15 Generating Functions

~~Generating Functions are one of the most surprising and useful inventions in Discrete Mathematics. Roughly speaking, generating functions transform problems about *sequences* into problems about *functions*. This is great because we've got piles of mathematical machinery for manipulating functions. Thanks to generating functions, we can apply all that machinery to problems about sequences. In this way, we can use generating functions to solve all sorts of counting problems.~~

Several flavors of generating functions such as *ordinary*, *exponential*, and *Dirichlet* come up regularly in combinatorial mathematics. In addition, *Z-transforms*, which are closely related to ordinary generating functions, are important in control theory and signal processing. But ordinary generating functions are enough to illustrate the power of the idea, so we'll stick to them. So from now on *generating function* will mean the ordinary kind, and we will offer a taste of this large subject by showing how generating functions can be used to solve certain kinds of counting problems and how they can be used to find simple formulas for *linear-recursive* functions.

15.1 Infinite Series

Informally, a generating function, $F(x)$, is an infinite series

$$F(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3 + \cdots . \tag{15.1}$$

~~For example,~~ the infinite geometric series

$$G(x) ::= 1 + x + x^2 + \cdots + x^n + \cdots . \tag{15.2}$$

is a familiar generating function, and we can illustrate typical reasoning about generating functions by deriving a simple formula for $G(x)$. The approach is actually a simpler version of the perturbation method of Section 13.1.2. ~~Namely,~~

$$\begin{array}{rl} G(x) = & 1 + x + x^2 + x^3 + \cdots + x^n + \cdots \\ -xG(x) = & \quad\; - x - x^2 - x^3 - \cdots - x^n - \cdots \\ \hline G(x) - xG(x) = 1. & \end{array}$$

Solving for $G(x)$ gives

$$\sum_{n=0}^{\infty} x^n = G(x) = \frac{1}{1-x}. \tag{15.3}$$

Continuing with this approach yields a nice formula for

$$N(x) ::= 1 + 2x + 3x^2 + \cdots + (n+1)x^n + \cdots. \tag{15.4}$$

~~Namely~~

$$\begin{array}{rcccccccc}
N(x) = & 1 & +2x & +3x^2 & +4x^3 & +\cdots & + & (n+1)x^n & +\cdots \\
-xN(x) = & & -\ x & -2x^2 & -3x^3 & -\cdots & - & nx^n & -\cdots \\
\hline
N(x) - xN(x) = & 1 & +\ x & +\ x^2 & +\ x^3 & +\cdots & + & x^n & +\cdots \\
= G(x). & & & & & & & &
\end{array}$$

Solving for $N(x)$ gives

$$\sum_{n=0}^{\infty}(n+1)x^n = N(x) = \frac{G(x)}{1-x} = \frac{1}{(1-x)^2}. \tag{15.5}$$

We use the notation $[x^n]F(x)$ for the coefficient of x^n in the generating function $F(x)$. That is, $[x^n]F(x) ::= f_n$ for $F(x)$ given by equation (15.1). For example, we now have

$$[x^n]\left(\frac{1}{1-x}\right) = 1$$

$$[x^n]\left(\frac{1}{(1-x)^2}\right) = n+1.$$

15.1.1 Never Mind Convergence

The numerical values of $G(x)$ are undefined when $|x| \geq 1$ because the geometric series diverges. ~~So equation (15.3) holds numerically only when $|x| < 1$; likewise for equation (15.5).~~ But in the context of generating functions, we regard infinite series as formal algebraic objects and equations such as (15.3) and (15.5) as symbolic identities that hold for purely algebraic reasons. In fact, good use can be made of generating functions determined by infinite series that don't converge anywhere. We'll explain this further at the end of the chapter, but for now it's enough to know that we needn't worry about convergence.

15.2 Counting with Generating Functions

Generating functions are particularly useful for representing and counting the number of ways to select n things. ~~For example,~~ if there are two flavors of donuts

—chocolate and vanilla —let d_n be the number of ways to select n chocolate or vanilla flavored donuts. So $d_n = n + 1$ because there are $n + 1$ such donut selections, namely, all chocolate, 1 vanilla and $n - 1$ chocolate, 2 vanilla and $n - 2$ chocolate,..., all vanilla. We define a generating function, $D(x)$, for counting these donut selections by letting the coefficient of x^n be d_n. So by equation (15.5)

$$D(x) = \frac{1}{(1-x)^2}. \tag{15.6}$$

More generally, suppose we have two kinds of things —say apples and bananas —and some constraints on how many of each may be selected. Say there are a_n ways to select n apples and b_n ways to select n bananas. So the generating function for counting apples would be

$$A(x) ::= \sum_{n=0}^{\infty} a_n x^n,$$

and for bananas would be

$$B(x) ::= \sum_{n=0}^{\infty} b_n x^n.$$

Now suppose apples come in baskets of 6, so there is no way to select 1 to 5 apples, one way to select 6 apples, no way to select 7, etc. In other words,

$$a_n = \begin{cases} 1 & \text{if } n \text{ is a multiple of 6,} \\ 0 & \text{otherwise.} \end{cases}$$

In this case we would have

$$\begin{aligned} A(x) &= 1 + x^6 + x^{12} + \cdots + x^{6n} + \cdots \\ &= 1 + x^6 + \left(x^6\right)^2 + \cdots + \left(x^6\right)^n + \cdots \\ &= \frac{1}{1-x^6}. \end{aligned}$$

Let's also suppose there are two kinds of bananas —red and yellow. Now $b_n = n + 1$ by the same reasoning used to count selections of n chocolate and vanilla donuts, so we would have

$$B(x) = \frac{1}{(1-x)^2}.$$

So how many ways are there to select a mix of n apples and bananas? We could select one apple in a_1 ways and then $n - 1$ bananas in b_{n-1} ways, for a total of

$a_1 b_{n-1}$ ways to select n apples and bananas using only one apple. More generally, we could select k apples in a_k ways and then $n-k$ bananas in b_{n-k} ways, for a total of $a_k b_{n-k}$ ways to select select n apples and bananas including exactly k apples. So the total number of ways to select a mix of n apples and bananas is

$$a_0 b_n + a_1 b_{n-1} + a_2 b_{n-1} + \cdots + a_n b_0. \qquad (15.7)$$

Now here's the cool connection between counting and generating functions: expression (15.7) is equal to the coefficient of x^n in the product $A(x)B(x)$.

