

Generating Functions I - How to Count with Power Series

Nic Ford

1 Introduction

In this article, which will be the first in a short series, we're going to explore something called *generating functions*, a method for solving counting problems by manipulating polynomials and power series. One of the amazing things about this technique is that it involves a lot of ideas from calculus, like derivatives and Taylor series, that were invented to study continuous processes and would seem to have no business showing up in a context like counting, where everything is discrete and all the answers to your questions are integers.

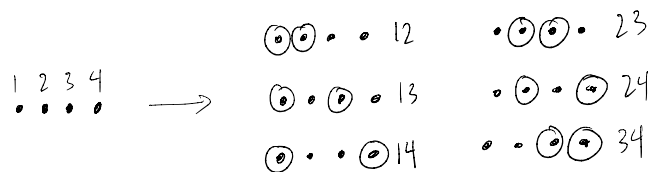
Because of this, while I'm not assuming any prior exposure to generating functions, you will need some familiarity with calculus, especially derivatives and Taylor series, to follow along. (It might also be helpful if you've seen Pascal's triangle in some context before, but we'll quickly recap most of what we'll need to know about it when it comes up.) We'll start by introducing the technique with a simple counting problem, and then go through three more examples that can be solved in a similar way.

Generating functions are applicable to many more problems than these, and in future articles in this series I hope to explore a couple that get more intricate. For now, though, we need to learn what it is we're doing before making things too complicated. Let's get started!

2 Our First Generating Function

2.1 Binomial Coefficients and Pascal's Triangle

We'll start our investigation of counting problems with a somewhat famous counting problem. Suppose you're given a set of n objects, and, for some $k \leq n$, you want to pick k of the objects and set them aside. (You can't pick any of the objects more than once, and order doesn't matter.) How many ways are there to do this? For specific values of n and k , you can answer this question just by listing all the possibilities and counting. For example, if $n = 4$ and $k = 2$, there are 6:

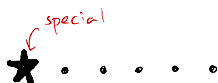


The notation mathematicians use for this number is $\binom{n}{k}$, pronounced “ n choose k .” (You might have also seen the notation ${}_nC_k$.) So the figure above shows that $\binom{4}{2} = 6$. I encourage you to verify that the other $\binom{4}{k}$ ’s are:

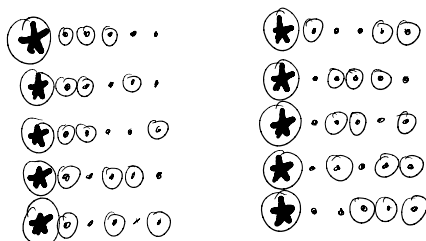
$$\binom{4}{0} = 1; \quad \binom{4}{1} = 4; \quad \binom{4}{2} = 6; \quad \binom{4}{3} = 4; \quad \binom{4}{4} = 1.$$

These $\binom{n}{k}$ numbers are called **binomial coefficients**. You might have recognized this list of numbers — that is, 1, 4, 6, 4, 1 — as the fourth row of Pascal’s triangle, and it is always true that the $\binom{n}{k}$ ’s form the n ’th row of Pascal’s triangle if you use the convention that the top row is row 0. (In fact, I like to take this as the *definition* of Pascal’s triangle, but you don’t have to if you have another definition you prefer.)

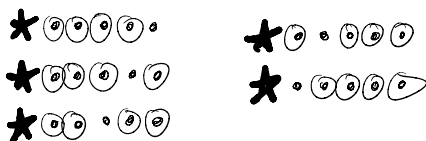
There are an enormous number of interesting patterns in Pascal’s triangle. One of the most famous is that each number in it is the sum of two numbers right above it, which in our notation amounts to the claim that $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$. There’s a nice way to see this purely in terms of our choosing- k -things-from- n -things interpretation. Suppose I know the values of all the $\binom{5}{k}$ ’s and I’m interested in computing $\binom{6}{4}$. Pick one of the 6 objects in your set and call it the “special” object.



Once you’ve done this, the ways of picking 4 objects can be divided into two classes: the ones that include the special object, and the ones that don’t. If your group *does* include the special object, then you need to pick 3 of the remaining 5 non-special objects to get up to 4, and there are $\binom{5}{3} = 10$ ways to do this.



If your group *doesn’t* include the special object, then you’ve chosen 4 of the remaining 5 non-special objects, and there are $\binom{5}{4} = 5$ ways to do this.



Putting this together, we therefore get that $\binom{6}{4} = \binom{5}{3} + \binom{5}{4} = 10 + 5 = 15$, finishing the proof that $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ when $n = 6$ and $k = 4$. I encourage you to convince yourself that the same method will work for any choice of n and k .

