, ..., and the mth subset of the split be the final k_m elements of the permutation. This map is a $k_1!\, k_2! \cdots k_m!$-to-1 function from the $n!$ permutations to the $(k_1, k_2, \ldots, k_m)$-splits of A, so from the Division Rule we conclude the Subset Split Rule:

Definition 15.6.1. For $n, k_1, \ldots, k_m \in \mathbb{N}$, such that $k_1 + k_2 + \cdots + k_m = n$, define the *multinomial coefficient*

$$\binom{n}{k_1, k_2, \ldots, k_m} ::= \frac{n!}{k_1!\, k_2! \ldots k_m!}.$$

Rule 15.6.2 (Subset Split Rule). *The number of* $(k_1, k_2, \ldots, k_m)$*-splits of an* n*-element set is*

$$\binom{n}{k_1, \ldots, k_m}.$$

15.6.2 The Bookkeeper Rule

We can also generalize our count of n-bit sequences with k ones to counting sequences of n letters over an alphabet with more than two letters. For example, how many sequences can be formed by permuting the letters in the 10-letter word BOOKKEEPER?

Notice that there are 1 B, 2 O's, 2 K's, 3 E's, 1 P, and 1 R in BOOKKEEPER. This leads to a straightforward bijection between permutations of BOOKKEEPER and (1,2,2,3,1,1)-splits of $\{1, 2, \ldots, 10\}$. Namely, map a permutation to the sequence of sets of positions where each of the different letters occur.

For example, in the permutation BOOKKEEPER itself, the B is in the 1st position, the O's occur in the 2nd and 3rd positions, K's in 4th and 5th, the E's in the 6th, 7th and 9th, P in the 8th, and R is in the 10th position. So BOOKKEEPER maps to

$$(\{1\}, \{2, 3\}, \{4, 5\}, \{6, 7, 9\}, \{8\}, \{10\}).$$

From this bijection and the Subset Split Rule, we conclude that the number of ways to rearrange the letters in the word BOOKKEEPER is:

$$\frac{\overbrace{10!}^{\text{total letters}}}{\underbrace{1!}_{\text{B's}}\ \underbrace{2!}_{\text{O's}}\ \underbrace{2!}_{\text{K's}}\ \underbrace{3!}_{\text{E's}}\ \underbrace{1!}_{\text{P's}}\ \underbrace{1!}_{\text{R's}}}$$

This example generalizes directly to an exceptionally useful counting principle which we will call the

Rule 15.6.3 (Bookkeeper Rule). *Let $l_1, \ldots, l_m$ be distinct elements. The number of sequences with k_1 occurrences of l_1, and k_2 occurrences of l_2, ..., and k_m occurrences of l_m is*

$$\binom{k_1 + k_2 + \cdots + k_m}{k_1, \ldots, k_m}.$$

For example, suppose you are planning a 20-mile walk, which should include 5 northward miles, 5 eastward miles, 5 southward miles, and 5 westward miles. How many different walks are possible?

There is a bijection between such walks and sequences with 5 N's, 5 E's, 5 S's, and 5 W's. By the Bookkeeper Rule, the number of such sequences is:

$$\frac{20!}{(5!)^4}.$$

15.7 The Binomial Theorem

Counting gives insight into one of the basic theorems of algebra. A *binomial* is a sum of two terms, such as $a + b$. Now consider its 4th power, $(a + b)^4$.

If we multiply out this 4th power expression completely, we get

$$\begin{aligned}(a+b)^4 = \quad & aaaa + aaab + aaba + aabb \\ & + abaa + abab + abba + abbb \\ & + baaa + baab + baba + babb \\ & + bbaa + bbab + bbba + bbbb\end{aligned}$$

Notice that there is one term for every sequence of a's and b's. So there are 2^4 terms, and the number of terms with k copies of b and $n-k$ copies of a is:

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

by the Bookkeeper Rule. Hence, the coefficient of $a^{n-k}b^k$ is $\binom{n}{k}$. So for $n = 4$, this means:

$$(a+b)^4 = \binom{4}{0}\cdot a^4b^0 + \binom{4}{1}\cdot a^3b^1 + \binom{4}{2}\cdot a^2b^2 + \binom{4}{3}\cdot a^1b^3 + \binom{4}{4}\cdot a^0b^4$$

In general, this reasoning gives the Binomial Theorem:

Theorem 15.7.1 (Binomial Theorem). *For all $n \in \mathbb{N}$ and $a, b \in \mathbb{R}$:*

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k}b^k$$

The Binomial Theorem explains why the n choose k number is called a *binomial coefficient*.

This reasoning about binomials extends nicely to *multinomials*, which are sums of two or more terms. For example, suppose we wanted the coefficient of

$$bo^2k^2e^3pr$$

in the expansion of $(b+o+k+e+p+r)^{10}$. Each term in this expansion is a product of 10 variables where each variable is one of b, o, k, e, p, or r. Now, the coefficient of $bo^2k^2e^3pr$ is the number of those terms with exactly 1 b, 2 o's, 2 k's, 3 e's, 1 p, and 1 r. And the number of such terms is precisely the number of rearrangements of the word BOOKKEEPER:

$$\binom{10}{1,2,2,3,1,1} = \frac{10!}{1!\,2!\,2!\,3!\,1!\,1!}.$$

This reasoning extends to a general theorem.

Theorem 15.7.2 (Multinomial Theorem). *For all $n \in \mathbb{N}$,*

$$(z_1 + z_2 + \cdots + z_m)^n = \sum_{\substack{k_1,\ldots,k_m \in \mathbb{N} \\ k_1+\cdots+k_m=n}} \binom{n}{k_1, k_2, \ldots, k_m} z_1^{k_1} z_2^{k_2} \cdots z_m^{k_m}.$$

You'll be better off remembering the reasoning behind the Multinomial Theorem rather than this cumbersome formal statement.

15.8 A Word about Words

Someday you might refer to the Subset Split Rule or the Bookkeeper Rule in front of a roomful of colleagues and discover that they're all staring back at you blankly. This is not because they're dumb, but rather because we made up the name "Bookkeeper Rule". However, the rule is excellent and the name is apt, so we suggest that you play through: "You know? The Bookkeeper Rule? Don't you guys know *anything???*"

The Bookkeeper Rule is sometimes called the "formula for permutations with indistinguishable objects." The size k subsets of an n-element set are sometimes called *k-combinations*. Other similar-sounding descriptions are "combinations with repetition, permutations with repetition, r-permutations, permutations with indistinguishable objects," and so on. However, the counting rules we've taught you are sufficient to solve all these sorts of problems without knowing this jargon, so we won't burden you with it.

15.9 Counting Practice: Poker Hands

Five-Card Draw is a card game in which each player is initially dealt a *hand* consisting of 5 cards from a deck of 52 cards.[3] (Then the game gets complicated, but let's not worry about that.) The number of different hands in Five-Card Draw is the number of 5-element subsets of a 52-element set, which is

$$\binom{52}{5} = 2,598,960.$$

Let's get some counting practice by working out the number of hands with various special properties.

[3] There are 52 cards in a standard deck. Each card has a *suit* and a *rank*. There are four suits:

♠ (spades) ♡ (hearts) ♣ (clubs) ◇ (diamonds)

And there are 13 ranks, listed here from lowest to highest:

$$\overset{\text{Ace}}{A}, 2, 3, 4, 5, 6, 7, 8, 9, \overset{\text{Jack}}{J}, \overset{\text{Queen}}{Q}, \overset{\text{King}}{K}.$$

Thus, for example, 8♡ is the 8 of hearts and A♠ is the ace of spades.

15.9.1 Hands with a Four-of-a-Kind

A *Four-of-a-Kind* is a set of four cards with the same rank. How many different hands contain a Four-of-a-Kind? Here are a couple examples:

$$\{8\spadesuit, 8\diamondsuit, Q\heartsuit, 8\heartsuit, 8\clubsuit\}$$
$$\{A\clubsuit, 2\clubsuit, 2\heartsuit, 2\diamondsuit, 2\spadesuit\}$$

As usual, the first step is to map this question to a sequence-counting problem. A hand with a Four-of-a-Kind is completely described by a sequence specifying:

1. The rank of the four cards.
2. The rank of the extra card.
3. The suit of the extra card.

Thus, there is a bijection between hands with a Four-of-a-Kind and sequences consisting of two distinct ranks followed by a suit. For example, the three hands above are associated with the following sequences:

$$(8, Q, \heartsuit) \leftrightarrow \{8\spadesuit, 8\diamondsuit, 8\heartsuit, 8\clubsuit, Q\heartsuit\}$$
$$(2, A, \clubsuit) \leftrightarrow \{2\clubsuit, 2\heartsuit, 2\diamondsuit, 2\spadesuit, A\clubsuit\}$$

Clever that it is so short!

-I would have made it more complicated!

Now we need only count the sequences. There are 13 ways to choose the first rank, 12 ways to choose the second rank, and 4 ways to choose the suit. Thus, by the Generalized Product Rule, there are $13 \cdot 12 \cdot 4 = 624$ hands with a Four-of-a-Kind. This means that only 1 hand in about 4165 has a Four-of-a-Kind. Not surprisingly, Four-of-a-Kind is considered to be a very good poker hand!

15.9.2 Hands with a Full House

A *Full House* is a hand with three cards of one rank and two cards of another rank. Here are some examples:

$$\{2\spadesuit, 2\clubsuit, 2\diamondsuit, J\clubsuit, J\diamondsuit\}$$
$$\{5\diamondsuit, 5\clubsuit, 5\heartsuit, 7\heartsuit, 7\clubsuit\}$$

Again, we shift to a problem about sequences. There is a bijection between Full Houses and sequences specifying:

1. The rank of the triple, which can be chosen in 13 ways.
2. The suits of the triple, which can be selected in $\binom{4}{3}$ ways.
3. The rank of the pair, which can be chosen in 12 ways.
4. The suits of the pair, which can be selected in $\binom{4}{2}$ ways.