15.2.1 Products of Generating Functions

In other words, we're claiming that

Rule (Product)**.**

$$[x^n](A(x) \cdot B(x)) = a_0 b_n + a_1 b_{n-1} + a_2 b_{n-1} + \cdots + a_n b_0. \qquad (15.8)$$

To explain the generating function Product Rule, we can think about evaluating the product $A(x) \cdot B(x)$ by using a table to identify all the cross-terms from the product of the sums:

	$b_0 x^0$	$b_1 x^1$	$b_2 x^2$	$b_3 x^3$	$\ldots$
$a_0 x^0$	$a_0 b_0 x^0$	$a_0 b_1 x^1$	$a_0 b_2 x^2$	$a_0 b_3 x^3$	$\ldots$
$a_1 x^1$	$a_1 b_0 x^1$	$a_1 b_1 x^2$	$a_1 b_2 x^3$	$\ldots$	
$a_2 x^2$	$a_2 b_0 x^2$	$a_2 b_1 x^3$	$\ldots$		
$a_3 x^3$	$a_3 b_0 x^3$	$\ldots$			
$\vdots$	$\ldots$				

In this layout, all the terms involving the same power of x lie on a 45-degree sloped diagonal. So the index-n diagonal contains all the x^n-terms, and the coefficient of x^n in the product $A(x) \cdot B(x)$ is the sum of all the coefficients of the terms on this diagonal, namely, (15.7). The sequence of coefficients of the product $A(x) \cdot B(x))$ is called the *convolution* of the sequences $(a_0, a_1, a_2, \ldots)$ and $(b_0, b_1, b_2, \ldots)$. In addition to their algebraic role, convolutions of sequences play a prominent role in signal processing and control theory.

This Product Rule provides the algebraic justification for the fact that a geometric series equals $1/(1-x)$ regardless of convergence. Namely, according to the Product Rule, the product of the geometric series and the series

$$1 + (-1)x + 0x^2 + \cdots + 0x^n + \cdots$$

for $1 - x$ is the series

$$1 + 0x + 0x^2 + \cdots + 0x^n + \cdots$$

for the constant 1. So with multiplication defined by the Product Rule, the geometric series is the multiplicative inverse, $1/(1-x)$, of $1 - x$.

Similar reasoning justifies multiplying a generating function by a constant term by term. That is, a special case of the Product Rule is the

Rule (Constant Factor)**.** *For any constant, c, and generating function, $F(x)$,*

$$[x^n](c \cdot F(x)) = c \cdot [x^n]F(x). \tag{15.9}$$

15.2.2 The Convolution Rule

We can summarize the discussion above with the

Rule (Convolution)**.** *Let $A(x)$ be the generating function for selecting items from a set $\mathcal{A}$, and let $B(x)$ be the generating function for selecting items from a set $\mathcal{B}$ disjoint from $\mathcal{A}$. The generating function for selecting items from the union $\mathcal{A} \cup \mathcal{B}$ is the product $A(x) \cdot B(x)$.*

The Rule depends on a precise definition of what "selecting items from the union $\mathcal{A} \cup \mathcal{B}$" means. Informally, the idea is that the restrictions on the selection of items from sets $\mathcal{A}$ and $\mathcal{B}$ carry over to selecting items from $\mathcal{A} \cup \mathcal{B}$. Formally, the Convolution Rule applies when there is a bijection between n-element selections from $\mathcal{A} \cup \mathcal{B}$ and ordered pairs of selections from the sets $\mathcal{A}$ and $\mathcal{B}$ containing a total of n elements. We think the informal statement is clear enough.

15.2.3 Counting Donuts with the Convolution Rule

We can use the Convolution Rule to derive in another way the generating function $D(x)$ for the number of ways to select chocolate and vanilla donuts given in (15.6). Namely, there is only one way to select exactly n chocolate donuts. That means every coefficient of the generating function for selecting n chocolate donuts equals one. So the generating function for chocolate donut selections is $1/(1-x)$; likewise for the generating function for selecting only vanilla donuts. Now by the Convolution Rule, the generating function for the number of ways to select n donuts when both chocolate and vanilla flavors are available is

$$D(x) = \frac{1}{1-x} \cdot \frac{1}{1-x} = \frac{1}{(1-x)^2}.$$

So we have derived (15.6) without appeal to (15.5).

The first general counting problem we considered was the number of ways to select a n doughnuts when k flavors were available. Our application of the Convolution Rule for two flavors carries right over to this general case, and we conclude that the generating function for selections of donuts when k flavors are available is $1/(1-x)^k$. So we have

$$[x^n]\left(\frac{1}{(1-x)^k}\right) = \binom{n+(k-1)}{n} \tag{15.10}$$

by Corollary 14.5.3.

Extracting Coefficients from Maclauren's Theorem

We've used a donut-counting argument to derive the coefficients of $1/(1-x)^k$, but it's instructive to derive this coefficient algebraically, which we can do using Maclauren's Theorem:

Theorem 15.2.1 (Maclauren's Theorem)**.**

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \cdots + \frac{f^{(n)}(0)}{n!}x^n + \cdots.$$

This theorem says that the nth coefficient of $1/(1-x)^k$ is equal to its nth derivative evaluated at 0 and divided by $n!$. Computing the nth derivative turns out not to be very difficult

$$\frac{d^n}{d^n x}\frac{1}{(1-x)^k} = k(k+1)\cdots(k+n-1)(1-x)^{-(k+n)}$$

(see Problem 15.3), so

$$\begin{aligned}
[x^n]\left(\frac{1}{(1-x)^k}\right) &= \left(\frac{d^n}{d^n x}\frac{1}{(1-x)^k}\right)(0)\frac{1}{n!} \\
&= \frac{k(k+1)\cdots(k+n-1)(1-0)^{-(k+n)}}{n!} \\
&= \binom{n+(k-1)}{n}.
\end{aligned}$$

So instead of using the donut-counting formula (15.10) to find the coefficients of x^n, we could have used this algebraic argument and the Convolution Rule to derive the donut-counting formula.