2.2 Another Pattern in Pascal's Triangle

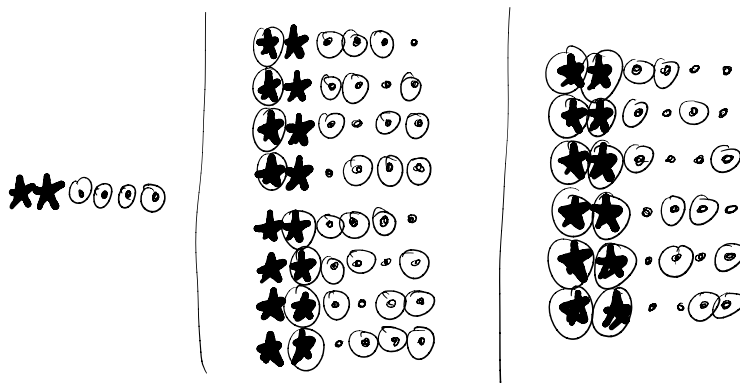
The main topic of this section will be a generalization of this fact and the method we used to prove it. So if it isn't yet clear *why* the process we just went through can be turned into an argument that $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ for any n and k , I encourage you to take a moment to think through it some more, possibly working it out for another choice of n and k , or finding another explanation somewhere. (There's a nice one on the Wikipedia page for "Pascal's rule".)

We were able to compute $\binom{6}{4}$ by splitting our set of 6 objects into a set of 1 and a set of 5. What if we had instead split them up some other way, say into 2 and 4? Our experience with the last example suggests that this should probably result in a different relationship between binomial coefficients, some way to write $\binom{6}{4}$ in terms of $\binom{2}{k}$'s and $\binom{4}{k}$'s. Let's see what happens when we apply the logic that worked before to this new case.

To emphasize the parallels between our new method and our old one, let's call the objects in our set of 2 "special" and mark them the same way as before:



Now, instead of dividing our ways of picking 4 objects into two classes like we did before, the natural thing to do is to divide them into *three* classes: the ones that uses 0 special objects, the ones that use 1 special object, and the ones that use 2 special objects:



How many entries are there in each class? The first and last ones are the simplest. If I don't pick any special objects, then I need to pick 4 non-special objects, and there's just $\binom{4}{4} = 1$ way to do this. If I pick 2 special objects, I need 2 non-special objects, so there are $\binom{4}{2} = 6$ ways to do this.

This might lead you to think that the middle class has $\binom{4}{3} = 4$ members, but a look at the picture above (and the fact that we already know the final answer is supposed to be 15 rather than $1 + 6 + 4 = 11$) will tell you that this isn't enough. What this misses is that there are also $\binom{2}{1} = 2$ ways to pick the 1 special object to go along with the 3 non-specials. So in fact, the middle class has $\binom{2}{1} \binom{4}{3} = 2 \cdot 4 = 8$ members.

Pulling this all together gives us our desired formula:

$$\binom{6}{4} = \binom{2}{0} \binom{4}{4} + \binom{2}{1} \binom{4}{3} + \binom{2}{2} \binom{4}{2} = 1 \cdot 1 + 2 \cdot 4 + 1 \cdot 6 = 15.$$

The final answer (which after all we already knew) is much less interesting than the pattern: notice that every term is a product of two binomial coefficients, one with a 2 on top and one with a 4, and that the bottom numbers always add up to 4, reflecting the fact that each term corresponds to a different way of allocating our 4 objects between the special and non-special subsets.

(There are two similar terms we could have written down reflecting the other two ways of writing 4 as a sum of two nonnegative integers, namely $\binom{2}{3}\binom{4}{1}$ and $\binom{2}{4}\binom{4}{0}$. These would correspond to picking 3 or 4 special objects, which of course you can't do because there are only 2 of them. For this reason, the usual mathematical convention is that $\binom{n}{k} = 0$ whenever $k > n$, and with that convention in place you're free to either include these extra terms or not, since they'll both just be zero.)

Following this recipe for a general n and k gives us a new formula relating the binomial coefficients on row n of Pascal's triangle to the ones on rows p and q whenever $p + q = n$: we have

$$\binom{n}{k} = \binom{p}{0}\binom{q}{k} + \binom{p}{1}\binom{q}{k-1} + \cdots + \binom{p}{k}\binom{q}{0}.$$

The sum will have $k + 1$ terms in it, one for each way of writing k as a sum of two nonnegative integers. Some of these terms might have binomial coefficients with their bottom number bigger than their top number, in which case the term will be zero and you're free to drop it if you want. I encourage you to check that if you take $p = 1$ and $q = n - 1$ and drop all the terms that end up equalling zero, you recover the $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ rule from our earlier discussion!

