Generating Functions

Alexander Kroitor 1787469

24/05/2019

Probability and Statistics,

Linear Algebra II

Contents

1 Abstract

This paper looks at the fundamental ideas behind, and the important rules for dealing with, generating functions. Four types of generating functions are looked at: ordinary power, exponential, moment, and Dirichlet. A focus is placed upon the applications of the moment generating function and its uses in probability analysis. Linear Algebra is looked at as an alternative to using generating functions, along with a limited trip into the Jordan Canonical Form and how to both construct it and use it in the appropriate ways.

2 Introduction

2.1 What is a Sequence and Motivations

We call an ordered collection of objects a *sequence*. Sequences appear quite often in mathematics, and have wide-ranging applications. While it is often simple to talk about a sequence recursively (having a relationship between the n-th element and previous elements), it is often more useful to figure out how to define every element of a sequence without referring to previous elements. For example, the popular *Fibonacci Sequence* is defined in such a recursive fashion:

$$f_n = f_{n-1} + f_{n-2} \qquad (n \geq 2;\ F_0 = 0;\ F_1 = 1)$$

Calculating the first elements of the set we get:

$$\{f_i\} = 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...$$

Sequences don't need to be so simply defined, as the sequence of the maximum value reached for a given starting value for the Collatz conjecture '3n+1' sequence (sequence A025586 in the OEIS):

$$\{c_i\} = 1, 2, 16, 4, 16, 16, 52, 8, 52, 16, ...$$

Sequences don't even have to be explicitly defined at all, as the sequence of the prime numbers (sequence A000040 in the OEIS):

$$\{p_i\} = 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...$$

One can imagine fairly easily that sequences pop up all over the place in mathematics, but we still face a big problem: how do we come up with these sequences? It was fairly easy to build up the first sequence, but how do we find other, more complex, sequences? To answer that we turn to generating functions.

2.2 Why bother using Generating Functions?

Consider again the Fibonacci sequence.

Constructing the first few elements of this sequence by directly adding the previous two terms isn't hard, but it sure is harder than what we want. What we want is a way to find a specific element in the sequence without all the long addition involved. This may seem quite difficult, but becomes very simple once we start using some generating functions or linear algebra techniques. For now, let's introduce the fundamental idea of generating functions.

2.3 What is a Generating Function?

Consider a sequence $\{a_n\}$.

The fundamental idea behind generating functions lies behind the fact that we can, in a sense, turn this sequence into a function A(x) by attaching each element of the *sequence* A to a corresponding infinite degree polynomial (in the case of ordinary power series generating functions) *function* A(x). We generally attach the n-th element in the sequence to the x^n term of the polynomial. This is done by assigning a_n as the coefficient of the x^n in the polynomial.

$$\{a_n\} = a_0, a_1, a_2, a_3, ... \Rightarrow A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ...$$

Definition: The ordinary power series generating function of a sequence $\{a_n\}_0^\infty$ is a function A(x) such that:

$$A(x) = \sum_{n=0}^{\infty} a_n x^n \tag{1}$$

The idea behind generating functions is NOT to evaluate this function at any particular x (though we may do so if we wish; if that gives us anything useful is another issue). The fundamental idea is to artificially take a sequence and turn it into a function in a specific way, specifically so that we can spot some pattern in the coefficients of the power series representation of the function (which we declared equal to the sequence in the first place). This specific type of generating functions is called an ordinary power series generating function, or an OPSGF, and it is both the most common and the first type of generating function that we will look at.

This rendition of the sequence is certainly useful; if we assume that we have a different way of describing the function A(x), something that we will spend quite a bit of time on, we can do exciting things like:

$$A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ...$$
$$A(0) = a_0$$

or even

$$\begin{aligned} A(1) &= a_0 + a_1 + a_2 + a_3 + ... \\ &= \sum_{n=0}^{\infty} a_n. \end{aligned}$$

One can imagine that this would be quite useful for identifying certain elements; if we can somehow find a way to express the coefficient of x^n in the expansion of a generating function F we can identify it as the n-th element in whatever sequence that F is generating.

3 Theory

3.1 How do we set up a generating function?

The process of constructing an ordinary power series generating function is as follows [taken from Wil94]:

Given: a recurrence formula that is to be solved by the method of generating functions.