The example hands correspond to sequences as shown below:

$$(2, \{\spadesuit, \clubsuit, \diamondsuit\}, J, \{\clubsuit, \diamondsuit\}) \leftrightarrow \{2\spadesuit, 2\clubsuit, 2\diamondsuit, J\clubsuit, J\diamondsuit\}$$
$$(5, \{\diamondsuit, \clubsuit, \heartsuit\}, 7, \{\heartsuit, \clubsuit\}) \leftrightarrow \{5\diamondsuit, 5\clubsuit, 5\heartsuit, 7\heartsuit, 7\clubsuit\}$$

By the Generalized Product Rule, the number of Full Houses is:

$$13 \cdot \binom{4}{3} \cdot 12 \cdot \binom{4}{2}.$$

We're on a roll —but we're about to hit a speed bump.

15.9.3 Hands with Two Pairs

How many hands have *Two Pairs*; that is, two cards of one rank, two cards of another rank, and one card of a third rank? Here are examples:

$$\{3\diamondsuit, 3\spadesuit, Q\diamondsuit, Q\heartsuit, A\clubsuit\}$$
$$\{9\heartsuit, 9\diamondsuit, 5\heartsuit, 5\clubsuit, K\spadesuit\}$$

Each hand with Two Pairs is described by a sequence consisting of:

1. The rank of the first pair, which can be chosen in 13 ways.
2. The suits of the first pair, which can be selected $\binom{4}{2}$ ways.
3. The rank of the second pair, which can be chosen in 12 ways.
4. The suits of the second pair, which can be selected in $\binom{4}{2}$ ways.
5. The rank of the extra card, which can be chosen in 11 ways.
6. The suit of the extra card, which can be selected in $\binom{4}{1} = 4$ ways.

Thus, it might appear that the number of hands with Two Pairs is:

$$13 \cdot \binom{4}{2} \cdot 12 \cdot \binom{4}{2} \cdot 11 \cdot 4.$$

Wrong answer! The problem is that there is *not* a bijection from such sequences to hands with Two Pairs. This is actually a 2-to-1 mapping. For example, here are the pairs of sequences that map to the hands given above:

$$\begin{array}{ll} (3, \{\diamondsuit, \spadesuit\}, Q, \{\diamondsuit, \heartsuit\}, A, \clubsuit) & \searrow \\ & \{3\diamondsuit,\ 3\spadesuit,\ Q\diamondsuit,\ Q\heartsuit,\ A\clubsuit\} \\ (Q, \{\diamondsuit, \heartsuit\}, 3, \{\diamondsuit, \spadesuit\}, A, \clubsuit) & \nearrow \\ & \\ (9, \{\heartsuit, \diamondsuit\}, 5, \{\heartsuit, \clubsuit\}, K, \spadesuit) & \searrow \\ & \{9\heartsuit,\ 9\diamondsuit,\ 5\heartsuit,\ 5\clubsuit,\ K\spadesuit\} \\ (5, \{\heartsuit, \clubsuit\}, 9, \{\heartsuit, \diamondsuit\}, K, \spadesuit) & \nearrow \end{array}$$

The problem is that nothing distinguishes the first pair from the second. A pair of 5's and a pair of 9's is the same as a pair of 9's and a pair of 5's. We avoided this difficulty in counting Full Houses because, for example, a pair of 6's and a triple of kings is different from a pair of kings and a triple of 6's.

We ran into precisely this difficulty last time, when we went from counting arrangements of *different* pieces on a chessboard to counting arrangements of two *identical* rooks. The solution then was to apply the Division Rule, and we can do the same here. In this case, the Division rule says there are twice as many sequences as hands, so the number of hands with Two Pairs is actually:

$$\frac{13 \cdot \binom{4}{2} \cdot 12 \cdot \binom{4}{2} \cdot 11 \cdot 4}{2}.$$

Another Approach

The preceding example was disturbing! One could easily overlook the fact that the mapping was 2-to-1 on an exam, fail the course, and turn to a life of crime. You can make the world a safer place in two ways:

Or a ton of other things!

1. Whenever you use a mapping $f : A \to B$ to translate one counting problem to another, check that the same number elements in A are mapped to each element in B. If k elements of A map to each of element of B, then apply the Division Rule using the constant k.

2. As an extra check, try solving the same problem in a different way. Multiple approaches are often available —and all had better give the same answer!

(Sometimes different approaches give answers that *look* different, but turn out to be the same after some algebra.)

We already used the first method; let's try the second. There is a bijection between hands with two pairs and sequences that specify:

1. The ranks of the two pairs, which can be chosen in $\binom{13}{2}$ ways.
2. The suits of the lower-rank pair, which can be selected in $\binom{4}{2}$ ways.
3. The suits of the higher-rank pair, which can be selected in $\binom{4}{2}$ ways.
4. The rank of the extra card, which can be chosen in 11 ways.
5. The suit of the extra card, which can be selected in $\binom{4}{1} = 4$ ways.

For example, the following sequences and hands correspond:

$$(\{3, Q\}, \{\diamondsuit, \spadesuit\}, \{\diamondsuit, \heartsuit\}, A, \clubsuit) \leftrightarrow \{3\diamondsuit,\ 3\spadesuit,\ Q\diamondsuit,\ Q\heartsuit,\ A\clubsuit\}$$
$$(\{9, 5\}, \{\heartsuit, \clubsuit\}, \{\heartsuit, \diamondsuit\}, K, \spadesuit) \leftrightarrow \{9\heartsuit,\ 9\diamondsuit,\ 5\heartsuit,\ 5\clubsuit,\ K\spadesuit\}$$

Thus, the number of hands with two pairs is:

$$\binom{13}{2} \cdot \binom{4}{2} \cdot \binom{4}{2} \cdot 11 \cdot 4.$$

This is the same answer we got before, though in a slightly different form.

15.9.4 Hands with Every Suit

How many hands contain at least one card from every suit? Here is an example of such a hand:

$$\{7\diamondsuit,\ K\clubsuit,\ 3\diamondsuit,\ A\heartsuit,\ 2\spadesuit\}$$

Each such hand is described by a sequence that specifies:

1. The ranks of the diamond, the club, the heart, and the spade, which can be selected in $13 \cdot 13 \cdot 13 \cdot 13 = 13^4$ ways.
2. The suit of the extra card, which can be selected in 4 ways.
3. The rank of the extra card, which can be selected in 12 ways.

└ a specific hand
- can be multiple
- one for each

For example, the hand above is described by the sequence:

$$(7, K, A, 2, \diamondsuit, 3) \leftrightarrow \{7\diamondsuit,\ K\clubsuit,\ A\heartsuit,\ 2\spadesuit,\ 3\diamondsuit\}.$$

Are there other sequences that correspond to the same hand? There is one more! We could equally well regard either the $3\diamondsuit$ or the $7\diamondsuit$ as the extra card, so this is actually a 2-to-1 mapping. Here are the two sequences corresponding to the example hand:

$$\begin{array}{ll} (7, K, A, 2, \diamondsuit, 3) & \searrow \\ & \{7\diamondsuit,\ K\clubsuit,\ A\heartsuit,\ 2\spadesuit,\ 3\diamondsuit\} \\ (3, K, A, 2, \diamondsuit, 7) & \nearrow \end{array}$$

Therefore, the number of hands with every suit is:

$$\frac{13^4 \cdot 4 \cdot 12}{2}.$$

15.10 Inclusion-Exclusion

How big is a union of sets? For example, suppose there are 60 math majors, 200 EECS majors, and 40 physics majors. How many students are there in these three departments? Let M be the set of math majors, E be the set of EECS majors, and P be the set of physics majors. In these terms, we're asking for $|M \cup E \cup P|$.

The Sum Rule says that if M, E, and P are disjoint, then the sum of their sizes is

$$|M \cup E \cup P| = |M| + |E| + |P|.$$

However, the sets M, E, and P might *not* be disjoint. For example, there might be a student majoring in both math and physics. Such a student would be counted twice on the right side of this equation, once as an element of M and once as an element of P. Worse, there might be a triple-major[4] counted *three* times on the right side!

Our most-complicated counting rule determines the size of a union of sets that are not necessarily disjoint. Before we state the rule, let's build some intuition by considering some easier special cases: unions of just two or three sets.

[4] . . . though not at MIT anymore.

15.10.1 Union of Two Sets

For two sets, S_1 and S_2, the *Inclusion-Exclusion Rule* is that the size of their union is:

$$|S_1 \cup S_2| = |S_1| + |S_2| - |S_1 \cap S_2| \tag{15.2}$$

Intuitively, each element of S_1 is accounted for in the first term, and each element of S_2 is accounted for in the second term. Elements in *both* S_1 and S_2 are counted *twice* —once in the first term and once in the second. This double-counting is corrected by the final term.

15.10.2 Union of Three Sets

So how many students are there in the math, EECS, and physics departments? In other words, what is $|M \cup E \cup P|$ if:

$$|M| = 60$$
$$|E| = 200$$
$$|P| = 40.$$

The size of a union of three sets is given by a more complicated Inclusion-Exclusion formula:

$$\begin{aligned} |S_1 \cup S_2 \cup S_3| = {} & |S_1| + |S_2| + |S_3| \\ & - |S_1 \cap S_2| - |S_1 \cap S_3| - |S_2 \cap S_3| \\ & + |S_1 \cap S_2 \cap S_3|. \end{aligned}$$

Remarkably, the expression on the right accounts for each element in the union of S_1, S_2, and S_3 exactly once. For example, suppose that x is an element of all three sets. Then x is counted three times (by the $|S_1|$, $|S_2|$, and $|S_3|$ terms), subtracted off three times (by the $|S_1 \cap S_2|$, $|S_1 \cap S_3|$, and $|S_2 \cap S_3|$ terms), and then counted once more (by the $|S_1 \cap S_2 \cap S_3|$ term). The net effect is that x is counted just once.

If x is in two sets (say, S_1 and S_2), then x is counted twice (by the $|S_1|$ and $|S_2|$ terms) and subtracted once (by the $|S_1 \cap S_2|$ term). In this case, x does not contribute to any of the other terms, since $x \notin S_3$.