2.3 Convolution and Multiplying Polynomials

This operation, where we express the entries on the n 'th row of Pascal's triangle in terms of a sum of products of entries from the p 'th and q 'th rows, is important enough to have a name. Suppose I've got two lists of numbers, call them (a_0, a_1, \dots, a_m) and (b_0, b_1, \dots, b_n) . The **convolution** of these two lists is the list $(c_0, c_1, \dots, c_{m+n})$ whose k 'th entry is given by

$$c_k = a_0 b_k + a_1 b_{k-1} + \cdots + a_k b_0,$$

where any terms that go off the end of our original lists are taken to be 0. (The act of computing a convolution is called "*convolving*," like how performing multiplication is called "multiplying.")

For example, the convolution of the list $(3, 2, 1)$ with the list $(4, 5)$ is $(12, 23, 14, 5)$. Here's how to work out the 14 in some detail, and I encourage you to do something similar for the other three entries to make sure the operation is clear:

$$\begin{array}{l} \begin{array}{ccc} a_0 & a_1 & a_2 \\ \parallel & \parallel & \parallel \\ (3, & 2, & 1) \end{array} * \begin{array}{cc} b_0 & b_1 \\ \parallel & \parallel \\ (4, & 5) \end{array} = (c_0, c_1, c_2, c_3) \\ c_2 = \cancel{a_0 b_2} + a_1 b_1 + a_2 b_0 \\ \quad \quad \quad \curvearrowright \text{There is no } b_2, \text{ so this is zero} \\ = 2 \cdot 5 + 1 \cdot 4 = 14 \end{array}$$

(The $*$ on the top line is a common symbol for convolution. The easiest thing to forget when doing these computations is that our lists start with the 0'th entry! This convention is

necessary to line things up with the story we just told about binomial coefficients, and for the other applications we're about to explore.) Using this new terminology, we can express the result of our whole preceding conversation in a single sentence: if $p + q = n$, then the n 'th row of Pascal's triangle is the convolution of the p 'th and q 'th rows.

There is a very nice way to think about convolution in terms of multiplying polynomials, which is probably easiest to understand through an example. For each of our two lists from earlier, write down the polynomial whose x^k coefficient is the k 'th entry on the list, like this:

$$3 + 2x + 1x^2 \quad 4 + 5x.$$

Now, notice that when you multiply these two polynomials, the coefficients of the result are the same as the convolution we computed before:

$$(3 + 2x + 1x^2) \cdot (4 + 5x) = 12 + 23x + 14x^2 + 5x^3.$$

This is not a coincidence! In order to find the x^k coefficient of the product, we need to look at all pairs of terms from the two polynomials we're multiplying where the exponents on x add up to k ; I encourage you to convince yourself that the resulting computation is exactly the same as the one we described earlier for finding the k 'th entry in the convolution of the two original lists.

This observation — that when you multiply two polynomials, the resulting sequence of coefficients is the convolution of the original two sequences of coefficients — leads us, finally, to our main definition. Given a list of numbers (a_0, a_1, \dots, a_n) , the polynomial $a_0 + a_1x + \dots + a_nx^n$ is called the **generating function** of the original list.

As we've seen in our example with the rows of Pascal's triangle, convolution can be an important tool for thinking about counting problems, and the concept of generating functions gives us a nice way to do the "bookkeeping" associated with convolution in terms of an operation that's more familiar. (Among other things, identifying convolution with polynomial multiplication immediately tells us that convolution is commutative and associative and distributes over addition.) Let's see how this looks in our example.

Let's write $f_n(x)$ for the generating function of the n 'th row of Pascal's triangle. For the rows we considered in our example above, the associated f_n 's are

$$\begin{aligned} f_2(x) &= 1 + 2x + x^2 \\ f_4(x) &= 1 + 4x + 6x^2 + 4x^3 + x^4 \\ f_6(x) &= 1 + 6x + 15x^2 + 20x^3 + 15x^4 + 6x^5 + x^6. \end{aligned}$$

The rule we found relating the rows of the triangle to each other can be summarized very succinctly in terms of the f_n 's: we showed that $f_{p+q}(x) = f_p(x)f_q(x)$. (If you'd like, you can check that $f_6(x) = f_2(x)f_4(x)$ directly, and observe that the computations you perform to find each coefficient are exactly the same as the ones we did before.)