1. Give a name to the generating function that you will look for, and write out that function in terms of the unknown sequence (e.g., call it A(x), and define it to be $\sum_{n\geq 0} a_n x^n$).
2. Multiply both sides of the recurrence by x^n, and sum over all values of n for which the recurrence holds.
3. Express both sides of the resulting equation explicitly in terms of your generating function A(x).
4. Solve the resulting equation for the unknown generating function A(x).
5. If you want an exact formula for the sequence that is defined by the given recurrence relation, then attempt to get such a formula by expanding A(x) into a power series by any method you can think of. In particular, if A(x) is a rational function (quotient of two polynomials), then success will result from expanding in partial fractions and then handling each of the resulting terms separately.

With this process in mind, let's try applying it to a very simple recurrence relationship:

$$a_{n+1} = 2a_n + 1 \qquad (n \geq 0, a_0 = 0)$$

Applying step 1 we declare that we want our generating function A(x) to be equal to $\sum_{n\geq 0} a_n x^n$.

Applying step 2 we multiply both sides of the relationship by x^n and then hit both sides with a $\sum_{n\geq 0}$ from the left:

$$\sum_{n\geq 0} a_{n+1}x^n = 2\sum_{n\geq 0} a_n x^n + \sum_{n\geq 0} x^n$$
$$\sum_{n\geq 0} a_{n+1}x^n = 2A(x) + \frac{1}{1-x}$$

Applying step 3 is more complicated, but with by expanding and fiddling with the left term we can change it into something similar to A(x):

$$\begin{aligned}
\sum_{n\geq 0} a_{n+1}x^n &= a_1 + a_2x + a_3x^2 + ... \\
&= a_1 + a_2x + a_3x^2 + ... * \frac{x}{x} \\
&= \frac{a_1x + a_2x^2 + a_3x^3 + ...}{x} + \frac{a_0 - a_0}{x} \\
&= \frac{a_0 + a_1x + a_2x^2 + a_3x^3 + ...}{x} - \frac{a_0}{x} \\
&= \frac{A(x)}{x} - \frac{a_0}{x} \\
&= \frac{A(x)}{x}
\end{aligned}$$

Leaving us with:

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

Applying step 4 here is simple algebraic manipulation:

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

At this point we have our generating function A(x), but it doesn't really tell us much; by expanding the generating function using partial fractions we can reveal something about our sequence, which fulfills step 5:

$$\begin{aligned}A(x) &= \frac{x}{(1-x)(1-2x)} = \frac{N}{1-2x} + \frac{M}{1-x} \\ &= \frac{1}{1-2x} - \frac{1}{1-x} \\ &= \sum_{n\geq 0}(2x)^n - \sum_{n\geq 0} x^n \\ &= (2^0 + 2^1x + 2^2x^2 + ...) - (1 + x + x^2 + ...) \\ &= (2^0 - 1) + (2^1 - 1)x + (2^2 - 1)x^2 + ...\end{aligned}$$

At this point something not-so-obvious is clear; the coefficient of the n-th element in our function A(x) is $2^n - 1$, which of course means that the n-th element of our original sequence is $2^n - 1$! By using generating functions we have found a hidden way of finding each element in our sequence. Now that we have discussed the basics of generating functions, let us digress and talk about the same way we can approach this sort of problem with linear algebra.

3.2 The Linear Algebra Approach

3.2.1 How do we use the linear algebra approach?

If you have read this paper up until this point, you have likely already taken a linear algebra class, and are thus familiar with the process of multiplying matrices and vectors together. This is worth mentioning, as many people may prefer to use the linear algebra approach merely because they are used to the concepts involved.

The linear algebra approach to this sort of recurrence problem relies mostly on the diagonalization of a suitable matrix; how we will find this matrix will be discussed shortly.

Looking at an recurrence relationship, say $a_{n+2} = Ca_{n+1} + D_n$ with $n \geq 0$, we can in a sense store the current elements of the recurrence relationship that we are looking at in a vector and write:

$$\vec{v}_n = \begin{pmatrix} a_n \\ a_{n-1} \\ a_{n-2} \end{pmatrix} \xrightarrow{*\text{A}} \begin{pmatrix} a_{n+1} \\ a_n \\ a_{n-1} \end{pmatrix} = \vec{v}_{n+1}$$

where the vector elements are being mapped to one higher value of n. We can express each term as a linear combination of the previous terms:

$$\begin{aligned} a_{n+1} &= Ca_n + Da_{n-1} \\ a_n &= a_n \\ a_{n-1} &= a_{n-1} \end{aligned}$$

which is clearly just the same as multiplying the original vector $\left(\begin{smallmatrix} a_n \\ a_{n-1} \\ a_{n-2} \end{smallmatrix}\right)$ by a matrix A with

$$A = \begin{bmatrix} C & D & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

such that

$$A\vec{v}_n = \begin{bmatrix} C & D & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} a_n \\ a_{n-1} \\ a_{n-2} \end{pmatrix} = \begin{pmatrix} Ca_n + Da_{n-1} \\ a_n \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} a_{n+1} \\ a_n \\ a_{n-1} \end{pmatrix} = \vec{v}_{n+1}$$