So we can't answer the original question without knowing the sizes of the various intersections. Let's suppose that there are:

4	math - EECS double majors
3	math - physics double majors
11	EECS - physics double majors
2	triple majors

Then $|M \cap E| = 4+2$, $|M \cap P| = 3+2$, $|E \cap P| = 11+2$, and $|M \cap E \cap P| = 2$. Plugging all this into the formula gives:

$$\begin{aligned} |M \cup E \cup P| &= |M| + |E| + |P| - |M \cap E| - |M \cap P| - |E \cap P| + |M \cap E \cap P| \\ &= 60 + 200 + 40 - 6 - 5 - 13 + 2 \\ &= 278 \end{aligned}$$

15.10.3 Sequences with 42, 04, or 60

In how many permutations of the set $\{0, 1, 2, \ldots, 9\}$ do either 4 and 2, 0 and 4, or 6 and 0 appear consecutively? For example, none of these pairs appears in:

$$(7, 2, 9, 5, 4, 1, 3, 8, 0, 6).$$

The 06 at the end doesn't count; we need 60. On the other hand, both 04 and 60 appear consecutively in this permutation:

$$(7, 2, 5, \underline{6}, \underline{0}, \underline{4}, 3, 8, 1, 9).$$

Let P_{42} be the set of all permutations in which 42 appears. Define P_{60} and P_{04} similarly. Thus, for example, the permutation above is contained in both P_{60} and P_{04}, but not P_{42}. In these terms, we're looking for the size of the set $P_{42} \cup P_{04} \cup P_{60}$.

First, we must determine the sizes of the individual sets, such as P_{60}. We can use a trick: group the 6 and 0 together as a single symbol. Then there is an immediate bijection between permutations of $\{0, 1, 2, \ldots 9\}$ containing 6 and 0 consecutively and permutations of:

$$\{60, 1, 2, 3, 4, 5, 7, 8, 9\}.$$

For example, the following two sequences correspond:

$$(7, 2, 5, \underline{6}, \underline{0}, 4, 3, 8, 1, 9) \longleftrightarrow (7, 2, 5, \underline{60}, 4, 3, 8, 1, 9).$$

There are 9! permutations of the set containing 60, so $|P_{60}| = 9!$ by the Bijection Rule. Similarly, $|P_{04}| = |P_{42}| = 9!$ as well.

Next, we must determine the sizes of the two-way intersections, such as $P_{42} \cap P_{60}$. Using the grouping trick again, there is a bijection with permutations of the set:

$$\{42, 60, 1, 3, 5, 7, 8, 9\}.$$

Thus, $|P_{42} \cap P_{60}| = 8!$. Similarly, $|P_{60} \cap P_{04}| = 8!$ by a bijection with the set:

$$\{604, 1, 2, 3, 5, 7, 8, 9\}.$$

And $|P_{42} \cap P_{04}| = 8!$ as well by a similar argument. Finally, note that $|P_{60} \cap P_{04} \cap P_{42}| = 7!$ by a bijection with the set:

$$\{6042, 1, 3, 5, 7, 8, 9\}.$$

does this cover every possibility?

Plugging all this into the formula gives:

$$|P_{42} \cup P_{04} \cup P_{60}| = 9! + 9! + 9! - 8! - 8! - 8! + 7!.$$

15.10.4 Union of n Sets

The size of a union of n sets is given by the following rule.

Rule 15.10.1 (Inclusion-Exclusion)**.**

$$|S_1 \cup S_2 \cup \cdots \cup S_n| =$$

	the sum of the sizes of the individual sets
minus	*the sizes of all two-way intersections*
plus	*the sizes of all three-way intersections*
minus	*the sizes of all four-way intersections*
plus	*the sizes of all five-way intersections, etc.*

weird as it gets bigger

never did in Geo 444

The formulas for unions of two and three sets are special cases of this general rule.

This way of expressing Inclusion-Exclusion is easy to understand and nearly as precise as expressing it in mathematical symbols, but we'll need the symbolic version below, so let's work on deciphering it now.

We already have a concise notation for the sum of sizes of the individual sets, namely,

$$\sum_{i=1}^{n} |S_i|.$$

A "two-way intersection" is a set of the form $S_i \cap S_j$ for $i \neq j$. We regard $S_j \cap S_i$ as the same two-way intersection as $S_i \cap S_j$, so we can assume that $i < j$. Now we can express the sum of the sizes of the two-way intersections as

$$\sum_{1 \leq i < j \leq n} |S_i \cap S_j|.$$

only want one way

Similarly, the sum of the sizes of the three-way intersections is

$$\sum_{1 \leq i < j < k \leq n} |S_i \cap S_j \cap S_k|.$$

but how to generalize further?

These sums have alternating signs in the Inclusion-Exclusion formula, with the sum of the k-way intersections getting the sign $(-1)^{k-1}$. This finally leads to a symbolic version of the rule:

Rule (Inclusion-Exclusion).

$$\begin{aligned}\left|\bigcup_{i=1}^{n} S_i\right| = &\sum_{i=1}^{n} |S_i| \\ &- \sum_{1\le i<j\le n} |S_i \cap S_j| \\ &+ \sum_{1\le i<j<k\le n} |S_i \cap S_j \cap S_k| + \cdots \\ &+ (-1)^{n-1} \left|\bigcap_{i=1}^{n} S_i\right|.\end{aligned}$$

A proof of the rule is given in Problem 15.23.

15.10.5 Computing Euler's Function

As an example, let's use Inclusion-Exclusion to calculate Euler's function, $\phi(n)$. By definition, $\phi(n)$ is the number of nonnegative integers less than a positive integer n that are relatively prime to n. But the set S of nonnegative integers less than n that are *not* relatively prime to n will be easier to count.

Suppose the prime factorization of n is $p_1^{e_1} \cdots p_m^{e_m}$ for distinct primes p_i. This means that the integers in S are precisely the nonnegative integers less than n that are divisible by at least one of the p_i's. Letting C_a be the set of nonnegative integers less than n that are divisible by a, we have

$$S = \bigcup_{i=1}^{m} C_{p_i}.$$

We'll be able to find the size of this union using Inclusion-Exclusion because the intersections of the C_p's are easy to count. For example, $C_p \cap C_q \cap C_r$ is the set of nonnegative integers less than n that are divisible by each of p, q and r. But since the p, q, r are distinct primes, being divisible by each of them is the same as being divisible by their product. Now observe that if k is a positive divisor of n, then exactly n/k nonnegative integers less than n are divisible by k, namely, $0, k, 2k, \ldots, ((n/k)-1)k$. So exactly n/pqr nonnegative integers less than n are divisible by all three primes p, q, r. In other words,

$$|C_p \cap C_q \cap C_r| = \frac{n}{pqr}.$$

Reasoning this way about all the intersections among the C_p's and applying Inclusion-Exclusion, we get

$$\begin{aligned}
|S| &= \left|\bigcup_{i=1}^{m} C_{p_i}\right| \\
&= \sum_{i=1}^{m} |C_{p_i}| - \sum_{1\le i<j\le m} |C_{p_i} \cap C_{p_j}| \\
&\quad + \sum_{1\le i<j<k\le m} |C_{p_i} \cap C_{p_j} \cap C_{p_k}| - \cdots + (-1)^{m-1} \left|\bigcap_{i=1}^{m} C_{p_i}\right| \\
&= \sum_{i=1}^{m} \frac{n}{p_i} - \sum_{1\le i<j\le m} \frac{n}{p_i p_j} \\
&\quad + \sum_{1\le i<j<k\le m} \frac{n}{p_i p_j p_k} - \cdots + (-1)^{m-1} \frac{n}{p_1 p_2 \cdots p_n} \\
&= n\left(\sum_{i=1}^{m} \frac{1}{p_i} - \sum_{1\le i<j\le m} \frac{1}{p_i p_j} + \sum_{1\le i<j<k\le m} \frac{1}{p_i p_j p_k} - \cdots + (-1)^{m-1} \frac{1}{p_1 p_2 \cdots p_n}\right)
\end{aligned}$$

But $\phi(n) = n - |S|$ by definition, so

$$\begin{aligned}
\phi(n) &= n\left(1 - \sum_{i=1}^{m} \frac{1}{p_i} + \sum_{1\le i<j\le m} \frac{1}{p_i p_j} - \sum_{1\le i<j<k\le m} \frac{1}{p_i p_j p_k} + \cdots + (-1)^{m} \frac{1}{p_1 p_2 \cdots p_n}\right) \\
&= n \prod_{i=1}^{m} \left(1 - \frac{1}{p_i}\right). \qquad (15.3)
\end{aligned}$$

Yikes! That was pretty hairy. Are you getting tired of all that nasty algebra? If so, then good news is on the way. In the next section, we will show you how to prove some heavy-duty formulas without using any algebra at all. Just a few words and you are done. No kidding.

15.11 Combinatorial Proofs

Suppose you have n different T-shirts, but only want to keep k. You could equally well select the k shirts you want to keep or select the complementary set of $n-k$ shirts you want to throw out. Thus, the number of ways to select k shirts from

among n must be equal to the number of ways to select $n-k$ shirts from among n. Therefore:

$$\binom{n}{k} = \binom{n}{n-k}.$$

This is easy to prove algebraically, since both sides are equal to:

$$\frac{n!}{k!\,(n-k)!}.$$

But we didn't really have to resort to algebra; we just used counting principles.

Hmmm....

15.11.1 Pascal's Identity

Ali, famed Math for Computer Science Teaching Assistant, has decided to try out for the US Olympic boxing team. After all, he's watched all of the *Rocky* movies and spent hours in front of a mirror sneering, "Yo, you wanna piece a' *me*?!" Ali figures that n people (including himself) are competing for spots on the team and only k will be selected. As part of maneuvering for a spot on the team, he needs to work out how many different teams are possible. There are two cases to consider:

- Ali *is* selected for the team, and his $k-1$ teammates are selected from among the other $n-1$ competitors. The number of different teams that can be formed in this way is:
$$\binom{n-1}{k-1}.$$
- Ali is *not* selected for the team, and all k team members are selected from among the other $n-1$ competitors. The number of teams that can be formed this way is:
$$\binom{n-1}{k}.$$

All teams of the first type contain Ali, and no team of the second type does; therefore, the two sets of teams are disjoint. Thus, by the Sum Rule, the total number of possible Olympic boxing teams is:

$$\binom{n-1}{k-1} + \binom{n-1}{k}.$$

Oscar, equally-famed Teaching Assistant, thinks Ali isn't so tough and so he might as well also try out. He reasons that n people (including himself) are trying out for k spots. Thus, the number of ways to select the team is simply:

$$\binom{n}{k}.$$

Oscar and Ali each correctly counted the number of possible boxing teams. Thus, their answers must be equal. So we know:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}.$$

This is called *Pascal's Identity*. And we proved it *without any algebra!* Instead, we relied purely on counting techniques.