15.2.4 The Binomial Theorem from the Convolution Rule

The Convolution Rule also provides a new perspective on the Binomial Theorem 14.6.4. Here is how. First, consider a single-element set $\{a_1\}$. The generating function for the number of ways to select n elements from this set is simply $1 + x$: we have 1 way to select zero elements, 1 way to select the one element, and 0 ways to select more than one element. Similarly, the number of ways to select n elements from any single-element set $\{a_i\}$ has the same generating function $1 + x$. Now by the Convolution Rule, the generating function for choosing a subset of n elements from the set $\{a_1, a_2, \ldots, a_m\}$ is the product $(1+x)^m$ of the generating function for selecting from each of the m one-element sets. Since we know that the number of ways to select n elements from a set of size m is $\binom{m}{n}$, we conclude that that

$$[x^n](1 + x)^m = \binom{m}{n},$$

which is a restatement of the Binomial Theorem 14.6.4.

So we have proved the Binomial Theorem without having to analyze the expansion of the expression $(1 + x)^m$ into a sum of products.

15.2.5 An "Impossible" Counting Problem

~~So far everything we've done with generating functions we could have done another way. But here is an absurd counting problem — really over the top! In how many ways can we fill a bag with n fruits subject to the following constraints?~~

- The number of apples must be even.
- The number of bananas must be a multiple of 5.
- There can be at most four oranges.
- There can be at most one pear.

For example, there are 7 ways to form a bag with 6 fruits:

Apples	6	4	4	2	2	0	0
Bananas	0	0	0	0	0	5	5
Oranges	0	2	1	4	3	1	0
Pears	0	0	1	0	1	0	1

~~These constraints are so~~ complicated that getting a nice answer may seem impossible. But let's see what generating functions reveal.

~~Let's first~~ construct a generating function for choosing apples. We can choose a set of 0 apples in one way, a set of 1 apple in zero ways (since the number of apples must be even), a set of 2 apples in one way, a set of 3 apples in zero ways, and so forth. So we have:

$$A(x) = 1 + x^2 + x^4 + x^6 + \cdots = \frac{1}{1-x^2}$$

Similarly, the generating function for choosing bananas is:

$$B(x) = 1 + x^5 + x^{10} + x^{15} + \cdots = \frac{1}{1-x^5}$$

Now, we can choose a set of 0 oranges in one way, a set of 1 orange in one way, and so on. However, we cannot choose more than four oranges, so we have the generating function:

$$O(x) = 1 + x + x^2 + x^3 + x^4 = \frac{1-x^5}{1-x}$$

Here we're using the formula (13.2) for a finite geometric sum. Finally, we can choose only zero or one pear, so we have:

$$P(x) = 1 + x$$

The Convolution Rule says that the generating function for choosing from among all four kinds of fruit is:

$$\begin{aligned} A(x)B(x)O(x)P(x) &= \frac{1}{1-x^2}\frac{1}{1-x^5}\frac{1-x^5}{1-x}(1+x) \\ &= \frac{1}{(1-x)^2} \\ &= 1 + 2x + 3x^2 + 4x^3 + \cdots \end{aligned}$$

Almost everything cancels! We're left with $1/(1-x)^2$, which we found a power series for earlier: the coefficient of x^n is simply $n+1$. Thus, the number of ways to form a bag of n fruits is just $n+1$. This is consistent with the example we worked out, since there were 7 different fruit bags containing 6 fruits. *Amazing!*

15.3 Partial Fractions

We got a simple solution to the "impossible" counting problem of Section 15.2.5 because its generating function simplified to the expression $1/(1-x)^2$ whose power series coefficients we already knew. ~~Of course the problem was contrived so this would work out.~~ To solve more general problems using generating functions, we need ways to find power series coefficients for generating functions given as formulas. Maclauren's Theorem 15.2.1 is a very general method for finding coefficients, but it only applies when formulas for repeated derivatives can be found, which isn't often. However, there is an automatic way to find the power series coefficients for any formula that is a quotient of polynomials, ~~namely, by using the method of~~ partial fractions from elementary calculus.

The partial fraction method is based on the fact that quotients of polynomials can be expressed as sums of terms whose power series coefficients have nice formulas. For example when the denominator polynomial has distint nonzero roots, the method rests on

Lemma 15.3.1. *Let $p(x)$ be a polynomial of degree less than n and let $\alpha_1, \ldots, \alpha_n$ be distinct, nonzero numbers. Then there are constants $c_1, \ldots, c_n$ such that*

$$\frac{p(x)}{(1-\alpha_1 x)(1-\alpha_2 x)\cdots(1-\alpha_n x)} = \frac{c_1}{1-\alpha_1 x} + \frac{c_2}{1-\alpha_2 x} + \cdots + \frac{c_n}{1-\alpha_n x}.$$

Let's illustrate the use of Lemma 15.3.1 by finding the power series coefficients for the function

$$R(x) ::= \frac{x}{1-x-x^2}.$$

We can use the quadratic formula to find the roots r_1, r_2 of the denominator, $1 - x - x^2$, ~~namely~~

$$r_1 = \frac{-1-\sqrt{5}}{2}, r_2 = \frac{-1+\sqrt{5}}{2}.$$

So

$$1 - x - x^2 = (x - r_1)(x - r_2) = r_1 r_2 (1 - x/r_1)(1 - x/r_2).$$

With a little algebra, we find that

$$R(x) = \frac{x}{(1-\alpha_1 x)(1-\alpha_2 x)}$$

where

$$\alpha_1 = \frac{1+\sqrt{5}}{2}$$
$$\alpha_2 = \frac{1-\sqrt{5}}{2}.$$

Next we find c_1 and c_2 which satisfy:

$$\frac{x}{(1-\alpha_1 x)(1-\alpha_2 x)} = \frac{c_1}{1-\alpha_1 x} + \frac{c_2}{1-\alpha_2 x} \tag{15.11}$$

In general, we can do this by plugging in a couple of values for x to generate two linear equations in c_1 and c_2 and then solve the equations for c_1 and c_2. A simpler approach in this case comes from multiplying both sides of (15.11) by the left hand denominator to get

$$x = c_1(1-\alpha_2 x) + c_2(1-\alpha_1 x).$$

Now letting $x = 1/\alpha_2$ we obtain

$$c_2 = \frac{1/\alpha_2}{1-\alpha_1/\alpha_2} = \frac{1}{\alpha_2-\alpha_1} = -\frac{1}{\sqrt{5}},$$

and similarly, letting $x = 1/\alpha_1$ we obtain

$$c_1 = \frac{1}{\sqrt{5}}.$$

Plugging these values for c_1, c_2 into equation (15.11) finally gives the partial fraction expansion

$$R(x) = \frac{x}{1-x-x^2} = \frac{1}{\sqrt{5}}\left(\frac{1}{1-\alpha_1 x} - \frac{1}{1-\alpha_2 x}\right)$$