This matrix multiplication form of sequences can be extended:

$$\begin{aligned} \vec{v}_{n+2} &= A\vec{v}_{n+1} = A(A\vec{v}_n) = A^2\vec{v}_n \\ \vec{v}_{n+i} &= A^i\vec{v}_n \\ \vec{v}_n &= A^{n-i}\vec{v}_i \end{aligned}$$

This can once again be extended using the process of diagonalization. Each matrix P cancels with its inverse matrix P^{-1} when A^i is expanded.

$$A^i = (PD\cancel{P^{-1}})(\cancel{P}D\cancel{P^{-1}})(\cancel{P}D\cancel{P^{-1}})...(\cancel{P}DP^{-1}) = (PD^iP^{-1})$$

In the end this sequence problem boils down to a diagonalization problem. Doing the same recurrence $a_{n+1} = 2a_n + 1 (n \geq 0, a_0 = 0)$ this way is as follows, sticking a 1 in the bottom of $\vec{v}_i$ in order to 'store' it:

$$\begin{aligned} \vec{v}_0 &= \begin{pmatrix} a_0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ A &= \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} = PDP^{-1} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \\ A^i &= PD^iP^{-1} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}^i \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \end{aligned}$$

thus

$$\vec{v}_n = A^n \vec{v}_0 = PD^nP^{-1}\vec{v}_0 = P \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}^n P^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$
$$= P \begin{bmatrix} 1 & 0 \\ 0 & 2^n \end{bmatrix} P^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$
$$= \begin{pmatrix} 2^n - 1 \\ 1 \end{pmatrix} = \begin{pmatrix} a_n \\ 1 \end{pmatrix}$$

so, exactly as we found before using generating functions:

$$a_n = 2^n - 1$$

Clearly this method works just as well for this sequence, although the messy process of diagonalization makes it a little nastier to do by hand, though with a good computer program this becomes less of a hindrance.

3.2.2 Why linear algebra isn't so great after all, and how to get around that

To motivate this section we will start by doing two sample problems using both linear algebra and generating functions to show the shortcomings of each. First we'll do the recurrence relation $a_{n+2} = 3a_{n+1} - 2a_n$ $(a_0 = 3, a_1 = 4)$ and next we'll do the recurrence relation $a_{n+2} = 6a_{n+1} - 9a_n$ $(a_0 = b, a_1 = c)$.

Starting off as we usually do, we take our recurrence relation and apply our procedure to it:

$$\begin{aligned}
a_{n+2} &= 3a_{n+1} - 2a_n \qquad (a_0 = 3, a_1 = 4) \\
\sum a_{n+2}x^n &= \sum 3a_{n+1}x^n - \sum 2a_nx^n \\
\frac{\sum a_nx^n - a_0 - a_1x}{x^2} &= 3\frac{\sum a_nx^n - a_0}{x} - 2\sum a_nx^n \\
\frac{A(x) - 3 - 4x}{x^2} &= \frac{3A(x) - 9}{x} - 2A(x) \\
A(x) &= \frac{3 - 5x}{1 - 3x + 2x^2} \\
&= \frac{1}{1 - 2x} + \frac{2}{1 - x} \\
&= \sum (2x)^n + 2\sum (x)^n \\
[x^n]A(x) &= 2^n + 2
\end{aligned}$$