15.11.2 Finding a Combinatorial Proof

A *combinatorial proof* is an argument that establishes an algebraic fact by relying on counting principles. Many such proofs follow the same basic outline:

1. Define a set S.
2. Show that $|S| = n$ by counting one way.
3. Show that $|S| = m$ by counting another way.
4. Conclude that $n = m$.

In the preceding example, S was the set of all possible Olympic boxing teams. Ali computed

$$|S| = \binom{n-1}{k-1} + \binom{n-1}{k}$$

by counting one way, and Oscar computed

$$|S| = \binom{n}{k}$$

by counting another way. Equating these two expressions gave Pascal's Identity.

More typically, the set S is defined in terms of simple sequences or sets rather than an elaborate story. Here is another colorful example of a combinatorial argument.

Theorem 15.11.1.

$$\sum_{r=0}^{n} \binom{n}{r}\binom{2n}{n-r} = \binom{3n}{n}$$

Proof. We give a combinatorial proof. Let S be all n-card hands that can be dealt from a deck containing n differnet red cards and $2n$ different black cards. First, note that every $3n$-element set has

$$|S| = \binom{3n}{n}$$

n-element subsets.

From another perspective, the number of hands with exactly r red cards is

$$\binom{n}{r}\binom{2n}{n-r}$$

since there are $\binom{n}{r}$ ways to choose the r red cards and $\binom{2n}{n-r}$ ways to choose the $n-r$ black cards. Since the number of red cards can be anywhere from 0 to n, the total number of n-card hands is:

$$|S| = \sum_{r=0}^{n} \binom{n}{r}\binom{2n}{n-r}.$$

Equating these two expressions for $|S|$ proves the theorem. ■

Combinatorial proofs are almost magical. Theorem 15.11.1 looks pretty scary, but we proved it without any algebraic manipulations at all. The key to constructing a combinatorial proof is choosing the set S properly, which can be tricky. Generally, the simpler side of the equation should provide some guidance. For example, the right side of Theorem 15.11.1 is $\binom{3n}{n}$, which suggests that it will be helpful to choose S to be all n-element subsets of some $3n$-element set.

15.12 The Pigeonhole Principle

Here is an old puzzle:

> A drawer in a dark room contains red socks, green socks, and blue socks. How many socks must you withdraw to be sure that you have a matching pair?

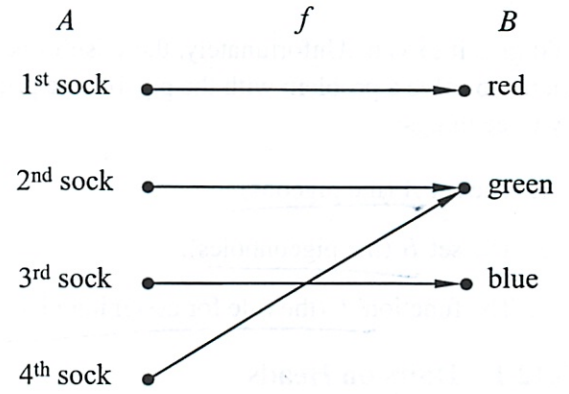


Figure 15.3 One possible mapping of four socks to three colors.

For example, picking out three socks is not enough; you might end up with one red, one green, and one blue. The solution relies on the

Pigeonhole Principle

If there are more pigeons than holes they occupy, then at least two pigeons must be in the same hole.

What pigeons have to do with selecting footwear under poor lighting conditions may not be immediately obvious, but if we let socks be pigeons and the colors be three pigeonholes, then as soon as you pick four socks, there are bound to be two in the same hole, that is, with the same color. So four socks are enough to ensure a matched pair. For example, one possible mapping of four socks to three colors is shown in Figure 15.3.

A rigorous statement of the Principle goes this way:

Rule 15.12.1 (Pigeonhole Principle). *If $|A| > |B|$, then for every total function $f : A \to B$, there exist two different elements of A that are mapped by f to the same element of B.*

Stating the Principle this way may be less intuitive, but it should now sound familiar: it is simply the contrapositive of the Mapping Rules injective case (5.2). Here, the pigeons form set A, the pigeonholes are the set B, and f describes which hole each pigeon occupies.

Mathematicians have come up with many ingenious applications for the pigeonhole principle. If there were a cookbook procedure for generating such arguments,

we'd give it to you. Unfortunately, there isn't one. One helpful tip, though: when you try to solve a problem with the pigeonhole principle, the key is to clearly identify three things:

1. The set A (the pigeons).
2. The set B (the pigeonholes).
3. The function f (the rule for assigning pigeons to pigeonholes).

15.12.1 Hairs on Heads

There are a number of generalizations of the pigeonhole principle. For example:

Rule 15.12.2 (Generalized Pigeonhole Principle). *If $|A| > k \cdot |B|$, then every total function $f : A \to B$ maps at least $k+1$ different elements of A to the same element of B.*

For example, if you pick two people at random, surely they are extremely unlikely to have *exactly* the same number of hairs on their heads. However, in the remarkable city of Boston, Massachusetts there are actually *three* people who have exactly the same number of hairs! Of course, there are many bald people in Boston, and they all have zero hairs. But we're talking about non-bald people; say a person is non-bald if they have at least ten thousand hairs on their head.

Boston has about 500,000 non-bald people, and the number of hairs on a person's head is at most 200,000. Let A be the set of non-bald people in Boston, let $B = \{10,000, 10,001, \ldots, 200,000\}$, and let f map a person to the number of hairs on his or her head. Since $|A| > 2|B|$, the Generalized Pigeonhole Principle implies that at least three people have exactly the same number of hairs. We don't know who they are, but we know they exist!

15.12.2 Subsets with the Same Sum

For your reading pleasure, we have displayed ninety 25-digit numbers in Figure 15.4. Are there two different subsets of these 25-digit numbers that have the same sum? For example, maybe the sum of the last ten numbers in the first column is equal to the sum of the first eleven numbers in the second column?

Finding two subsets with the same sum may seem like a silly puzzle, but solving these sorts of problems turns out to be useful in diverse applications such as finding good ways to fit packages into shipping containers and decoding secret messages.

It turns out that it is hard to find different subsets with the same sum, which is why this problem arises in cryptography. But it is easy to prove that two such subsets *exist*. That's where the Pigeonhole Principle comes in.

00204801353855029644480 38	31710048321735013941130 17
57632573310834796474093 98	82473310000429953116460 21
04894459918669156762409 92	32082344215973686470192 65
58009491235489891226286 63	84962439971234759227663 10
10826620324303796513709 81	34372546563551578648691 13
60429008011992802180260 01	85183991406760026607474 77
11784808947697061789949 93	35748833930586539237113 65
61161717891377378967014 05	85436912834701914523337 63
12531273516832396938513 27	36449099460404801899691 49
61448689730015823697235 12	86753092583741370924613 52
13015051292340778110690 11	37900441327370840944172 46
62473145938511692347461 52	86943211123639968672966 65
13115671111438664338821 94	38703321274379713553228 15
68144289442668749634882 74	87723212036084772458511 54
14700294527212035876862 14	40805058045778014513631 00
68708529455438868491478 81	87914221617225825463410 91
15782710472862574994338 86	41672834610257023481249 20
69149555081209500937323 97	90626280245921262839732 85
16382439218521762431923 54	42359968311237777882112 49
69496324513659871524235 41	91378455669255263498977 94
17635802191319859631023 65	46709394457494390421112 20
71282111436136198284156 50	91537629668031892919344 19
18262277956018422310296 94	48153793518653842796134 27
71739200836518623079253 94	92708801940776364069842 49
18439718626751020372014 20	48370529482129226044421 90
72156548742117556762205 87	93243014807221034903792 04
23969511937221345261772 37	51063894238550185506715 30
72569328471643910402330 50	94360908321466951471405 81
27813945682685998010963 54	51423681920047692180699 10
73328226570752354316203 17	94753081597345382490132 38
27966051967136104054080 19	51812340961301440840418 56
74264418295415734449641 39	94923766239174869749232 02
29310163947619752631903 47	51982673981256179943913 48
76321981265318093271863 21	95119725587798802882529 79
29334580582944051551972 96	53175929403162312197583 72
77121544322119128823105 11	96024134246191871125522 64
30755144104909759203153 48	53843581267717941283569 47
78589186642402623566100 10	96312171149061292194611 11
81494367168713711619320 35	31576931053251112843219 93
31114749852527934528600 17	54392117122489019954234 41
78981567867632129631786 79	99081898531027533359813 19
31456215879361201184387 01	56103798260928381927604 58
81475910170375733378486 16	99132374763417642998139 87
31489012556288811031985 49	56323175554652286776760 44
56921683746370196174237 12	81760638316825365713067 91

Figure 15.4 Ninety 25-digit numbers. Can you find two different subsets of these numbers that have the same sum?

Let A be the collection of all subsets of the 90 numbers in the list. Now the sum of any subset of numbers is at most $90 \cdot 10^{25}$, since there are only 90 numbers and every 25-digit number is less than 10^{25}. So let B be the set of integers $\{0, 1, \ldots, 90 \cdot 10^{25}\}$, and let f map each subset of numbers (in A) to its sum (in B).

We proved that an n-element set has 2^n different subsets in Section 15.2. Therefore:

$$|A| = 2^{90} \geq 1.237 \times 10^{27}$$

On the other hand:

$$|B| = 90 \cdot 10^{25} + 1 \leq 0.901 \times 10^{27}.$$

Both quantities are enormous, but $|A|$ is a bit greater than $|B|$. This means that f maps at least two elements of A to the same element of B. In other words, by the Pigeonhole Principle, two different subsets must have the same sum!

Notice that this proof gives no indication *which* two sets of numbers have the same sum. This frustrating variety of argument is called a *nonconstructive proof*.

The $100 prize for two same-sum subsets

To see if was possible to actually *find* two different subsets of the ninety 25-digit numbers with the same sum, we offered a $100 prize to the first student who did it. We didn't expect to have to pay off this bet, but we underestimated the ingenuity and initiative of the students. One computer science major wrote a program that cleverly searched only among a reasonably small set of "plausible" sets, sorted them by their sums, and actually found a couple with the same sum. He won the prize. A few days later, a math major figured out how to reformulate the sum problem as a "lattice basis reduction" problem; then he found a software package implementing an efficient basis reduction procedure, and using it, he very quickly found lots of pairs of subsets with the same sum. He didn't win the prize, but he got a standing ovation from the class —staff included.