Each term in the partial fractions expansion has a simple power series given by the geometric sum formula:

$$\frac{1}{1-\alpha_1 x} = 1 + \alpha_1 x + \alpha_1^2 x^2 + \cdots$$
$$\frac{1}{1-\alpha_2 x} = 1 + \alpha_2 x + \alpha_2^2 x^2 + \cdots$$

Substituting in these series gives a power series for the generating function:

$$R(x) = \frac{1}{\sqrt{5}}\left((1+\alpha_1 x + \alpha_1^2 x^2 + \cdots) - (1 + \alpha_2 x + \alpha_2^2 x^2 + \cdots)\right),$$

so

$$
\begin{aligned}
[x^n]R(x) &= \frac{\alpha_1^n - \alpha_2^n}{\sqrt{5}} \\
&= \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right) \qquad (15.12)
\end{aligned}
$$

15.3.1 Partial Fractions with Repeated Roots

Lemma 15.3.1 generalizes to the case when the denominator polynomial has a repeated nonzero root with multiplicity m by expanding the quotient into a sum a terms of the form

$$\frac{c}{(1-\alpha x)^k}$$

where α is the reciprocal of the root and $k \leq m$. A formula for the coefficients of such a term follows from the donut formula (15.10). ~~Namely,~~

$$[x^n]\left(\frac{c}{(1-\alpha x)^k}\right) = c\alpha^n \binom{n-(k-1)}{n}. \qquad (15.13)$$

When $\alpha = 1$, this follows from the donut formula (15.10) and termwise multiplication by the constant c. The case for arbitrary α follows by substituting αx for x in the power series; this changes x^n into $(\alpha x)^n$ and so has the effect of multiplying the coefficient of x^n by α^n.[1]

15.4 Solving Linear Recurrences

15.4.1 A Generating Function for the Fibonacci Numbers

The Fibonacci numbers $f_0, f_1, \ldots, f_n, \ldots$ are defined recursively as follows:

$$
\begin{aligned}
f_0 &::= 0 \\
f_1 &::= 1 \\
f_n &= ::= f_{n-1} + f_{n-2} \qquad (\text{for } n \geq 2).
\end{aligned}
$$

Generating functions will now allow us to derive an astonishing closed formula for f_n.

[1] In other words,

$$[x^n]F(\alpha x) = \alpha^n \cdot [x^n]F(x).$$

~~Namely, let~~ $F(x)$ be the generating function for the sequence of Fibonacci numbers, that is,

$$F(x) ::= f_0 + f_1 x + f_2 x^2 + \cdots f_n x^n + \cdots .$$

Reasoning as we did at the start of this chapter to derive the formula for a geometric series, we have

$$\begin{array}{rclclclclclc}
F(x) & = & f_0 & + & f_1 x & + & f_2 x^2 & + & \cdots & + & f_n x^n + \cdots . \\
-xF(x) & = & & - & f_0 x & - & f_1 x^2 & - & \cdots & - & f_{n-1} x^n + \cdots . \\
-x^2 F(x) & = & & & & - & f_0 x^2 & - & \cdots & - & f_{n-2} x^n + \cdots . \\
\hline
F(x)(1 - x - x^2) & = & f_0 & + & (f_1 - f_0)x & + & 0x^2 & + & \cdots & + & 0x^n + \cdots . \\
& = & 0 & + & 1x & + & 0x^2 & = & x, & &
\end{array}$$

so

$$F(x) = \frac{x}{1 - x - x^2}.$$

But ~~wait,~~ $F(x)$ is the same as the function we used to illustrate the partial fraction method for finding coefficients in Section 15.3. So by equation (15.12), we find that

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

As a formula for Fibonacci numbers, this is ~~astonishing and maybe scary~~. ~~From the formula, it's not even obvious that its value is an integer.~~ But the formula is very useful. For example, it provides (via the repeated squaring method) a much more efficient way to compute Fibonacci numbers than crunching through the recurrence. It also clearly reveals the exponential growth of these numbers.

15.4.2 The Towers of Hanoi

According to legend, there is a temple in Hanoi with three posts and 64 gold disks of different sizes. Each disk has a hole through the center so that it fits on a post. In the misty past, all the disks were on the first post, with the largest on the bottom and the smallest on top, as shown in Figure 15.1.

Monks in the temple have labored through the years since to move all the disks to one of the other two posts according to the following rules:

- The only permitted action is removing the top disk from one post and dropping it onto another post.

- A larger disk can never lie above a smaller disk on any post.

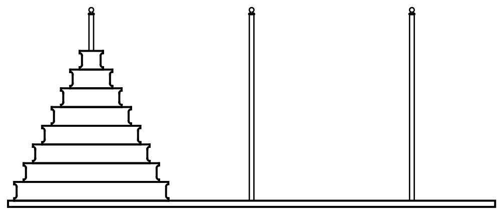

Figure 15.1 The initial configuration of the disks in the Towers of Hanoi problem.

So, for example, picking up the whole stack of disks at once and dropping them on another post is illegal. That's good, because the legend says that when the monks complete the puzzle, the world will end!

To clarify the problem, suppose there were only 3 gold disks instead of 64. Then the puzzle could be solved in 7 steps as shown in Figure 15.2.

The questions we must answer are, "Given sufficient time, can the monks succeed?" If so, "How long until the world ends?" And, most importantly, "Will this happen before the final exam?"

A Recursive Solution

The Towers of Hanoi problem can be solved recursively. As we describe the procedure, we'll also analyze the minimum number, t_n, of steps required to solve the n-disk problem. For example, some experimentation shows that $t_1 = 1$ and $t_2 = 3$. The procedure illustrated above shows that t_3 is at most 7, though there might be a solution with fewer steps.

The recursive solution has three stages, which are described below and illustrated in Figure 15.3. For clarity, the largest disk is shaded in the figures.

Stage 1. Move the top $n-1$ disks from the first post to the second using the solution for $n-1$ disks. This can be done in t_{n-1} steps.

Stage 2. Move the largest disk from the first post to the third post. This takes just 1 step.

Stage 3. Move the $n-1$ disks from the second post to the third post, again using the solution for $n-1$ disks. This can also be done in t_{n-1} steps.