Where $[x^n]A(x)$ indicates the coefficient of x^n in the function A(x). The linear algebra method goes similarly smoothly:

$$\vec{v}_1 = \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}$$

$$A^n = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix}^n = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}^n \begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 * 2^n - 1 & -2 * 2^n + 2 \\ 2^n - 1 & -2^n + 2 \end{bmatrix}$$

$$\vec{v}_n = A^{n-1}\vec{v}_1$$

$$= \begin{bmatrix} 2 * 2^{(n-1)} - 1 & -2 * 2^{(n-1)} + 2 \\ 2^{(n-1)} - 1 & -2^{(n-1)} + 2 \end{bmatrix} \begin{pmatrix} 4 \\ 3 \end{pmatrix}$$

$$= \begin{pmatrix} 2^n + 2 \\ \frac{2^n+4}{2} \end{pmatrix}$$

$$a_n = 2^n + 2$$

The previous problem is an example of the sort of problem where both methods work without any hitches, but the following one will demand a little more attention for both methods:

$$a_{n+2} = 6a_{n+1} - 9a_n \qquad (a_0 = b, a_1 = c)$$

$$A(x) = \frac{b + cx - 6bx}{1 - 6x + 9x^2} = \frac{b + cx - 6bx}{(1 - 3x)^2}$$

$$= b\frac{1}{(1-3x)^2} + c\frac{x}{(1-3x)^2} - 6b\frac{x}{(1-3x)^2}$$

Here we find ourselves in a small pickle! We know how to turn $\frac{1}{1-x}$ into a power series, but how do we turn $\frac{1}{(1-x)^2}$ into one? In fact it turns out that

$$\sum_n n^i x^n = \sum \left(x\frac{d}{dx}\right)^i x^n = \left(x\frac{d}{dx}\right)^i \sum x^n = \left(x\frac{d}{dx}\right)^i \frac{1}{1-x}$$

$$\left(x\frac{d}{dx}\right) \frac{1}{1-x} = \frac{x}{(1-x)^2} = \sum nx^n$$

Applying this gets us somewhere:

$$
\begin{aligned}
A(x) &= b * \frac{3x}{3x}\frac{1}{(1-3x)^2} + c * \frac{3}{3}\frac{x}{(1-3x)^2} - 6b * \frac{3}{3}\frac{x}{(1-3x)^2} \\
&= \frac{b}{3x}\frac{3x}{(1-3x)^2} + \frac{c}{3}\frac{3x}{(1-3x)^2} - 2b\frac{3x}{(1-3x)^2} \\
&= \frac{b}{3x}\sum n(3)^n(x)^n + \frac{c}{3}\sum n(3)^n(x)^n - 2b\sum n(3)^n(x)^n \\
&= \frac{b}{3}\sum n(3)^n(x)^{n-1} + \frac{c}{3}\sum n(3)^n(x)^n - 2b\sum n(3)^n(x)^n \\
[x^n]A(x) &= \frac{b}{3}(n+1)3^{n+1} + \frac{c}{3}(n)3^n - 2b(n)3^n \\
&= (3b(1-n)+cn)3^{n-1}
\end{aligned}
$$

Which is about it for the generating functions method of solving this problem. The linear algebra approach is a little more upsetting:

$$
\begin{aligned}
\vec{v}_1 &= \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} c \\ b \end{pmatrix} \\
A^n &= \begin{bmatrix} 6 & -9 \\ 1 & 0 \end{bmatrix}^n
\end{aligned}
$$

If we want to apply the typical process of diagonalization to this matrix we calculate the following to solve for the matrices eigenvalues:

$$
\begin{aligned}
0 &= |\lambda I - A| \\
&= \begin{vmatrix} \lambda - 6 & 9 \\ -1 & \lambda \end{vmatrix} \\
&= \lambda^2 - 6\lambda + 9 \\
\lambda_1 &= 3 \\
\lambda_2 &= 3
\end{aligned}
$$

$$
\left[\begin{array}{cc|c} -3 & 9 & 0 \\ -1 & 3 & 0 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & -3 & 0 \\ 0 & 0 & 0 \end{array}\right]
$$

$$
\vec{\varepsilon}_1 = \vec{\varepsilon}_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \Rightarrow P = \begin{pmatrix} 3 & 3 \\ 1 & 1 \end{pmatrix}
$$

This is a big problem! The entirety of diagonalization depends on being able to find an invertible matrix P composed of the eigenvectors of A; by the invertible matrix theorem if the eigenvectors of A are colinear we cannot invert our matrix P to diagonalize A, and thus cannot solve our problem using our current linear algebra technique. This roadblock provides our motivation for the covering of the Jordan Canonical Form (or the Jordan Normal Form), which will help us "diagonalize" even if our P matrices are not invertible.