The $500 Prize for Sets with Distinct Subset Sums

How can we construct a set of n positive integers such that all its subsets have *distinct* sums? One way is to use powers of two:

$$\{1, 2, 4, 8, 16\}$$

This approach is so natural that one suspects all other such sets must involve larger numbers. (For example, we could safely replace 16 by 17, but not by 15.) Remarkably, there are examples involving *smaller* numbers. Here is one:

$$\{6, 9, 11, 12, 13\}$$

One of the top mathematicians of the Twentieth Century, Paul Erdős, conjectured in 1931 that there are no such sets involving *significantly* smaller numbers. More precisely, he conjectured that the largest number in such a set must be greater than $c2^n$ for some constant $c > 0$. He offered $500 to anyone who could prove or disprove his conjecture, but the problem remains unsolved.

15.13 A Magic Trick

There is a Magician and an Assistant. The Assistant goes into the audience with a deck of 52 cards while the Magician looks away.

Five audience members each select one card from the deck. The Assistant then gathers up the five cards and holds up four of them so the Magician can see them. The Magician concentrates for a short time and then correctly names the secret, fifth card!

Since we don't really believe the Magician can read minds, we know the Assistant has somehow communicated the secret card to the Magician. Since real Magicians and Assistants are not to be trusted, we can expect that the Assistant would illegitimately signal the Magician with coded phrases or body language, but they don't have to cheat in this way. In fact, the Magician and Assistant could be kept out of sight of each other while some audience member holds up the 4 cards designated by the Assistant for the Magician to see.

Of course, without cheating, there is still an obvious way the Assistant can communicate to the Magician: he can choose any of the $4! = 24$ permutations of the 4 cards as the order in which to hold up the cards. However, this alone won't

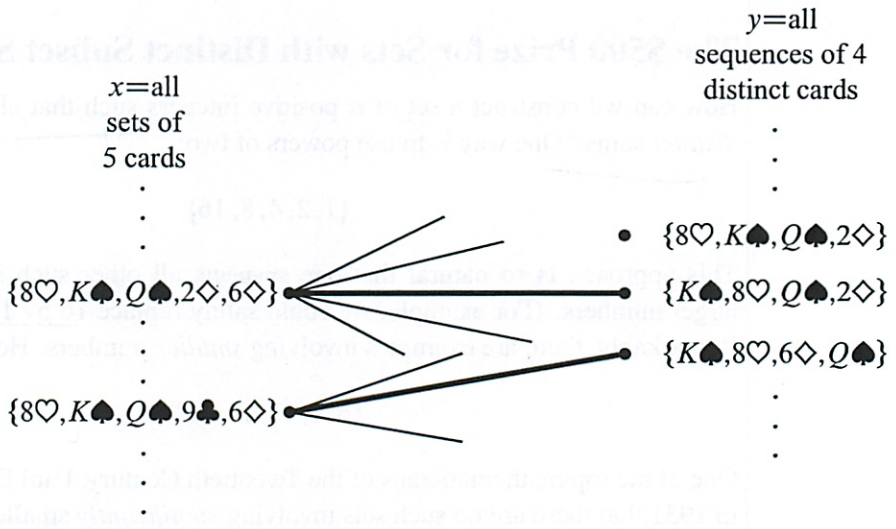


Figure 15.5 The bipartite graph where the nodes on the left correspond to *sets* of 5 cards and the nodes on the right correspond to *sequences* of 4 cards. There is an edge between a set and a sequence whenever all the cards in the sequence are contained in the set.

quite work: there are 48 cards remaining in the deck, so the Assistant doesn't have enough choices of orders to indicate exactly what the secret card is (though he could narrow it down to two cards).

15.13.1 The Secret

The method the Assistant can use to communicate the fifth card exactly is a nice application of what we know about counting and matching.

The Assistant has a second legitimate way to communicate: he can choose *which of the five cards to keep hidden*. Of course, it's not clear how the Magician could determine which of these five possibilities the Assistant selected by looking at the four visible cards, but there is a way, as we'll now explain.

The problem facing the Magician and Assistant is actually a bipartite matching problem. Put all the *sets* of 5 cards in a collection X on the left. And put all the *sequences* of 4 distinct cards in a collection Y on the right. These are the two sets of vertices in the bipartite graph. There is an edge between a set of 5 cards and a sequence of 4 if every card in the sequence is also in the set. In other words, if the audience selects a set of 5 cards, then the Assistant must reveal a sequence of 4 cards that is adjacent in the bipartite graph. Some edges are shown in the diagram in Figure 15.5.

For example,

$$\{8\heartsuit, K\spadesuit, Q\spadesuit, 2\diamondsuit, 6\diamondsuit\} \tag{15.4}$$

is an element of X on the left. If the audience selects this set of 5 cards, then there are many different 4-card sequences on the right in set Y that the Assistant could choose to reveal, including $(8\heartsuit, K\spadesuit, Q\spadesuit, 2\diamondsuit)$, $(K\spadesuit, 8\heartsuit, Q\spadesuit, 2\diamondsuit)$, and $(K\spadesuit, 8\heartsuit, 6\diamondsuit, Q\spadesuit)$.

What the Magician and his Assistant need to perform the trick is a *matching* for the X vertices. If they agree in advance on some matching, then when the audience selects a set of 5 cards, the Assistant reveals the matching sequence of 4 cards. The Magician uses the matching to find the audience's chosen set of 5 cards, and so he can name the one not already revealed.

For example, suppose the Assistant and Magician agree on a matching containing the two bold edges in Figure 15.5. If the audience selects the set

$$\{8\heartsuit, K\spadesuit, Q\spadesuit, 9\clubsuit, 6\diamondsuit\}, \tag{15.5}$$

then the Assistant reveals the corresponding sequence

$$(K\spadesuit, 8\heartsuit, 6\diamondsuit, Q\spadesuit). \tag{15.6}$$

Using the matching, the Magician sees that the hand (15.5) is matched to the sequence (15.6), so he can name the one card in the corresponding set not already revealed, namely, the $9\clubsuit$. Notice that the fact that the sets are *matched*, that is, that different sets are paired with *distinct* sequences, is essential. For example, if the audience picked the previous hand (15.4), it would be possible for the Assistant to reveal the same sequence (15.6), but he better not do that; if he did, then the Magician would have no way to tell if the remaining card was the $9\clubsuit$ or the $2\diamondsuit$.

So how can we be sure the needed matching can be found? The answer is that each vertex on the left has degree $5 \cdot 4! = 120$, since there are five ways to select the card kept secret and there are 4! permutations of the remaining 4 cards. In addition, each vertex on the right has degree 48, since there are 48 possibilities for the fifth card. So this graph is *degree-constrained* according to Definition 11.5.5, and so has a matching by Theorem 11.5.6.

In fact, this reasoning shows that the Magician could still pull off the trick if 120 cards were left instead of 48, that is, the trick would work with a deck as large as 124 different cards —without any magic!

15.13.2 The Real Secret

But wait a minute! It's all very well in principle to have the Magician and his Assistant agree on a matching, but how are they supposed to remember a matching

with $\binom{52}{5} = 2,598,960$ edges? For the trick to work in practice, there has to be a way to match hands and card sequences mentally and on the fly.

We'll describe how in lecture....

15.13.3 The Same Trick with Four Cards?

Suppose that the audience selects only *four* cards and the Assistant reveals a sequence of *three* to the Magician. Can the Magician determine the fourth card?

Let X be all the sets of four cards that the audience might select, and let Y be all the sequences of three cards that the Assistant might reveal. Now, on one hand, we have

$$|X| = \binom{52}{4} = 270,725$$

by the Subset Rule. On the other hand, we have

$$|Y| = 52 \cdot 51 \cdot 50 = 132,600$$

by the Generalized Product Rule. Thus, by the Pigeonhole Principle, the Assistant must reveal the *same* sequence of three cards for at least

$$\left\lceil \frac{270,725}{132,600} \right\rceil = 3$$

different four-card hands. This is bad news for the Magician: if he sees that sequence of three, then there are at least three possibilities for the fourth card which he cannot distinguish. So there is no legitimate way for the Assistant to communicate exactly what the fourth card is!

Problems for Section 15.2

Practice Problems

Problem 15.1.
How many ways are there to select k out of n books on a shelf so that there are always at least 3 unselected books between selected books? (Assume n is large enough for this to be possible.)

Class Problems

Problem 15.2.
A license plate consists of either:

- 3 letters followed by 3 digits (standard plate)

- 5 letters (vanity plate)
- 2 characters—letters or numbers (big shot plate)

Let L be the set of all possible license plates.

(a) Express L in terms of

$$\mathcal{A} = \{A, B, C, \ldots, Z\}$$
$$\mathcal{D} = \{0, 1, 2, \ldots, 9\}$$

using unions ($\cup$) and set products ($\times$).

(b) Compute $|L|$, the number of different license plates, using the sum and product rules.

Problem 15.3. **(a)** How many of the billion numbers in the range from 1 to 10^9 contain the digit 1? (*Hint:* How many don't?)

(b) There are 20 books arranged in a row on a shelf. Describe a bijection between ways of choosing 6 of these books so that no two adjacent books are selected and 15-bit strings with exactly 6 ones.

Problem 15.4.

(a) Let $\mathcal{S}_{n,k}$ be the possible nonnegative integer solutions to the inequality

$$x_1 + x_2 + \cdots + x_k \leq n. \tag{15.7}$$

That is

$$\mathcal{S}_{n,k} ::= \{(x_1, x_2, \ldots, x_k) \in \mathbb{N}^k \mid (15.7) \text{ is true}\}.$$

Describe a bijection between $\mathcal{S}_{n,k}$ and the set of binary strings with n zeroes and k ones.

(b) Let $\mathcal{L}_{n,k}$ be the length k weakly increasing sequences of nonnegative integers $\leq n$. That is

$$\mathcal{L}_{n,k} ::= \{(y_1, y_2, \ldots, y_k) \in \mathbb{N}^k \mid y_1 \leq y_2 \leq \cdots \leq y_k \leq n\}.$$

Describe a bijection between $\mathcal{L}_{n,k}$ and $\mathcal{S}_{n,k}$.

Problem 15.5.