This algorithm shows that t_n, the minimum number of steps required to move n disks to a different post, is at most $t_{n-1} + 1 + t_{n-1} = 2t_{n-1} + 1$. We can use this fact to upper bound the number of operations required to move towers of various

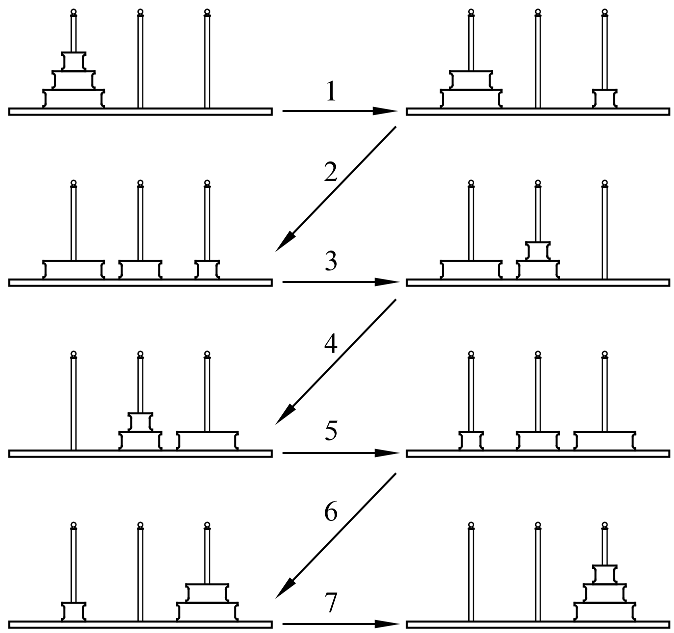

Figure 15.2 The 7-step solution to the Towers of Hanoi problem when there are $n = 3$ disks.

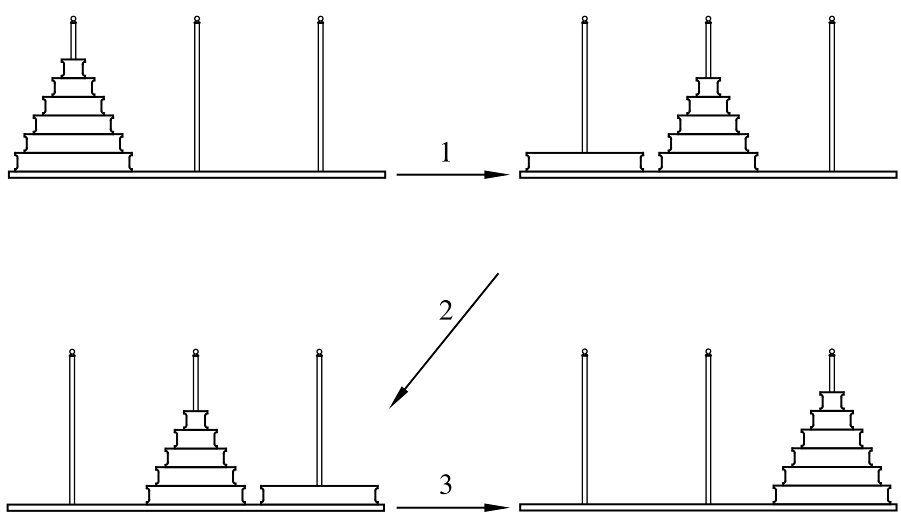

Figure 15.3 A recursive solution to the Towers of Hanoi problem.

heights:

$$t_3 \leq 2 \cdot t_2 + 1 = 7$$
$$t_4 \leq 2 \cdot t_3 + 1 \leq 15$$

Continuing in this way, we could eventually compute an upper bound on t_{64}, the number of steps required to move 64 disks. So this algorithm answers our first question: given sufficient time, the monks can finish their task and end the world. ~~This is a shame. After all that effort, they'd probably want to smack a few high fives and go out for burgers and ice cream, but nope — world's over.~~

~~Finding a Recurrence~~

~~We cannot yet compute the exact number of steps that the monks need to move the 64 disks, only an upper bound. Perhaps, having pondered the problem since the beginning of time, the monks have devised a better algorithm.~~

~~In fact,~~ there is no better ~~algorithm, and here is why. At~~ some step, the monks must move the largest disk from the first post to a different post. For this to happen, the $n-1$ smaller disks must all be stacked out of the way on the only remaining post. Arranging the $n-1$ smaller disks this way requires at least t_{n-1} moves. After the largest disk is moved, at least another t_{n-1} moves are required to pile the $n-1$ smaller disks on top.

This argument shows that the number of steps required is at least $2t_{n-1} + 1$. Since we gave an algorithm using exactly that number of steps, we can now write an expression for t_n, the number of moves required to complete the Towers of Hanoi problem with n disks:

$$\begin{aligned} t_0 &= 0 \\ t_n &= 2t_{n-1} + 1 \qquad \text{(for } n \geq 1\text{)}. \end{aligned}$$

Solving the Recurrence

We can now find a formula for t_n using generating functions. ~~Namely,~~ let $T(x)$ be the generating function for the t_n's, that is,

$$T(x) ::= t_0 + t_1 x + t_2 x^2 + \cdots t_n x^n + \cdots.$$

Reasoning as we did for the Fibonacci recurrence, we have

$$\begin{array}{rcrcrcccr} T(x) & = & t_0 & + & t_1 x & + & \cdots & + & t_n x^n + \cdots \\ -2xT(x) & = & & - & 2t_0 x & - & \cdots & - & 2t_{n-1} x^n + \cdots \\ -1/(1-x) & = & -1 & - & 1x & - & \cdots & - & 1x^n + \cdots \\ \hline T(x)(1-2x) - 1/(1-x) & = & t_0 - 1 & + & 0x & + & \cdots & + & 0x^n + \cdots \\ & = & -1, & & & & & & \end{array}$$

so

$$T(x)(1-2x) = \frac{1}{1-x} - 1 = \frac{x}{1-x},$$

and

$$T(x) = \frac{x}{(1-2x)(1-x)}.$$

Using partial fractions,

$$\frac{x}{(1-2x)(1-x)} = \frac{c_1}{1-2x} + \frac{c_2}{1-x}$$

for some constants c_1, c_2. Now multiplying both sides by the left hand denominator gives

$$x = c_1(1-x) + c_2(1-2x).$$

Substituting $1/2$ for x yields $c_1 = 1$ and substituting 1 for x yields $c_2 = -1$, which gives

$$T(x) = \frac{1}{1-2x} - \frac{1}{1-x}.$$

Finally we can read off the simple formula for the numbers of steps needed to move a stack of n disks:

$$t_n = [x^n]T(x) = [x^n]\left(\frac{1}{1-2x}\right) - [x^n]\left(\frac{1}{1-x}\right) = 2^n - 1.$$

15.4.3 Solving General Linear Recurrences

An equation of the form

$$f(n) = c_1 f(n-1) + c_2 f(n-2) + \cdots + c_d f(n-d) + h(n) \tag{15.14}$$

for constants $c_i \in \mathbb{C}$ is called a *degree d linear recurrence* with inhomogeneous term $h(n)$.