Something kind of satisfying happens if you apply this rule to some f_n to split it up as much as possible. We can use it to write $f_n(x) = f_{n-1}(x)f_1(x)$, and then apply it again to the f_{n-1} to get $f_n(x) = f_{n-2}(x)f_1(x)^2$, and so on until we get $f_n(x) = f_1(x)^n$. Since (as I encourage you to check) we can directly compute $f_1(x) = 1 + x$, we arrive at the following fact:

$$(1 + x)^n = f_n(x) = \binom{n}{0} + \binom{n}{1}x + \binom{n}{2}x^2 + \dots + \binom{n}{n}x^n.$$

This is called the **binomial theorem**, and you might or might not have run across it before. (A “binomial” is a polynomial with two terms, like the $1 + x$ that appears in this statement. This is also the source of the name “binomial coefficient” for $\binom{n}{k}$.) It’s often taught in high school math classes in a slightly different form involving $(x + y)^n$; if you’ve seen that version before it’s a nice exercise to try to prove it from the one we just proved here.

From here we can even directly extract a formula for $\binom{n}{k}$. In general, you can get the x^k coefficient of any polynomial $p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ by taking the derivative k times and then plugging in zero: if you do this, you get $p^{(k)}(0) = k!a_k$. In our present case, we want to take the k ’th derivative of $(1 + x)^n$, which you can do using the regular chain rule and power rule from calculus. I encourage you to check that

$$f_n^{(k)}(x) = n \cdot (n - 1) \cdot (n - 2) \cdots (n - k + 1)(1 + x)^{n-k} = \frac{n!}{(n - k)!}(1 + x)^{n-k},$$

and so, putting all this together, we learn that

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

It’s possible to prove the binomial theorem directly, without having to know anything about convolution and generating functions. (I won’t spoil all the details here in case you’d like to think about it, but the key step is to imagine expanding out the product of n copies of $1 + x$ and figuring out why the number of copies of x^k you get is equal to the number of ways of choosing k objects from a set of n .) But now that we’ve seen the concept of generating functions, we can apply it to a variety of other problems where things won’t be so simple. We’ll go through a few now.

3 Generating Functions as Power Series

3.1 Multiset Binomial Coefficients

Our first example will be a slight variation of the one we just did. Consider a set of n objects just like before, but this time we’ll pick k of them *with repeats allowed*. Order still doesn’t matter, so for example if $n = 4$ and $k = 2$ there are 10 possibilities: 11, 12, 13, 14, 22, 23, 24, 33, 34, and 44.

The number of ways to do this has the slightly unwieldy name **multiset binomial coefficient**. It’s written $\binom{n}{k}$, pronounced “ n multichoose k .” Our goal is to use generating functions to find a formula for these numbers like the one we just found for the $\binom{n}{k}$ ’s.

Right away there’s a big difference between this example and the previous one: now that we’re allowing repeats, $\binom{n}{k}$ can be nonzero even if $k > n$. (As a quick example, check that $\binom{2}{6} = 7$.) This means that if we pick an n and want to form a generating function m_n for the sequence $(\binom{n}{0}, \binom{n}{1}, \dots)$ like we did for the ordinary binomial coefficients, it will need to have infinitely many terms! For example, when $n = 4$ the first few terms would look like

$$m_4(x) = 1 + 4x + 10x^2 + 20x^3 + 35x^4 + 56x^5 + \cdots,$$

with terms continuing on forever.

Is this a problem? Strictly speaking, this means that the generating function for this sequence isn’t a *polynomial*, since by definition polynomials have finitely many terms. Instead it’s what’s

called a **power series**, that is, an expression of the form $a_0 + a_1x + a_2x^2 + \dots$, where all the a_k 's are potentially nonzero. There are good reasons to be nervous about manipulating these objects as if they were polynomials! We'll circle back to these questions later, but it will be easier to think about them once we've seen how the method works, so for now let's press on and see what we can learn about the $\binom{n}{k}$'s by taking this perspective.

The first big observation is that, if we split up our set of n into a set of p and a set of q like we did before, then the convolution relationship we found between the $\binom{n}{k}$'s also holds for the $\binom{n}{k}$'s, and for exactly the same reason: you can divide up all the ways of picking k objects according to how many are chosen from the first subset and how many are chosen from the second. The fact that repeats are allowed doesn't affect this argument in the slightest! In other words, we once again have that

$$\binom{n}{k} = \binom{p}{0} \binom{q}{k} + \binom{p}{1} \binom{q}{k-1} + \dots + \binom{p}{k} \binom{q}{0}.$$

We used the corresponding fact about $\binom{n}{k}$'s to prove that $f_n(x) = f_p(x)f_q(x)$ whenever $p + q = n$, and therefore $f_n(x) = f_1(x)^n$. Does this work for the m_n 's? Can you even multiply power series like that?