An n-vertex *numbered tree* is a tree whose vertex set is $\{1, 2, \ldots, n\}$ for some $n > 2$. We define the *code* of the numbered tree to be a sequence of $n - 2$ integers from 1 to n obtained by the following recursive process:[5]

> If there are more than two vertices left, write down the *father* of the largest leaf, delete this *leaf*, and continue this process on the resulting smaller tree.
>
> If there are only two vertices left, then stop—the code is complete.

For example, the codes of a couple of numbered trees are shown in the Figure 15.6.

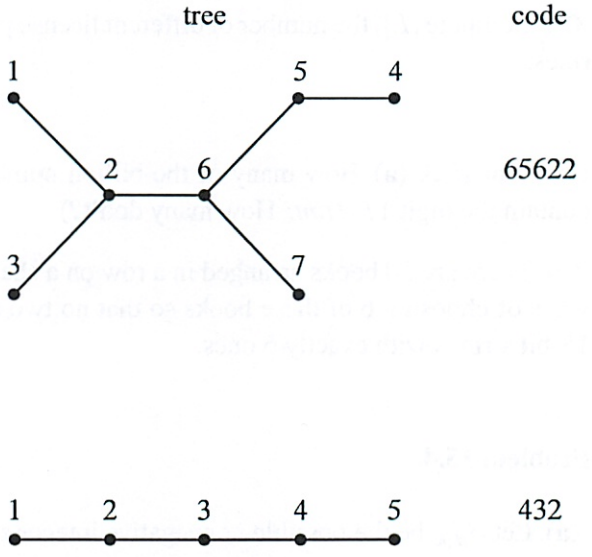


Figure 15.6

(a) Describe a procedure for reconstructing a numbered tree from its code.

(b) Conclude there is a bijection between the n-vertex numbered trees and $\{1, \ldots, n\}^{n-2}$, and state how many n-vertex numbered trees there are.

Homework Problems

Problem 15.6.

Answer the following questions with a number or a simple formula involving factorials and binomial coefficients. Briefly explain your answers.

[5]The necessarily unique node adjacent to a leaf is called its *father*.

(a) How many ways are there to order the 26 letters of the alphabet so that no two of the vowels `a`, `e`, `i`, `o`, `u` appear consecutively and the last letter in the ordering is not a vowel?

Hint: Every vowel appears to the left of a consonant.

(b) How many ways are there to order the 26 letters of the alphabet so that there are *at least two* consonants immediately following each vowel?

(c) In how many different ways can $2n$ students be paired up?

(d) Two n-digit sequences of digits 0,1,...,9 are said to be of the *same type* if the digits of one are a permutation of the digits of the other. For $n = 8$, for example, the sequences `03088929` and `00238899` are the same type. How many types of n-digit integers are there?

Problem 15.7.

In a standard 52-card deck, each card has one of thirteen *rank*s in the set, R, and one of four *suit*s in the set, S, where

$$R ::= \{A, 2, \ldots, 10, J, Q, K\},$$
$$S ::= \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}.$$

A 5-card *hand* is a set of five distinct cards from the deck.

For each part describe a bijection between a set that can easily be counted using the Product and Sum Rules of Ch. 15.1, and the set of hands matching the specification. *Give bijections, not numerical answers.*

For instance, consider the set of 5-card hands containing all 4 suits. Each such hand must have 2 cards of one suit. We can describe a bijection between such hands and the set $S \times R_2 \times R^3$ where R_2 is the set of two-element subsets of R. Namely, an element

$$(s, \{r_1, r_2\}, (r_3, r_4, r_5)) \in S \times R_2 \times R^3$$

indicates

1. the repeated suit, $s \in S$,
2. the set, $\{r_1, r_2\} \in R_2$, of ranks of the cards of suit, s, and
3. the ranks (r_3, r_4, r_5) of remaining three cards, listed in increasing suit order where $\clubsuit \prec \diamondsuit \prec \heartsuit \prec \spadesuit$.

For example,

$$(\clubsuit, \{10, A\}, (J, J, 2)) \longleftrightarrow \{A\clubsuit, 10\clubsuit, J\diamondsuit, J\heartsuit, 2\spadesuit\}.$$

(a) A single pair of the same rank (no 3-of-a-kind, 4-of-a-kind, or second pair).

(b) Three or more aces.

Problems for Section 15.4

Class Problems

Problem 15.8.

Your class tutorial has 12 students, who are supposed to break up into 4 groups of 3 students each. Your Teaching Assistant (TA) has observed that the students waste too much time trying to form balanced groups, so they decided to pre-assign students to groups and email the group assignments to his students.

(a) Your TA has a list of the 12 students in front of him, so they divides the list into consecutive groups of 3. For example, if the list is ABCDEFGHIJKL, the TA would define a sequence of four groups to be $(\{A, B, C\}, \{D, E, F\}, \{G, H, I\}, \{J, K, L\})$. This way of forming groups defines a mapping from a list of twelve students to a sequence of four groups. This is a k-to-1 mapping for what k?

(b) A group assignment specifies which students are in the same group, but not any order in which the groups should be listed. If we map a sequence of 4 groups,

$$(\{A, B, C\}, \{D, E, F\}, \{G, H, I\}, \{J, K, L\}),$$

into a group assignment

$$\{\{A, B, C\}, \{D, E, F\}, \{G, H, I\}, \{J, K, L\}\},$$

this mapping is j-to-1 for what j?

(c) How many group assignments are possible?

(d) In how many ways can $3n$ students be broken up into n groups of 3?

Problem 15.9.

A pizza house is having a promotional sale. Their commercial reads:

> We offer 9 different toppings for your pizza! Buy 3 large pizzas at the regular price, and you can get each one with as many different toppings as you wish, absolutely free. That's 22,369,621 different ways to choose your pizzas!

The ad writer was a former Harvard student who had evaluated the formula $(2^9)^3/3!$ on his calculator and gotten close to $22,369,621$. Unfortunately, $(2^9)^3/3!$ is obviously not an integer, so clearly something is wrong. What mistaken reasoning might have led the ad writer to this formula? Explain how to fix the mistake and get a correct formula.

Problem 15.10.
Answer the following quesions using the Generalized Product Rule.

(a) Next week, I'm going to get really fit! On day 1, I'll exercise for 5 minutes. On each subsequent day, I'll exercise 0, 1, 2, or 3 minutes more than the previous day. For example, the number of minutes that I exercise on the seven days of next week might be 5, 6, 9, 9, 9, 11, 12. How many such sequences are possible?

(b) An *r-permutation* of a set is a sequence of r distinct elements of that set. For example, here are all the 2-permutations of $\{a, b, c, d\}$:

$$\begin{array}{lll} (a,b) & (a,c) & (a,d) \\ (b,a) & (b,c) & (b,d) \\ (c,a) & (c,b) & (c,d) \\ (d,a) & (d,b) & (d,c) \end{array}$$

How many r-permutations of an n-element set are there? Express your answer using factorial notation.

(c) How many $n \times n$ matrices are there with *distinct* entries drawn from $\{1, \ldots, p\}$, where $p \geq n^2$?

Problem 15.11. **(a)** There are 30 books arranged in a row on a shelf. In how many ways can eight of these books be selected so that there are at least two unselected books between any two selected books?

(b) How many nonnegative integer solutions are there for the following equality?

$$x_1 + x_2 + \cdots + x_m = k. \tag{15.8}$$

(c) How many nonnegative integer solutions are there for the following inequality?

$$x_1 + x_2 + \cdots + x_m \leq k. \tag{15.9}$$

(d) How many length-m weakly increasing sequences of nonnegative integers $\leq k$ are there?

Exam Problems

Problem 15.12.
Suppose that two identical 52-card decks are mixed together. Write a simple formula for the number of 104-card double-deck mixes that are possible.

Problems for Section 15.7

Practice Problems

Problem 15.13.
Find the coefficients of $x^{10}y^5$ in $(19x + 4y)^{15}$

Class Problems

Problem 15.14.
Find the coefficients of

(a) x^5 in $(1 + x)^{11}$

(b) x^8y^9 in $(3x + 2y)^{17}$

(c) a^6b^6 in $(a^2 + b^3)^5$

Problem 15.15. **(a)** Use the Multinomial Theorem 15.7.2 to prove that

$$(x_1 + x_2 + \cdots + x_n)^p \equiv x_1^p + x_2^p + \cdots + x_n^p \pmod{p} \tag{15.10}$$

for all primes p. (Do not prove it using Fermat's "little" Theorem. The point of this problem is to offer an independent proof of Fermat's theorem.)

Hint: Explain why $\binom{p}{k_1,k_2,\ldots,k_n}$ is divisible by p if all the k_i's are positive integers less than p.

(b) Explain how (15.10) immediately proves Fermat's Little Theorem 8.6.4: $n^{p-1} \equiv 1 \pmod{p}$ when n is not a multiple of p.

Homework Problems

Problem 15.16.
The *degree sequence* of a simple graph is the weakly decreasing sequence of degrees of its vertices. For example, the degree sequence for the 5-vertex numbered tree pictured in the Figure 15.6 in Problem 15.5 is $(2, 2, 2, 1, 1)$ and for the 7-vertex tree it is $(3, 3, 2, 1, 1, 1, 1)$.

We're interested in counting how many numbered trees there are with a given degree sequence. We'll do this using the bijection defined in Problem 15.5 between n-vertex numbered trees and length $n-2$ code words whose characters are integers between 1 and n.

The *occurrence number* for a character in a word is the number of times that the character occurs in the word. For example, in the word `65622`, the occurrence number for `6` is two, and the occurrence number for `5` is one. The *occurrence sequence* of a word is the weakly decreasing sequence of occurrence numbers of characters in the word. The occurrence sequence for this word is $(2, 2, 1)$ because it has two occurrences of each of the characters `6` and `2`, and one occurrence of `5`.

(a) There is simple relationship between the degree sequence of an n-vertex numbered tree and the occurrence sequence of its code. Describe this relationship and explain why it holds. Conclude that counting n-vertex numbered trees with a given degree sequence is the same as counting the number of length $n-2$ code words with a given occurrence sequence.

Hint: How many times does a vertex of degree, d, occur in the code?

For simplicity, let's focus on counting 9-vertex numbered trees with a given degree sequence. By part (a), this is the same as counting the number of length 7 code words with a given occurrence sequence.

Any length 7 code word has a *pattern*, which is another length 7 word over the alphabet `a,b,c,d,e,f,g` that has the same occurrence sequence.