3.2.3 The Jordan Canonical Form

The Jordan Canonical Form involves taking a matrix A and constructing a similar matrix J (such that $A = PJP^{-1}$) with it. In this case, however, J is not a diagonal matrix holding the eigenvalues of A along its diagonal. Instead, J is of the following form, with λ_n as the n-th eigenvalue of A (and λ_j possibly equal to λ_k, meaning that we can have multiple blocks with the same eigenvalues:

$$J = \begin{bmatrix} J_1 & 0 & \dots & 0 & 0 \\ 0 & J_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & J_{i-1} & 0 \\ 0 & 0 & \dots & 0 & J_i \end{bmatrix}$$

$$J_n = \begin{bmatrix} \lambda_n & 1 & \dots & 0 & 0 \\ 0 & \lambda_n & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \lambda_n & 1 \\ 0 & 0 & \dots & 0 & \lambda_n \end{bmatrix}$$

That is to say that that the matrix J is composed of many blocks, each consisting of a diagonal of the repeating corresponding eigenvalue and 1s all above the diagonal.

We call these J_ns Jordan blocks, and say that our matrix J is composed of many Jordan blocks. We call an n by n Jordan block a block of size n.

Putting a matrix into JCF has two roadblocks that we will have to discuss: how to find the size of the Jordan blocks for corresponding eigenvalues, and how to construct the P matrix.

The method [Linb] for figuring out how to construct our J blocks is as follows:

1. Find the characteristic polynomial for the matrix and express it as $p_A(x) = \prod_i (x - \lambda_i)^{a_i}$, where a_i is the algebraic multiplicity of the eigenvalue λ_i.

2. Calculate the nullity of $(A - \lambda_i)^r$ for r = 1. Repeat this process, increasing r by 1 each time, until $N[(A - \lambda_i)^r]$ is equal to a_i (until it is equal to the exponent of the eigenvalue we're talking about in the first place). If we write these nullities in a sequence we get $0 = N_0 < N_1 < ... < N_{l_k} = a_i$.

3. This sequence gives us information about the sizes of our Jordan blocks: $N_i - N_{i-1}$ gives us the number of Jordan blocks with at *least* size i with our eigenvalue. This knowledge can be used to construct the J matrix.

Of course an example:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$p_A(x) = (x-1)^6$$

$$(A-I)^1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 1 \\ -1 & -1 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow N_1 = 3$$

$$(A-I)^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow N_2 = 5$$

$$(A-I)^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow N_3 = 6$$

$$N_3 - N_2 = 1$$
$$N_2 - N_1 = 2$$
$$N_1 - N_0 = 3$$

Based on this, there is one block with at least size 3, 2 blocks with at least size 2, and 3 blocks with at least size 1. It follows that there is 1 size 3 block, 1 size 2 block, and 1 size 1 block. From this we can construct our matrix J (with the blocks positioned wherever we want them):

Now that we have discussed the basic idea behind finding the J matrix in the JCF decomposition, let's talk about how to find the P matrix (the weak-point that got us before).

Instead of simply finding the eigenvectors of our matrix, which was precisely what got us before (we only had 1 eigenvector but we needed 2), we find the generalized eigenevectors of our matrix [Lina]. While a *normal* eigenvector is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

defined as $\vec{x}$ such that $(A - \lambda I)\vec{x} = \vec{0}$, a *generalized* eigenvector is defined as being a vector $\vec{x}$ such that $(A - \lambda I)^k \vec{x} = \vec{0}$ for some integer k [OS06]. We can think of these eigenvectors as a chain; the first 'real' eigenevector v_1 gets mapped to $\vec{0}$ by multiplication by $(A - \lambda I)$. The second generalized eigenevector v_2 gets mapped to v_1 by multiplication by $(A - \lambda I)$ (and since $\vec{v}_1$ gets mapped to $\vec{0}$ by $(A - \lambda I)$ then v_2 gets mapped to $\vec{0}$ by $(A - \lambda I)^2$. We continue this process of solving $(A - \lambda I)\vec{v}_i = \vec{v}_{i-1}$ until we find enough generalized eigenvectors to fill up the corresponding spots in our P matrix. For a new A (one that isn't a humongous 6 by 6 matrix) we have:

$$A = \begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 4 & -4 & 7 \end{bmatrix} \Rightarrow \vec{v}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

This matrix has only one eigenvector for its single eigenvalue of 5. We now solve for the next element in the chain, finding $\vec{v}_2$ such that $(A - 5I)\vec{v}_2 = \vec{v}_1$. This gives us the vector:

$$\vec{v}_2 = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

Repeating this procedure gives us our $\vec{v}_3$:

$$\vec{v}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

We can now see where the whole idea of a chain comes in:

$$\vec{v}_3 \xrightarrow{\text{*(A-5I)}} \vec{v}_2 \xrightarrow{\text{*(A-5I)}} \vec{v}_1 \xrightarrow{\text{*(A-5I)}} \vec{0}$$

