Lesson M4.1: Introduction to Generating Functions

Lesson M4.1: Introduction to Generating Functions

Learning Goals

- Define and describe ordinary generating functions and their uses
- Use generating functions to solve problems at an appropriate level

Textbook References

Grimaldi: 9.1

Keller and Trotter: 8.1-8.3

Levin: 5.1; 2.1 and 2.2 may also be a useful refresher on sequences.

Lesson

In the last module, we learned that Rook polynomials are a convenient structure for storing a sequence of rook numbers in an indexed, organized, and concise way. That was so nice, we want to generalize it- because there was nothing special about rook numbers; we can do this with any sequence!

Suppose we care about the sequence

$$(s_n)_{n\geq 0} = (0, 1, 0, 0, 2, 0, 0, 0, 3, 0, 0, 0, 0, 4, \ldots).$$

This is a nice pattern- and we notice both that:

- The sequence can be described by its nonzero terms
- The nonzero terms are very sparse

This combination of observations means that by writing down or storing information about the whole sequence, we are being really inefficient!

To efficiently describe our sequence, we'd like to store only the nonzero terms, along with a record of which index they belong to- polynomials are a good way to do that.

We can again write the n^{th} term of our sequence as the coefficient of x^n. In this general setting, the result is a *power series* (a fancy name for a potentially infinite polynomial) that encodes our sequence. In this case,

$$\begin{aligned} p(x) &= s_0 x^0 + s_1 x + s_2 x^2 + s_3 x^3 + \cdots \\ &= 0 + 1 \cdot x + 0x^2 + 0x^3 + 2x^4 + 0x^5 + 0x^6 + 0x^7 + 3x^8 + \cdots \\ &= x + 2x^4 + 3x^8 + 4x^{13} + \cdots \end{aligned}$$

This power series, in many fewer terms, actually encodes our entire sequence without losing any information. A power series that is currently being understood to encode a particular sequence is called a *generating function.* To be precise:

Definition: An *ordinary generating function* (OGF) encodes the terms of a sequence $(s_n)_{n \geq 0}$ as the coefficients of x^n for $n \geq 0$.

That is, the OGF of $(s_n)_{n \geq 0}$ is $f(x) = \displaystyle\sum_{n=0}^{\infty} s_n x^n$.

In this case, we say the function $f(x)$ generates the sequence $(s_n)_{n \geq 0}$.

Why do we care about doing this? Well, rook polynomials did two things for us: allowed us to efficiently encode the rook numbers of chessboards, and also exploit the structure of polynomial/power series multiplication to find information about larger chessboards.

We have other sequences we care about, often associated with combinatorial objects- so again we'd like to know the most efficient way to store them (by allowing the removal of zero terms) and also set up a structure where we might be able to combine them somehow to learn about new things.

Let's begin by looking at an example where we can make our lives a lot easier by replacing some of the other tools we have with a couple of generating functions.

Example: How many non-negative integer solutions are there to $a+b+c=10$, where

$$\begin{gathered}2 \le a \le 4 \\ 3 \le b \le 5 \\ 2 \le c \le 5\end{gathered}$$

We already know a couple of ways to solve this problem:

Strategy 1: List solutions explicitly (there are 8): this is reasonably possible here, but isn't usually a viable approach so we don't really want to make it our standard plan.

Strategy 2: Substitute to move each of the lower bounds up to 0, and use PIE to address the upper bounds. We know how to do this and it always works, but it seems like overkill for such a small problem!

Strategy 3: Represent possible values of each variable with a polynomial:
For each variable, the coefficient of x^i in the corresponding polynomial is the number of different ways the variable could attain the value i. Thus a term x^i has a nonzero coefficient (and appears in the polynomial) exactly when i is a possible value of the variable.

In this problem, there is only one way a particular variable could have a specific value, so all coefficients are 1.