(b) How many length 7 patterns are there with three occurrences of `a`, two occurrences of `b`, and one occurrence of `c` and `d`?

(c) How many ways are there to assign occurrence numbers to integers $1, 2, \ldots, 9$ so that a code word with those occurrence numbers would have the occurrence sequence $3, 2, 1, 1, 0, 0, 0, 0, 0$?

In general, to find the pattern of a code word, list its characters in decreasing order by *number of occurrences*, and list characters with the same number of occurrences in decreasing order. Then replace successive characters in the list by successive letters `a,b,c,d,e,f,g`. The code word `2468751`, for example, has the pattern `fecabdg`, which is obtained by replacing its characters `8,7,6,5,4,2,1` by `a,b,c,d,e,f,g`, respectively. The code word `2449249` has pattern `caabcab`, which is obtained by replacing its characters `4,9,2` by `a,b,c`, respectively.

(d) What length 7 code word has three occurrences of `7`, two occurrences of `8`, one occurrence each of `2` and `9`, and pattern `abacbad`?

(e) Explain why the number of 9-vertex numbered trees with degree sequence $(4, 3, 2, 2, 1, 1, 1, 1, 1)$ is the product of the answers to parts (b) and (c).

Problems for Section 15.8

Class Problems

Problem 15.17.

The Tao of BOOKKEEPER: we seek enlightenment through contemplation of the word $BOOKKEEPER$.

(a) In how many ways can you arrange the letters in the word $POKE$?

(b) In how many ways can you arrange the letters in the word BO_1O_2K? Observe that we have subscripted the O's to make them distinct symbols.

(c) Suppose we map arrangements of the letters in BO_1O_2K to arrangements of the letters in $BOOK$ by erasing the subscripts. Indicate with arrows how the arrangements on the left are mapped to the arrangements on the right.

$$
\begin{array}{ll}
O_2BO_1K & \\
KO_2BO_1 & BOOK \\
O_1BO_2K & OBOK \\
KO_1BO_2 & KOBO \\
BO_1O_2K & \ldots \\
BO_2O_1K & \\
\ldots &
\end{array}
$$

(d) What kind of mapping is this, young grasshopper?

(e) In light of the Division Rule, how many arrangements are there of $BOOK$?

(f) Very good, young master! How many arrangements are there of the letters in $KE_1E_2PE_3R$?

(g) Suppose we map each arrangement of $KE_1E_2PE_3R$ to an arrangement of $KEEPER$ by erasing subscripts. List all the different arrangements of $KE_1E_2PE_3R$ that are mapped to $REPEEK$ in this way.

(h) What kind of mapping is this?

(i) So how many arrangements are there of the letters in $KEEPER$?

(j) *Now you are ready to face the BOOKKEEPER!*

How many arrangements of $BO_1O_2K_1K_2E_1E_2PE_3R$ are there?

(k) How many arrangements of $BOOK_1K_2E_1E_2PE_3R$ are there?

(l) How many arrangements of $BOOKKE_1E_2PE_3R$ are there?

(m) How many arrangements of $BOOKKEEPER$ are there?

Remember well what you have learned: subscripts on, subscripts off.
This is the Tao of Bookkeeper.

(n) How many arrangements of $VOODOODOLL$ are there?

(o) How many length 52 sequences of digits contain exactly 17 two's, 23 fives, and 12 nines?

Problems for Section 15.9

Class Problems

Problem 15.18.
Solve the following counting problems by defining an appropriate mapping (bijective or k-to-1) between a set whose size you know and the set in question.

(a) How many different ways are there to select a dozen donuts if four varieties are available?

(b) In how many ways can Mr. and Mrs. Grumperson distribute 13 identical pieces of coal to their two—no, three!—children for Christmas?

(c) How many solutions over the nonnegative integers are there to the inequality:

$$x_1 + x_2 + \ldots + x_{10} \leq 100$$

(d) We want to count step-by-step paths between points in the plane with integer coordinates. Ony two kinds of step are allowed: a right-step which increments the x coordinate, and an up-step which increments the y coordinate.

(i) How many paths are there from $(0, 0)$ to $(20, 30)$?

(ii) How many paths are there from $(0, 0)$ to $(20, 30)$ that go through the point $(10, 10)$?

(iii) How many paths are there from $(0, 0)$ to $(20, 30)$ that do *not* go through either of the points $(10, 10)$ and $(15, 20)$?

Hint: Let P be the set of paths from $(0, 0)$ to $(20, 30)$, N_1 be the paths in P that go through $(10, 10)$ and N_2 be the paths in P that go through $(15, 20)$.

Problem 15.19.

Solve the following counting problems. Define an appropriate mapping (bijective or k-to-1) between a set whose size you know and the set in question.

(a) An independent living group is hosting nine new candidates for membership. Each candidate must be assigned a task: 1 must wash pots, 2 must clean the kitchen, 3 must clean the bathrooms, 1 must clean the common area, and 2 must serve dinner. Write a multinomial coefficient for the number of ways this can be done.

(b) How many nonnegative integers less than 1,000,000 have exactly one digit equal to 9 and have a sum of digits equal to 17?

Exam Problems

Problem 15.20.

Here are the solutions to the next 10 problem parts, in no particular order.

$$n^m \qquad m^n \qquad \frac{n!}{(n-m)!} \qquad \binom{n+m}{m} \qquad \binom{n-1+m}{m} \qquad \binom{n-1+m}{n} \qquad 2^{mn}$$

(a) How many solutions over the natural numbers are there to the inequality $x_1 + x_2 + \cdots + x_n \leq m$?

(b) How many length m words can be formed from an n-letter alphabet, if no letter is used more than once?

(c) How many length m words can be formed from an n-letter alphabet, if letters can be reused?

(d) How many binary relations are there from set A to set B when $|A| = m$ and $|B| = n$?

(e) How many injections are there from set A to set B, where $|A| = m$ and $|B| = n \geq m$?

(f) How many ways are there to place a total of m distinguishable balls into n distinguishable urns, with some urns possibly empty or with several balls?

(g) How many ways are there to place a total of m indistinguishable balls into n distinguishable urns, with some urns possibly empty or with several balls?

(h) How many ways are there to put a total of m distinguishable balls into n distinguishable urns with at most one ball in each urn?

Problems for Section 15.10

Practice Problems

Problem 15.21.
The working days in the next year can be numbered 1, 2, 3, ..., 300. I'd like to avoid as many as possible.

- On even-numbered days, I'll say I'm sick.
- On days that are a multiple of 3, I'll say I was stuck in traffic.
- On days that are a multiple of 5, I'll refuse to come out from under the blankets.

In total, how many work days will I *avoid* in the coming year?

Class Problems

Problem 15.22.
A certain company wants to have security for their computer systems. So they have given everyone a name and password. A length 10 word containing each of the characters:

a, d, e, f, i, l, o, p, r, s,

is called a *cword*. A password will be a cword which does not contain any of the subwords "fails", "failed", or "drop".

For example, the following two words are passwords:

adefiloprs, srpolifeda,

but the following three cwords are not:

a**drop**eflis, **failedrop**s, **drop**e**fails**.

(a) How many cwords contain the subword "drop"?

(b) How many cwords contain both "drop" and "fails"?

(c) Use the Inclusion-Exclusion Principle to find a simple formula for the number of passwords.

Problem 15.23.
Let's develop a proof of the Inclusion-Exclusion formula using high school algebra.

(a) Most high school students will get freaked by the following formula, even though they actually know the rule it expresses. How would you explain it to them?

$$\prod_{i=1}^{n}(1-x_i) = \sum_{I\subseteq\{1,\ldots,n\}}(-1)^{|I|}\prod_{j\in I}x_j. \tag{15.11}$$

Hint: Show them an example.

For any set, S, let M_S be the *membership* function of S:

$$M_S(x) = \begin{cases} 1 & \text{if } x \in S, \\ 0 & \text{if } x \notin S. \end{cases}$$

Let $S_1, \ldots, S_n$ be a sequence of finite sets, and abbreviate M_{S_i} as M_i. Let the domain of discourse, D, be the union of the S_i's. That is, we let

$$D ::= \bigcup_{i=1}^{n} S_i,$$

and take complements with respect to D, that is,

$$\overline{T} ::= D - T,$$

for $T \subseteq D$.

(b) Verify that for $T \subseteq D$ and $I \subseteq \{1, \ldots n\}$,

$$M_{\overline{T}} = 1 - M_T, \tag{15.12}$$

$$M_{(\bigcap_{i\in I} S_i)} = \prod_{i\in I} M_{S_i}, \tag{15.13}$$

$$M_{(\bigcup_{i\in I} S_i)} = 1 - \prod_{i\in I}(1-M_i). \tag{15.14}$$

(Note that (15.13) holds when I is empty because, by convention, an empty product equals 1, and an empty intersection equals the domain of discourse, D.)

(c) Use (15.11) and (15.14) to prove

$$M_D = \sum_{\emptyset\neq I\subseteq\{1,\ldots,n\}}(-1)^{|I|+1}\prod_{j\in I}M_j. \tag{15.15}$$

(d) Prove that

$$|T| = \sum_{u\in D} M_T(u). \tag{15.16}$$

(e) Now use the previous parts to prove

$$|D| = \sum_{\emptyset \neq I \subseteq \{1,\ldots,n\}} (-1)^{|I|+1} \left|\bigcap_{i \in I} S_i\right| \tag{15.17}$$

(f) Finally, explain why (15.17) immediately implies the usual form of the Inclusion-Exclusion Principle:

$$|D| = \sum_{i=1}^{n} (-1)^{i+1} \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ |I|=i}} \left|\bigcap_{j \in I} S_j\right|. \tag{15.18}$$

Homework Problems

Problem 15.24.

How many paths are there from point $(0, 0)$ to $(50, 50)$ if every step increments one coordinate and leaves the other unchanged? How many are there when there are impassable boulders sitting at points $(10, 11)$ and $(21, 20)$? (You do not have to calculate the number explicitly; your answer may be an expression involving binomial coefficients.)

Hint: Count the number of paths going through $(10, 11)$, the number through $(21, 20)$, and use Inclusion-Exclusion.

Problem 15.25.