In fact you can! The key thing to notice is that, if you try to multiply two power series, say

$$(a_0 + a_1x + a_2x^2 + \dots)(b_0 + b_1x + b_2x^2 + \dots),$$

then even though there are infinitely many terms, the coefficient of each x^k in the result only depends on finitely many of them. This is simply because the only terms from the two original power series that have any chance of giving me an x^k in the product are the ones with powers of x less than or equal to k . More specifically, the coefficient on x^k in the product will be

$$a_0b_k + a_1b_{k-1} + \dots + a_kb_0,$$

just like for polynomials.

In other words, the relationship we found between convolution and multiplication of polynomials also holds for power series. And so, since the $\binom{n}{k}$'s have the same convolution relationship to each other that the $\binom{n}{k}$'s did, their generating functions m_n also have the same relationship to each other: we get that $m_{p+q}(x) = m_p(x)m_q(x)$, and therefore that $m_n(x) = m_1(x)^n$.

Even though we also knew $f_n(x) = f_1(x)^n$, this doesn't mean m_n just equals f_n , because m_1 is not equal to f_1 . But it's not that hard to compute m_1 directly: it's the generating function for the $\binom{1}{k}$'s, and these numbers are all 1 regardless of k . (No matter what k is, there's only one way to pick k object with repeats from a set of size 1: you have to just pick that one object k times.) In other words,

$$m_1(x) = 1 + x + x^2 + x^3 + \dots$$

You might recognize this series from calculus. It's called the **geometric series**, and when $|x| < 1$ it converges to $1/(1-x)$. (If this doesn't ring a bell, that's fine! We'll come back to the identification of this series with $1/(1-x)$ in just a bit.) If we use this, we get a nice formula for our generating function:

$$m_n(x) = (1-x)^{-n}.$$

Just as we did for the binomial coefficients, we can get our promised formula for $\binom{n}{k}$ by differentiating this k times. It's good practice to work this out yourself, so I won't list all the steps here, but the final answer ends up being

$$\binom{n}{k} = \frac{(n+k-1)!}{k!(n-1)!} = \binom{n+k-1}{k}.$$

You can even check to see that this gives the right answer for a few small choices of n and k if you like!

3.2 Formal Power Series

Now, as promised, let's talk a bit about the mathematical meaning of the operations we just performed. As we said earlier, you'd be right to be suspicious about manipulating these infinite power series exactly as though they were finite polynomials. And there are indeed plenty of ways in which infinite series don't behave as "nicely" as finite sums. So if we're going to argue (as in fact we are) that there are no such issues here, it would be nice to have some justification.

One idea might be to think of our power series as **Taylor series** for particular functions, like you probably did when you studied calculus. It is possible to prove, for example, that if you've got Taylor series for two functions $f(x)$ and $g(x)$, then $f(x)g(x)$ also has a Taylor series, and its x^k term can be found using our convolution formula, just like you'd expect by naively manipulating power series like they were polynomials. One can also show that a function can only have one Taylor series, which we need to know whenever we want to argue that two sequences of numbers must be equal term by term just because their generating functions are the same function.

While you could set things up this way, it is in a certain sense missing the point. We don't really care about our generating functions *as functions*, that is to say, we're very seldom interested in actually plugging numbers in for x . (Notice that this never happened once in either of the previous two examples!) For our purposes, they're essentially just a bookkeeping device for keeping track of the sequences of coefficients, and especially for doing convolutions.

So instead of worrying about convergence, we can think of our power series that way: as just a slightly funny notation for the sequence of coefficients (a_0, a_1, a_2, \dots) . The corresponding expression $a_0 + a_1x + a_2x^2 + \dots$ is called a **formal power series**, "formal" because we are just thinking of this as a string of symbols and not as a function you could potentially plug some x into. (For this reason, the name "generating function" is kind of unfortunate! It might be better to call them "generating power series" or something, but the name is pretty firmly established so we're going to use it.)

If we want to think of our generating functions as formal power series instead of functions, then the operations we used to manipulate them will have to be built from scratch, since we can't fall back on thinking of them as functions. For example, since formal power series are just strings of symbols, we'll need to come up with a rule for how to add and multiply them. The obvious thing to do is to say that, given two formal power series $f(x) = a_0 + a_1x + \dots$ and $g(x) = b_0 + b_1x + \dots$, the sum $f(x) + g(x)$ is the power series whose k 'th term is $(a_k + b_k)x^k$, and the product $f(x)g(x)$ is the power series whose terms are given by our convolution formula. (From here, a very careful treatment would also require checking that addition and multiplication satisfy all the algebraic relationships you expect, like commutativity, associativity, and distributivity. Feel free to try your hand at any of these if you're interested!)