The methods above extend straightforwardly to solving linear recurrences with a large class of inhomogeneous terms. In particular, when the inhomogeneous term itself has a generating function that can be expressed as a quotient of polynomials, the approach used above to derive generating functions for the Fibonacci and Tower of Hanoi examples carries over to yield a quotient of polynomials that defines the generating function $f(0) + f(1)x + f(2)x^2 + \cdots$. Then partial fractions can be used to find a formula for $f(n)$ that is a linear combination of terms of the form $n^k \alpha^n$ where k is a nonnegative integer $\leq d$ and α is the reciprocal of a root of the denominator polynomial. For example, see Problems 15.11, 15.15, 15.14 and 15.12.

15.5 Formal Power Series

TBA - to appear

Problems for Section 15.3

Practice Problems

Problem 15.1.
You would like to buy a bouquet of flowers. You find an online service that will make bouquets of **lilies**, **roses** and **tulips**, subject to the following constraints:

- there must be at most 1 lily,
- there must be an odd number of tulips,
- there must be at least two roses.

Example: A bouquet of no lilies, 3 tulips, and 5 roses satisfies the constraints.

Express $B(x)$, the generating function for the number of ways to select a bouquet of n flowers, as a quotient of polynomials (or products of polynomials). You do not need to simplify this expression.

Problem 15.2.
Write a formula for the generating function whose successive coefficients are given by the sequence:

(a) 0, 0, 1, 1, 1,...

(b) 1, 1, 0, 0, 0,...

(c) 1, 0, 1, 0, 1, 0, 1,...

(d) 1, 4, 6, 4, 1, 0, 0, 0,...

(e) 1, 1, 1/2, 1/6, 1/24, 1/120,...

(f) 1, 2, 3, 4, 5,...

(g) 1, 4, 9, 16, 25,...

Class Problems

Problem 15.3.
Let $A(x) = \sum_{n=0}^{\infty} a_n x^n$. Then it's easy to check that

$$a_n = \frac{A^{(n)}(0)}{n!},$$

where $A^{(n)}$ is the nth derivative of A. Use this fact (which you may assume) instead of the Convolution Counting Principle, to prove that

$$\frac{1}{(1-x)^k} = \sum_{n=0}^{\infty} \binom{n+k-1}{k-1} x^n.$$

So if we didn't already know the Bookkeeper Rule, we could have proved it from this calculation and the Convolution Rule for generating functions.

Problem 15.4.
We are interested in generating functions for the number of different ways to compose a bag of n donuts subject to various restrictions. For each of the restrictions in (a)-(e) below, find a closed form for the corresponding generating function.

(a) All the donuts are chocolate and there are at least 3.

(b) All the donuts are glazed and there are at most 2.

(c) All the donuts are coconut and there are exactly 2 or there are none.

(d) All the donuts are plain and their number is a multiple of 4.

(e) The donuts must be chocolate, glazed, coconut, or plain with the numbers of each flavor subject to the constraints above.

(f) Find a closed form for the number of ways to select n donuts subject to the constraints of the previous part.

Problem 15.5. **(a)** Let

$$S(x) ::= \frac{x^2 + x}{(1-x)^3}.$$

What is the coefficient of x^n in the generating function series for $S(x)$?

(b) Explain why $S(x)/(1-x)$ is the generating function for the sums of squares. That is, the coefficient of x^n in the series for $S(x)/(1-x)$ is $\sum_{k=1}^{n} k^2$.

(c) Use the previous parts to prove that

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

Homework Problems

Problem 15.6.
We will use generating functions to determine how many ways there are to use pennies, nickels, dimes, quarters, and half-dollars to give n cents change.

(a) Write the sequence P_n for the number of ways to use only pennies to change n cents. Write the generating function for that sequence.

(b) Write the sequence N_n for the number of ways to use only nickels to change n cents. Write the generating function for that sequence.

(c) Write the generating function for the number of ways to use only nickels and pennies to change n cents.

(d) Write the generating function for the number of ways to use pennies, nickels, dimes, quarters, and half-dollars to give n cents change.

(e) Explain how to use this function to find out how many ways are there to change 50 cents; you do *not* have to provide the answer or actually carry out the process.

Problem 15.7.
Taking derivatives of generating functions is another useful operation. This is done termwise, that is, if

$$F(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3 + \cdots,$$

then

$$F'(x) ::= f_1 + 2f_2 x + 3f_3 x^2 + \cdots.$$

For example,

$$\frac{1}{(1-x)^2} = \left(\frac{1}{(1-x)}\right)' = 1 + 2x + 3x^2 + \cdots$$

so

$$H(x) ::= \frac{x}{(1-x)^2} = 0 + 1x + 2x^2 + 3x^3 + \cdots$$

is the generating function for the sequence of nonnegative integers. Therefore

$$\frac{1+x}{(1-x)^3} = H'(x) = 1 + 2^2x + 3^2x^2 + 4^2x^3 + \cdots,$$

so

$$\frac{x^2+x}{(1-x)^3} = xH'(x) = 0 + 1x + 2^2x^2 + 3^2x^3 + \cdots + n^2x^n + \cdots$$

is the generating function for the nonnegative integer squares.

(a) Prove that for all $k \in \mathbb{N}$, the generating function for the nonnegative integer kth powers is a quotient of polynomials in x. That is, for all $k \in \mathbb{N}$ there are polynomials $R_k(x)$ and $S_k(x)$ such that

$$[x^n]\left(\frac{R_k(x)}{S_k(x)}\right) = n^k. \tag{15.15}$$

Hint: Observe that the derivative of a quotient of polynomials is also a quotient of polynomials. It is not necessary work out explicit formulas for R_k and S_k to prove this part.