To use this strategy, then, we begin by writing the following polynomials:

$$\begin{aligned}
2 \le a \le 4 \qquad & \text{is represented by } f_a(x) = x^2 + x^3 + x^4 \\
3 \le b \le 5 \qquad & \text{is represented by } f_b(x) = x^3 + x^4 + x^5 \\
2 \le c \le 5 \qquad & \text{is represented by } f_c(x) = x^2 + x^3 + x^4 + x^5
\end{aligned}$$

We can then multiply the polynomials for each variable to get a generating function that describes the whole situation, or all possible ways to combine all possible values of each variable (whether or not we get the desired sum).

$$\begin{aligned}
f(x) &= f_a(x) f_b(x) f_c(x) \\
&= (x^2 + x^3 + x^4)(x^3 + x^4 + x^5)(x^2 + x^3 + x^4 + x^5) \\
&= x^7 + 3x^8 + 6x^9 + 8x^{10} + 8x^{11} + 6x^{12} + 3x^{13} + x^{14}
\end{aligned}$$

In this polynomial, the coefficient of x^i is the number of different ways we can get a sum of i using the allowed values of each of our variables.

Thus the number of solutions to our equation (where $a+b+c=10$) is the coefficient of x^{10}, which is 8, as expected.

Notice what just happened here-- we solved infinitely many problems simultaneously! We only needed the coefficient of x^{10} to solve this problem, but we have the coefficient of every power of x (most of them are just 0).

So not only have we solved our problem, we have actually done all the work to solve *any* problem where $a+b+c=n$ under these same bounds.

A new strategy, then, is to solve problems like this by representing possibilities for each part with a generating function, then multiplying to get the big picture. Not only is this process pretty simple and efficient, it nicely allows us to account for more interesting conditions placed on each part of the problem.

Example: Find an OGF that gives the number of non-negative integer solutions to the equation

$$q+r+s+t=n,$$

where

$$\begin{array}{ll} q \leq 4 & s \text{ is odd} \\ r > 7 & t \text{ is even} \end{array}$$

Let's start with a polynomial for each variable:

$f_q(x) = 1 + x + x^2 + x^3 + x^4$ since the possible values of q are 0, 1, 2, 3, and 4.

Similarly, r may take on any value of at least 8, so $f_r(x) = x^8 + x^9 + x^{10} + \cdots = \sum_{i=8}^{\infty} x^i$.

We approach s and t in the same way, with some care given to the way we write our summations (which are often the best way to carefully and concisely express the terms of our power series):

$$f_s(x) = x + x^3 + x^5 + x^7 + \cdots = \sum_{i=0}^{\infty} x^{2i+1}$$

$$f_t(x) = 1 + x^2 + x^4 + x^6 + \cdots = \sum_{i=0}^{\infty} x^{2i}$$

So overall, the number of solutions to our equation is the coefficient of x^n in the generating function

$$f(x) = f_q f_r f_x f_t(x) = (1 + x + x^2 + x^3 + x^4)\left(\sum_{i=8}^{\infty} x^i\right)\left(\sum_{i=0}^{\infty} x^{2i+1}\right)\left(\sum_{i=0}^{\infty} x^{2i}\right).$$

Now you try!

<u>Problem:</u> A hungry college student wants to order a lot of chicken nuggets. The restaurant sells nuggets in boxes of 2, 3, or 7. In how many different ways can they order n nuggets?

Hint: there is only one way to order a box of a particular size. The size of the boxes determines how many nuggets the ones from each kind of box can contribute to your total!

We want to find the number of non-negative integer solutions to the equation $2a + 3b + 7c = n$.

We start by writing down functions that represent the possible contributions of each *term* to the total:

$2a$ can contribute 0, 2, 4, 6, etc., so is represented by

$$f_a(x) = 1 + x^2 + x^4 + \cdots = \sum_{i=0}^{\infty} x^{2i}.$$

Similarly, $f_b(x) = \sum_{i=0}^{\infty} x^{3i}$ and $f_c(x) = \sum_{i=0}^{\infty} x^{7i}$.

So the number of possible solutions is the coefficient of x^n in

$$f(x) = f_a f_b f_c(x) = \left(\sum_{i=0}^{\infty} x^{2i}\right)\left(\sum_{i=0}^{\infty} x^{3i}\right)\left(\sum_{i=0}^{\infty} x^{7i}\right).$$