What about the fact that $1 + x + x^2 + \dots = 1/(1 - x)$? Didn't that require thinking about Taylor series and convergence and so on? You certainly *can* think of it that way, but that equation also has a perfectly reasonable interpretation in terms of formal power series: it says that the power series $1 + x + x^2 + \dots$ is the multiplicative inverse of $1 - x$. In other words, it says that if you multiply $1 + x + x^2 + \dots$ by $1 - x$ using the convolution rule, then you get 1, that is, you get that the x^0 coefficient is 1 and all the rest of the coefficients are 0. This is in fact true, and I strongly encourage you to try to prove it!

Derivatives can be handled in a similar formal way: rather than worry about limits and difference quotients, we can simply *define* the derivative of $a_0 + a_1x + a_2x^2 + a_3x^3 + \dots$ to be the

power series $a_1 + 2a_2x + 3a_3x^2 + \dots$. Just as for addition and multiplication, if you were worried about proving everything you'd want to check here that this derivative operation satisfies the product rule and chain rule and so on that you expect from calculus.

The formal power series perspective on generating functions is in many ways “cleaner” than thinking of them as actual functions. It absolves you of the need to worry about convergence — which after all is irrelevant to us, since we don't care about plugging values into our “functions” — and in particular it enables you to work with power series that might not converge *anywhere* except at $x = 0$. But it comes at the cost of requiring you, at least if you care about proving everything, to rebuild all the basic operations like addition, multiplication, and differentiation from scratch.

Whether this tradeoff feels “worth it” to you is mostly a matter of taste: while the convention in mathematics is to think of generating functions as formal power series, if you strongly prefer to think of them as Taylor series of actual functions, you won't really lose out on much. (And occasionally in can actually be useful to flip between the two perspectives.) Regardless, I hope to convince you through the next couple of examples that they're a very powerful tool for analyzing sequences of numbers, and that remains true no matter which way you think of them! We'll turn to those examples now.

4 Derangements

A **permutation of k** is a way of rearranging the numbers $1, 2, \dots, k$ so that each number appears exactly once. For example, there are 6 permutations of 3: 123, 132, 213, 231, 312, and 321.

Counting permutations is pretty straightforward: you have k choices for the first number, then $k - 1$ choices for the second number (since you can't reuse the one you already picked), $k - 2$ choices for the third, and so on. All together, this gives you $k \cdot (k - 1) \cdot (k - 2) \cdots 1 = k!$ permutations in total.

So there's no need to get generating functions involved to count permutations, but they will be helpful for answering a slightly trickier question. A permutation is a **derangement** if none of the numbers ends up in its original place. For example, 312 is a derangement of 3, but 321 is not, because the 2 stayed in the second position. Of the $4! = 24$ permutations of 4, only 9 are derangements: they are 2143, 2341, 2413, 3142, 3412, 3421, 4123, 4312, and 4321. Our goal will be to find a formula for the number of derangements of k .

Let's call this number D_k . (So, for example, the previous computation demonstrates that $D_4 = 9$.) There's a nice relationship between the number of permutations and the number of derangements that will help us find a formula for D_k . Any permutation, whether it's a derangement or not, has some number of **fixed points**, that is, numbers that remain in their original positions. For example, in the permutation 32415, the fixed points are the 2 and the 5. The thing to observe is that, once you've identified the fixed points, the permutation acts as a derangement on all the *remaining* numbers. (If not, one of them would have been a fixed point!)

This gives us a clever way to relate the number of permutations to the number of derangements: to specify a permutation of k , I can first pick some numbers, say p of them, to be the fixed points, and then pick a derangement of the remaining $k - p$ numbers. Since there are $\binom{k}{p}$ ways to pick the fixed points and D_{k-p} ways to pick the derangement, we see that

$$k! = \binom{k}{0}D_k + \binom{k}{1}D_{k-1} + \binom{k}{2}D_{k-2} + \cdots + \binom{k}{k}D_0.$$

The right side of this equation looks *almost* like a convolution of two sequences, which would let us rewrite it in terms of a product of generating functions, but there's one problem. If we really had a convolution, say of two sequences (a_0, a_1, \dots) and (b_0, b_1, \dots) , the k 'th term would look like

$$a_0 b_k + a_1 b_{k-1} + a_2 b_{k-2} + \dots + a_k b_0.$$

In other words, it would be a sum whose p 'th term comes from multiplying something depending just on p by something depending just on $k - p$. In our equation, we don't quite have this — the terms in our sum look like $\binom{k}{p} D_{k-p}$, and unfortunately $\binom{k}{p}$ depends on k as well as p .