(b) Conclude that if $f(n)$ is a function on the nonnegative integers defined recursively in the form

$$f(n) = af(n-1) + bf(n-2) + cf(n-3) + p(n)\alpha^n$$

where the $a, b, c, \alpha \in \mathbb{C}$ and p is a polynomial with complex coefficients, then the generating function for the sequence $f(0), f(1), f(2), \ldots$ will be a quotient of polynomials in x, and hence there is a closed form expression for $f(n)$.

Hint: Consider

$$\frac{R_k(\alpha x)}{S_k(\alpha x)}$$

Problem 15.8.

Miss McGillicuddy never goes outside without a collection of pets. In particular:

- She brings a positive number of songbirds, which always come in pairs.
- She may or may not bring her alligator, Freddy.

- She brings at least 2 cats.
- She brings two or more chihuahuas and labradors leashed together in a line.

Let P_n denote the number of different collections of n pets that can accompany her, where we regard chihuahuas and labradors leashed up in different orders as different collections, even if there are the same number chihuahuas and labradors leashed in the line.

For example, $P_6 = 4$ since there are 4 possible collections of 6 pets:

- 2 songbirds, 2 cats, 2 chihuahuas leashed in line
- 2 songbirds, 2 cats, 2 labradors leashed in line
- 2 songbirds, 2 cats, a labrador leashed behind a chihuahua
- 2 songbirds, 2 cats, a chihuahua leashed behind a labrador

And $P_7 = 16$ since there are 16 possible collections of 7 pets:

- 2 songbirds, 3 cats, 2 chihuahuas leashed in line
- 2 songbirds, 3 cats, 2 labradors leashed in line
- 2 songbirds, 3 cats, a labrador leashed behind a chihuahua
- 2 songbirds, 3 cats, a chihuahua leashed behind a labrador
- 4 collections consisting of 2 songbirds, 2 cats, 1 alligator, and a line of 2 dogs
- 8 collections consisting of 2 songbirds, 2 cats, and a line of 3 dogs.

(a) Let

$$P(x) ::= P_0 + P_1 x + P_2 x^2 + P_3 x^3 + \cdots$$

be the generating function for the number of Miss McGillicuddy's pet collections. Verify that

$$P(x) = \frac{4x^6}{(1-x)^2(1-2x)}.$$

(b) Find a simple formula for P_n.

Exam Problems

Problem 15.9.
T-Pain is planning an epic boat trip and he needs to decide what to bring with him.

- He must bring some burgers, but they only come in packs of 6.
- He and his two friends can't decide whether they want to dress formally or casually. He'll either bring 0 pairs of flip flops or 3 pairs.
- He doesn't have very much room in his suitcase for towels, so he can bring at most 2.
- In order for the boat trip to be truly epic, he has to bring at least 1 nautical-themed pashmina afghan.

(a) Let $B(x)$ be the generating function for the number of ways to bring n burgers, $F(x)$ for the number of ways to bring n pairs of flip flops, $T(x)$ for towels, and $A(x)$ for Afghans. Write simple formulas for each of these.

$$B(x) = \qquad\qquad F(x) =$$
$$T(x) = \qquad\qquad A(x) =$$

(b) Let g_n be the the number of different ways for T-Pain to bring n items (burgers, pairs of flip flops, towels, and/or afghans) on his boat trip. Let $G(x)$ be the generating function $\sum_{n=0}^{\infty} g_n x^n$. Verify that

$$G(x) = \frac{x^7}{(1-x)^2}.$$

(c) Find a simple formula for g_n.

Problems for Section 15.4

Practice Problems

Problem 15.10.
Let $b, c, a_0, a_1, a_2, \ldots$ be real numbers such that

$$a_n = b(a_{n-1}) + c$$

for $n \geq 1$.

Let $G(x)$ be the generating function for this sequence.

(a) Express the coefficient of x^n for $n \geq 1$ in the series expansion of $bxG(x)$ in terms of b and a_i for suitable i.

(b) The coefficient of x^n for $n \geq 1$ in the series expansion of $cx/(1-x)$ is

(c) Therefore, $G(x) - bxG(x) - cx/(1-x) =$

(d) Using the method of partial fractions, we can find real numbers d and e such that

$$G(x) = d/L(x) + e/M(x).$$

What are $L(x)$ and $M(x)$?

Class Problems

Problem 15.11.
The famous mathematician, Fibonacci, has decided to start a rabbit farm to fill up his time while he's not making new sequences to torment future college students. Fibonacci starts his farm on month zero (being a mathematician), and at the start of month one he receives his first pair of rabbits. Each pair of rabbits takes a month to mature, and after that breeds to produce one new pair of rabbits each month. Fibonacci decides that in order never to run out of rabbits or money, every time a batch of new rabbits is born, he'll sell a number of newborn pairs equal to the total number of pairs he had three months earlier. Fibonacci is convinced that this way he'll never run out of stock.

(a) Define the number, r_n, of pairs of rabbits Fibonacci has in month n, using a recurrence relation. That is, define r_n in terms of various r_i where $i < n$.

(b) Let $R(x)$ be the generating function for rabbit pairs,

$$R(x) ::= r_0 + r_1 x + r_2 x^2 + \cdots$$

Express $R(x)$ as a quotient of polynomials.

(c) Find a partial fraction decomposition of the generating function $R(x)$.

(d) Finally, use the partial fraction decomposition to come up with a closed form expression for the number of pairs of rabbits Fibonacci has on his farm on month n.

Problem 15.12.
Less well-known than the Towers of Hanoi —but no less fascinating —are the

Towers of Sheboygan. As in Hanoi, the puzzle in Sheboygan involves 3 posts and n rings of different sizes. The rings are placed on post #1 in order of size with the smallest ring on top and largest on bottom.

The objective is to transfer all n rings to post #2 via a sequence of moves. As in the Hanoi version, a move consists of removing the top ring from one post and dropping it onto another post with the restriction that a larger ring can never lie above a smaller ring. But unlike Hanoi, a local ordinance requires that **a ring can only be moved from post #1 to post #2, from post #2 to post #3, or from post #3 to post #1.** Thus, for example, moving a ring directly from post #1 to post #3 is not permitted.

(a) One procedure that solves the Sheboygan puzzle is defined recursively: to move an initial stack of n rings to the next post, move the top stack of $n-1$ rings to the furthest post by moving it to the next post two times, then move the big, nth ring to the next post, and finally move the top stack another two times to land on top of the big ring. Let s_n be the number of moves that this procedure uses. Write a simple linear recurrence for s_n.