A *derangement* is a permutation $(x_1, x_2, \ldots, x_n)$ of the set $\{1, 2, \ldots, n\}$ such that $x_i \neq i$ for all i. For example, $(2, 3, 4, 5, 1)$ is a derangement, but $(2, 1, 3, 5, 4)$ is not because 3 appears in the third position. The objective of this problem is to count derangements.

It turns out to be easier to start by counting the permutations that are *not* derangements. Let S_i be the set of all permutations $(x_1, x_2, \ldots, x_n)$ that are not derangements because $x_i = i$. So the set of non-derangements is

$$\bigcup_{i=1}^{n} S_i.$$

(a) What is $|S_i|$?

(b) What is $|S_i \cap S_j|$ where $i \neq j$?

(c) What is $|S_{i_1} \cap S_{i_2} \cap \cdots \cap S_{i_k}|$ where $i_1, i_2, \ldots, i_k$ are all distinct?

(d) Use the inclusion-exclusion formula to express the number of non-derangements in terms of sizes of possible intersections of the sets $S_1, \ldots, S_n$.

(e) How many terms in the expression in part (d) have the form $|S_{i_1} \cap S_{i_2} \cap \cdots \cap S_{i_k}|$?

(f) Combine your answers to the preceding parts to prove the number of non-derangements is:

$$n!\left(\frac{1}{1!} - \frac{1}{2!} + \frac{1}{3!} - \cdots \pm \frac{1}{n!}\right).$$

Conclude that the number of derangements is

$$n!\left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots \pm \frac{1}{n!}\right).$$

(g) As n goes to infinity, the number of derangements approaches a constant fraction of all permutations. What is that constant? *Hint:*

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Problem 15.26.

How many of the numbers $2, \ldots, n$ are prime? The Inclusion-Exclusion Principle offers a useful way to calculate the answer when n is large. Actually, we will use Inclusion-Exclusion to count the number of *composite* (nonprime) integers from 2 to n. Subtracting this from $n - 1$ gives the number of primes.

Let C_n be the set of composites from 2 to n, and let A_m be the set of numbers in the range $m + 1, \ldots, n$ that are divisible by m. Notice that by definition, $A_m = \emptyset$ for $m \geq n$. So

$$C_n = \bigcup_{i=2}^{n-1} A_i. \tag{15.19}$$

(a) Verify that if $m \mid k$, then $A_m \supseteq A_k$.

(b) Explain why the right hand side of (15.19) equals

$$\bigcup_{\text{primes } p \leq \sqrt{n}} A_p. \tag{15.20}$$

(c) Explain why $|A_m| = \lfloor n/m \rfloor - 1$ for $m \geq 2$.

(d) Consider any two relatively prime numbers $p, q \leq n$. What is the one number in $(A_p \cap A_q) - A_{p \cdot q}$?

(e) Let $\mathcal{P}$ be a finite set of at least two primes. Give a simple formula for

$$|\bigcap_{p \in \mathcal{P}} A_p|.$$

(f) Use the Inclusion-Exclusion principle to obtain a formula for $|C_{150}|$ in terms the sizes of intersections among the sets $A_2, A_3, A_5, A_7, A_{11}$. (Omit the intersections that are empty; for example, any intersection of more than three of these sets must be empty.)

(g) Use this formula to find the number of primes up to 150.

Problems for Section 15.11

Class Problems

Problem 15.27.
According to the Multinomial theorem, $(w + x + y + z)^n$ can be expressed as a sum of terms of the form

$$\binom{n}{r_1, r_2, r_3, r_4} w^{r_1} x^{r_2} y^{r_3} z^{r_4}.$$

(a) How many terms are there in the sum?

(b) The sum of these multinomial coefficients has an easily expressed value. What is it?

$$\sum_{\substack{r_1+r_2+r_3+r_4=n, \\ r_i \in \mathbb{N}}} \binom{n}{r_1, r_2, r_3, r_4} = ? \tag{15.21}$$

Hint: How many terms are there when $(w + x + y + z)^n$ is expressed as a sum of monomials in w, x, y, z *before* terms with like powers of these variables are collected together under a single coefficient?

Problem 15.28. **(a)** Give a combinatorial proof of the following theorem:

$$n2^{n-1} = \sum_{k=1}^{n} k \binom{n}{k} \tag{15.22}$$

Hint: Let S be the set of all length-n sequences of 0's, 1's and a single *.

(b) Now prove (15.22) algebraically by applying the Binomial Theorem to $(1 + x)^n$ and taking derivatives.

Homework Problems

Problem 15.29.
Prove the following identity by algebraic manipulation and by giving a combinatorial argument:

$$\binom{n}{r}\binom{r}{k} = \binom{n}{k}\binom{n-k}{r-k}$$

Problem 15.30. **(a)** Find a combinatorial (*not* algebraic) proof that

$$\sum_{i=0}^{n} \binom{n}{i} = 2^n.$$

(b) Below is a combinatorial proof of an equation. What is the equation?

Proof. Stinky Peterson owns n newts, t toads, and s slugs. Conveniently, he lives in a dorm with $n + t + s$ other students. (The students are distinguishable, but creatures of the same variety are not distinguishable.) Stinky wants to put one creature in each neighbor's bed. Let W be the set of all ways in which this can be done.

On one hand, he could first determine who gets the slugs. Then, he could decide who among his remaining neighbors has earned a toad. Therefore, $|W|$ is equal to the expression on the left.

On the other hand, Stinky could first decide which people deserve newts and slugs and then, from among those, determine who truly merits a newt. This shows that $|W|$ is equal to the expression on the right.

Since both expressions are equal to $|W|$, they must be equal to each other. ■

(Combinatorial proofs are real proofs. They are not only rigorous, but also convey an intuitive understanding that a purely algebraic argument might not reveal. However, combinatorial proofs are usually less colorful than this one.)

Problem 15.31.
According to the Multinomial Theorem 15.7.2, $(x_1 + x_2 + \cdots + x_k)^n$ can be expressed as a sum of terms of the form

$$\binom{n}{r_1, r_2, \ldots, r_k} x_1^{r_1} x_2^{r_2} \ldots x_k^{r_k}.$$

(a) How many terms are there in the sum?

(b) The sum of these multinomial coefficients has an easily expressed value:

$$\sum_{\substack{r_1+r_2+\cdots+r_k=n,\\ r_i \in \mathbb{N}}} \binom{n}{r_1, r_2, \ldots, r_k} = k^n \tag{15.23}$$

Give a combinatorial proof of this identity.

Hint: How many terms are there when $(x_1 + x_2 + \cdots + x_k)^n$ is expressed as a sum of monomials in x_i *before* terms with like powers of these variables are collected together under a single coefficient?

Problems for Section 15.12

Class Problems

Problem 15.32.
Solve the following problems using the pigeonhole principle. For each problem, try to identify the *pigeons*, the *pigeonholes*, and a *rule* assigning each pigeon to a pigeonhole.

(a) In a certain Institute of Technolofy, Every ID number starts with a 9. Suppose that each of the 75 students in a class sums the nine digits of their ID number. Explain why two people must arrive at the same sum.

(b) In every set of 100 integers, there exist two whose difference is a multiple of 37.

(c) For any five points inside a unit square (not on the boundary), there are two points at distance *less than* $1/\sqrt{2}$.

(d) Show that if $n + 1$ numbers are selected from $\{1, 2, 3, \ldots, 2n\}$, two must be consecutive, that is, equal to k and $k + 1$ for some k.

Homework Problems

Problem 15.33.
Pigeon Huntin'

(a) Show that any odd integer x in the range $10^9 < x < 2 \cdot 10^9$ containing all ten digits $0, 1, \ldots, 9$ must have consecutive even digits. *Hint:* What can you conclude about the parities of the first and last digit?

(b) Show that there are 2 vertices of equal degree in any finite undirected graph with $n \geq 2$ vertices. *Hint:* Cases conditioned upon the existence of a degree zero vertex.

Problem 15.34.
Show that for any set of 201 positive integers less than 300, there must be two whose quotient is a power of three (with no remainder).

Problems for Section 15.13

Class Problems

Problem 15.35. (a) Show that the Magician could not pull off the trick with a deck larger than 124 cards.

Hint: Compare the number of 5-card hands in an n-card deck with the number of 4-card sequences.

(b) Show that, in principle, the Magician could pull off the Card Trick with a deck of 124 cards.

Hint: Hall's Theorem and degree-constrained (11.5.5) graphs.

Problem 15.36.
The Magician can determine the 5th card in a poker hand when his Assisant reveals the other 4 cards. Describe a similar method for determining 2 hidden cards in a hand of 9 cards when your Assisant reveals the other 7 cards.

Homework Problems

Problem 15.37.
Section 15.13.3 explained why it is not possible to perform a four-card variant of the hidden-card magic trick with one card hidden. But the Magician and her Assistant are determined to find a way to make a trick like this work. They decide to change

the rules slightly: instead of the Assistant lining up the three unhidden cards for the Magician to see, he will line up all four cards with one card face down and the other three visible. We'll call this the *face-down four-card trick*.

For example, suppose the audience members had selected the cards 9♡, 10♢, A♣, 5♣. Then the Assistant could choose to arrange the 4 cards in any order so long as one is face down and the others are visible. Two possibilities are:

A♣	?	10♢	5♣
?	5♣	9♡	10♢

(a) Explain why there must be a bipartite matching which will in theory allow the Magician and Assistant to perform the face-down four-card trick.

(b) There is actually a simple way to perform the face-down four-card trick.[6]

> **Case 1.** *there are two cards with the same suit*: Say there are two ♠ cards. The Assistant proceeds as in the original card trick: he puts one of the ♠ cards *face up as the first card*. He will place the second ♠ card *face down*. He then uses a permutation of the face down card and the remaining two face up cards to code the offset of the face down card from the first card.
>
> **Case 2.** *all four cards have different suits*: Assign numbers 0, 1, 2, 3 to the four suits in some agreed upon way. The Assistant computes, s, the sum modulo 4 of the ranks of the four cards, and chooses the card with suit s to be placed *face down as the first card*. He then uses a permutation of the remaining three face-up cards to code the rank of the face down card.

Explain how in Case 2. the Magician can determine the face down card from the cards the Assistant shows her.

(c) Explain how any method for performing the face-down four-card trick can be adapted to perform the regular (5-card hand, show 4 cards) with a 52-card deck consisting of the usual 52 cards along with a 53rd card call the *joker*.

[6] This elegant method was devised in Fall '09 by student Katie E Everett.