Luckily, we can solve this if we use the formula for $\binom{k}{p}$, which was $\frac{k!}{p!(k-p)!}$. If you plug this in and then divide everything by $k!$, we're in much better shape: we get

$$1 = \frac{1}{0!k!} D_k + \frac{1}{1!(k-1)!} D_{k-1} + \frac{1}{2!(k-2)!} D_{k-2} + \dots + \frac{1}{k!0!} D_0.$$

Now it looks like a convolution! What we've learned is that convolving the sequence $(\frac{1}{0!}, \frac{1}{1!}, \frac{1}{2!}, \dots)$ with the sequence $(\frac{D_0}{0!}, \frac{D_1}{1!}, \frac{D_2}{2!}, \dots)$ gives us the sequence $(1, 1, 1, \dots)$.

What does this mean in terms of generating functions? We already know the generating function of the all-1's sequence from the last section: it's $1/(1-x)$. The generating function of the $\frac{1}{k!}$ sequence is probably familiar from calculus: recall that

$$e^x = \frac{1}{0!} + \frac{x}{1!} + \frac{x^2}{2!} + \dots$$

The last one we'll just have to make up a name for, say

$$d(x) = \frac{D_0}{0!} x + \frac{D_1}{1!} x + \frac{D_2}{2!} x^2 + \dots$$

When you rewrite our equation with the convolution in terms of these generating functions, you get $e^x d(x) = 1/(1-x)$. Luckily, this is very easy to solve: we see that

$$d(x) = \frac{e^{-x}}{1-x}.$$

(After the whole discussion of formal power series in the last section, it would be understandable if you were a bit squeamish about the appearance of e^x here. Isn't that an actual function, not just a formal power series? It is, of course, but for the computation we're doing here there's no reason you have to know that! If we wanted to stick purely to formal power series, we could *define* e^x to be the formal power series whose k 'th term is $x^k/k!$. For our argument to work this way, you'd then have to prove that when you multiply this formal power series by the formal power series for e^{-x} , you get 1. You can in fact do this without any calculus, although it's somewhat tricky, and we won't do it here. It's related to the fact that the alternating sum $\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \dots \pm \binom{n}{n}$ is always zero unless $n = 0$.)

From here, it's not that long a road to the promised formula for D_k . Rather than spell out all the steps here, I'll leave it as an exercise and just state the final answer: you end up with

$$\frac{D_k}{k!} = \frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \dots \pm \frac{1}{k!}.$$

In other words, you alternate plus and minus signs on each successive term, meaning that the sign on the $1/k!$ term at the end depends on whether k is odd or even. (If you try to prove this

yourself, try directly multiplying the power series for e^{-x} by the power series for $1/(1-x)$. What does the coefficient of x^k end up being?)

You could of course multiply this by $k!$ to get D_k by itself, but the way it's written here opens up another intriguing interpretation: because there are $k!$ permutations in total, the left side gives the *probability* that a randomly chosen permutation is a derangement. If you plug -1 into the power series for e^x , you get that

$$e^{-1} = \frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \cdots,$$

which gives us the surprising fact that as k gets large, the probability that a random permutation of k is a derangement approaches $1/e$.

In fact, it approaches $1/e$ quite quickly: because this is an alternating series, the sum on the right side of our equation is within $1/(k+1)!$ of the limit. This means that D_k is actually always the closest integer to $k!/e$ when k is at least 1, so you can also compute D_k by just dividing $k!$ by e and then rounding! For example, $4!/e \approx 8.8291$, and we counted at the start of the section that $D_4 = 9$.

5 The Fibonacci Numbers

We'll close with an application of generating functions that doesn't involve any convolutions. It comes from the famous **Fibonacci sequence**. This is the sequence of numbers which starts with $F_0 = 0$ and $F_1 = 1$, and where every term from F_2 onward is the sum of the previous two terms. The sequence starts with 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

Let's see if we can write down the generating function for the Fibonacci sequence. Say

$$\begin{aligned} b(x) &= F_0 + F_1x + F_2x^2 + F_3x^3 + \cdots \\ &= x + x^2 + 2x^3 + \cdots. \end{aligned}$$

The fact that each term (after the first two) is the sum of the previous two can be expressed pretty compactly as a fact about $b(x)$ using the following trick: multiplying a generating function by x just shifts all the coefficients over by one. This means that

$$xb(x) + x^2b(x) = F_0x + (F_0 + F_1)x^2 + (F_1 + F_2)x^3 + (F_2 + F_3)x^4 + \cdots.$$

When you do this, the coefficient on x^k (for $k \geq 2$) ends up being $F_{k-1} + F_{k-2}$, which is just equal to F_k . In other words, $xb(x) + x^2b(x)$ is *almost* just $b(x)$ itself: the only difference is that (since $F_0 = 0$) we're missing the x at the beginning of $b(x)$. This is easy to fix, though: just add it back on. We end up with the equation

$$b(x) = xb(x) + x^2b(x) + x.$$

This is fairly straightforward to solve for $b(x)$, giving us our generating function

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

Let's try to turn this into a formula for the Fibonacci numbers. We already know how to turn $1/(1-x)$ into a power series, and by plugging in the right values for x , we could extend this to

anything with a linear polynomial in the denominator. This suggests that it might be nice to find an expression for $b(x)$ involving linear denominators, rather than the quadratic we've got now.