(b) Let $S(x)$ be the generating function for the sequence $\langle s_0, s_1, s_2, \ldots \rangle$. Carefully show that

$$S(x) = \frac{x}{(1-x)(1-4x)}.$$

(c) Give a simple formula for s_n.

(d) A better (indeed optimal, but we won't prove this) procedure to solve the Towers of Sheboygan puzzle can be defined in terms of two mutually recursive procedures, procedure $P_1(n)$ for moving a stack of n rings 1 pole forward, and $P_2(n)$ for moving a stack of n rings 2 poles forward. This is trivial for $n = 0$. For $n > 0$, define:

$P_1(n)$: Apply $P_2(n-1)$ to move the top $n-1$ rings two poles forward to the third pole. Then move the remaining big ring once to land on the second pole. Then apply $P_2(n-1)$ again to move the stack of $n-1$ rings two poles forward from the third pole to land on top of the big ring.

$P_2(n)$: Apply $P_2(n-1)$ to move the top $n-1$ rings two poles forward to land on the third pole. Then move the remaining big ring to the second pole. Then apply $P_1(n-1)$ to move the stack of $n-1$ rings one pole forward to land on the first pole. Now move the big ring 1 pole forward again to land on the third pole. Finally, apply $P_2(n-1)$ again to move the stack of $n-1$ rings two poles forward to land on the big ring.

Let t_n be the number of moves needed to solve the Sheboygan puzzle using procedure $P_1(n)$. Show that

$$t_n = 2t_{n-1} + 2t_{n-2} + 3, \tag{15.16}$$

for $n > 1$.

Hint: Let u_n be the number of moves used by procedure $P_2(n)$. Express each of t_n and u_n as linear combinations of t_{n-1} and u_{n-1} and solve for t_n.

(e) Derive values a, b, c, α, β such that

$$t_n = a\alpha^n + b\beta^n + c.$$

Conclude that $t_n = o(s_n)$.

Homework Problems

Problem 15.13.
Generating functions provide an interesting way to count the number of strings of matched brackets. To do this, we'll use a description of these strings as the set, GoodCount, of strings of brackets with a good count.

Namely, one precise way to determine if a string is matched is to start with 0 and read the string from left to right, adding 1 to the count for each left bracket and subtracting 1 from the count for each right bracket. For example, here are the counts for the two strings above

$$\begin{array}{ccccccccccccc} & [&] &] & [& [& [& [& [&] &] &] &] \\ 0 & 1 & 0 & -1 & 0 & 1 & 2 & 3 & 4 & 3 & 2 & 1 & 0 \end{array}$$

$$\begin{array}{ccccccccccc} & [& [& [&] &] & [&] &] & [&] \\ 0 & 1 & 2 & 3 & 2 & 1 & 2 & 1 & 0 & 1 & 0 \end{array}$$

A string has a *good count* if its running count never goes negative and ends with 0. So the second string above has a good count, but the first one does not because its count went negative at the third step.

Definition. Let

$$\text{GoodCount} ::= \{s \in \{], [\}^* \mid s \text{ has a good count}\}.$$

The matched strings can now be characterized precisely as this set of strings with good counts.

Let c_n be the number of strings in GoodCount with exactly n left brackets, and let $C(x)$ be the generating function for these numbers:

$$C(x) ::= c_0 + c_1 x + c_2 x^2 + \cdots .$$

(a) The *wrap* of a string, s, is the string, $\textbf{[}\, s \,\textbf{]}$, that starts with a left bracket followed by the characters of s, and then ends with a right bracket. Explain why the generating function for the wraps of strings with a good count is $xC(x)$.

Hint: The wrap of a string with good count also has a good count that starts and ends with 0 and remains *positive* everywhere else.

(b) Explain why, for every string, s, with a good count, there is a unique sequence of strings $s_1, \ldots, s_k$ that are wraps of strings with good counts and $s = s_1 \cdots s_k$. For example, the string $r ::= \textbf{[[]][][[][]]} \in$ GoodCount equals $s_1 s_2 s_3$ where $s_1 ::= \textbf{[[]]}$, $s_2 ::= \textbf{[]}$, $s_3 ::= \textbf{[[][]]}$, and this is the only way to express r as a sequence of wraps of strings with good counts.

(c) Conclude that

$$C = 1 + xC + (xC)^2 + \cdots + (xC)^n + \cdots , \tag{15.17}$$

so

$$C = \frac{1}{1 - xC} , \tag{15.18}$$

and hence

$$C = \frac{1 \pm \sqrt{1 - 4x}}{2x} . \tag{15.19}$$

Let $D(x) ::= 2xC(x)$. Expressing D as a power series

$$D(x) = d_0 + d_1 x + d_2 x^2 + \cdots ,$$

we have

$$c_n = \frac{d_{n+1}}{2}. \tag{15.20}$$

(d) Use (15.19), (15.20), and the value of c_0 to conclude that

$$D(x) = 1 - \sqrt{1 - 4x}.$$

(e) Prove that

$$d_n = \frac{(2n-3) \cdot (2n-5) \cdots 5 \cdot 3 \cdot 1 \cdot 2^n}{n!}.$$

Hint: $d_n = D^{(n)}(0)/n!$

(f) Conclude that

$$c_n = \frac{1}{n+1}\binom{2n}{n}.$$

Exam Problems

Problem 15.14.
Define the sequence $r_0, r_1, r_2, \ldots$ recursively by the rule that $r_0 ::= 1$ and

$$r_n ::= 7r_{n-1} + (n+1) \qquad \text{for } n > 0.$$

Let $R(x) ::= \sum_0^\infty r_n x^n$ be the generating function of this sequence. Express $R(x)$ as a quotient of polynomials or products of polynomials. You do *not* have to find a closed form for r_n.

Problem 15.15.
Alyssa Hacker sends out a video that spreads like wildfire over the UToob network. On the day of the release —call it *day zero* —and the day following —call it *day one* —the video doesn't receive any hits. However, starting with day two, the number of hits, r_n, can be expressed as seven times the number of hits on the previous day, four times the number of hits the day before that, and the number of days that has passed since the release of the video plus one. So, for example on day 2, there will be $7 \times 0 + 4 \times 0 + 3 = 3$ hits.

(a) Give a linear a recurrence for r_n.

(b) Express the generating function $R(x) ::= \sum_0^\infty r_n x^n$ as a quotient of polynomials or products of polynomials. You do *not* have to find a closed form for r_n.

Index