You might or might not know that there's a fairly mechanical way to do this called *partial fraction decomposition*. Going through all the steps in detail doesn't make for the most exciting reading, so we'll do it very quickly here, and I encourage you to fill in the gaps if you're interested.

The first step is to factor that quadratic that shows up in the denominator. Using the quadratic formula, we can get that $x^2 + x - 1 = (x + \phi)(x + \bar{\phi})$, where

$$\phi = \frac{1 + \sqrt{5}}{2}; \quad \bar{\phi} = \frac{1 - \sqrt{5}}{2}.$$

This number ϕ is famous: it's called the **golden ratio**, and it has a lot of delightful properties which we sadly don't have space for here. The only thing we'll need for the present conversation is its role in the factorization of this particular polynomial.

The next step in the partial fractions procedure is to write

$$\frac{-x}{x^2 + x - 1} = \frac{-x}{(x + \phi)(x + \bar{\phi})} = \frac{p}{x + \phi} + \frac{q}{x + \bar{\phi}}$$

and solve for p and q . After some straightforward but tedious algebra, you end up with

$$p = -\frac{\phi}{\sqrt{5}}, \quad q = \frac{\bar{\phi}}{\sqrt{5}},$$

and so

$$b(x) = \frac{1}{\sqrt{5}} \left(\frac{\bar{\phi}}{x + \bar{\phi}} - \frac{\phi}{x + \phi} \right).$$

This is a big improvement: now all that's left to do is to plug in our formula for the power series of $1/(1-x)$. To make the algebra slightly easier, it will help to first work out that

$$\frac{a}{x+a} = \frac{1}{1+x/a} = \frac{1}{1-(-x/a)} = 1 + (-x/a) + (-x/a)^2 + \dots$$

When you plug this into our new formula for $b(x)$, the k 'th term ends up being

$$\frac{1}{\sqrt{5}} \left(\left(-\frac{1}{\bar{\phi}} \right)^k - \left(-\frac{1}{\phi} \right)^k \right) x^k.$$

This can be made to look somewhat nicer using the fact (which you can verify yourself if you like) that $-1/\bar{\phi} = \phi$ and $-1/\phi = \bar{\phi}$. With this in place, we finally get the nice formula

$$F_k = \frac{1}{\sqrt{5}} (\phi^k - \bar{\phi}^k).$$

Since $|\bar{\phi}| < 1$, the second term inside the parentheses gets small pretty quickly, so $\phi^k/\sqrt{5}$ is a pretty good approximation to the Fibonacci sequence for large k . In particular, this gives a proof of the somewhat famous fact that the ratio between two consecutive Fibonacci numbers approaches the golden ratio as you go further out in the sequence. (In fact, $|\bar{\phi}^k/\sqrt{5}|$ is always less than $\frac{1}{2}$, so kind of like we saw for derangements in the last section, you can also compute F_k by just rounding $\phi^k/\sqrt{5}$ to the nearest integer.)

6 Final Thoughts

If you don't worry about the details, you could describe all four of the examples we just went through — binomial coefficients, multiset binomial coefficients, derangements, and Fibonacci numbers — in essentially the same way. We started with some sequence of numbers we wanted to understand, then found some equation expressing some relationship that the numbers in the sequence had to each other. In each case, we found a way to express this relationship as an equation involving the generating function of the sequence, which we could solve for the generating function itself, and finally we went from here to a formula for the sequence we wanted.

One thing worth emphasizing is how mechanical this process can be once we have our hands on the generating function. This is one of the nicest features of this method: it might take some thinking to produce the generating function, but from there it's often just a matter of "turning the crank" to get an expression for the coefficients. I should mention that things are not always so nice! Later in this series we'll explore some examples where it's much easier to find the generating function for a sequence than it is to produce a formula for the terms, and even some where such a formula is not even known.

Even when this happens, the generating function can often be a useful source of information, even if that information is not as complete as it was in the cases we went through here. There's much more to be said here, and I hope you'll stick around!