We then use these three vectors to form our P matrix:

$$P = \begin{bmatrix} | & | & | \\ \vec{v}_1 & \vec{v}_2 & \vec{v}_3 \\ | & | & | \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

In general, if we have a J matrix consisting of i Jordan Blocks with size a_i and eigenvalues λ_i, our P matrix is the following:

$$P = \begin{bmatrix} | & & | & | & & | \\ \vec{v}_{\lambda_1,1} & \dots & \vec{v}_{\lambda_1,a_1} & \vec{v}_{\lambda_2,1} & \dots & \vec{v}_{\lambda_i,a_i} \\ | & & | & | & & | \end{bmatrix}$$

If there are repeats in the eigenvalues for different Jordan Blocks, the appropriate eigenvector is chosen to start off the chain for that Jordan Block (read: the eigenvector that creates a chain of length a_i).

After finishing our P matrix we compose our P, J, and P^{-1} to get the following JCF decomposition:

$$\begin{aligned} A &= \begin{bmatrix} 4 & 1 & 0 \\ 1 & 4 & 1 \\ 4 & -4 & 7 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 5 & 1 & 0 \\ 0 & 5 & 1 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -2 & 1 \end{bmatrix} \end{aligned}$$

Which gives us our answer. Finishing the motivating problem, and noting that $A^n = PJ^nP^{-1}$, we apply JCF decomposition to our matrix instead of just normally diagonalizing:

$$\begin{aligned} \vec{v}_1 &= \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} c \\ b \end{pmatrix} \\ A^n &= \begin{bmatrix} 6 & -9 \\ 1 & 0 \end{bmatrix}^n \end{aligned}$$

$$p_A(x) = (x-3)^2$$

$$\begin{aligned} N_1 &= 1 \\ N_2 &= 2 \end{aligned}$$

$$\begin{aligned} N_2 - N_1 &= 1 \\ N_1 - N_0 &= 1 \end{aligned}$$

As such there is only one block of size 2. Our first (and only true) eigenvector $\vec{v}_1$ is $\binom{3}{1}$, and the following generalized eigenvector we get from that is $\binom{1}{0}$. From these we get our Jordan Normal form of A, which can then be turned into A^n.

$$\begin{aligned} A^n &= \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 0 & 3 \end{bmatrix}^n \begin{bmatrix} 0 & 1 \\ 1 & -3 \end{bmatrix} \\ &= \begin{bmatrix} 3^n + 3^n * n & -3 * 3^n * n \\ \frac{3^n * n}{3} & 3^n - 3^n * n \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \vec{v}_n &= A^{n-1}\vec{v}_1 \\ &= \begin{bmatrix} 3^{n-1} + 3^{n-1} * (n-1) & -3 * 3^{n-1} * (n-1) \\ \frac{3^{n-1} * (n-1)}{3} & 3^{n-1} - 3^{n-1} * (n-1) \end{bmatrix} \begin{pmatrix} c \\ b \end{pmatrix} \\ &= \begin{pmatrix} \frac{3*3^n*b - 3*3^n*b*n + 3^n*c*n}{3} \\ \frac{6*3^n*b - 3^n*c - 3*3^n*b*n + 3^n*c*n}{9} \end{pmatrix} \end{aligned}$$

Which, not so coincidentally, corresponds to $a_n = (3b(1-n) + cn)3^{n-1}$. Clearly the linear algebra way works just as well in the sense that it gives us the same answer as the generating function approach, but for long recurrences it requires large matrices, and the cumbersome nullity/block size finding that come with it. In these situations generating functions are cleaner and nicer to use.

A parallel between the two different ways the methods go off course can be seen in the linear algebra approach. When we calculate the eigenvectors for our matrix A, what we are actually doing is finding some "eigenconditions", initial conditions that (if used) will result in an eigensequence (or one that increases by a fixed factor with each iteration). How these eigensequences manifest in the linear algebra approach is quite clear; they go through the P^{-1} matrix and get turned into something in the eigenbasis, then getting multiplied by the interior D matrix (which contains all the eigenvalues of this sequence), and then goes along its merry way having been multiplied by some constant. What happens in the generating function approach is more subtle; when the eigenconditions are initially put in, all but one of the bottom factors (such as $(1 - ax)$) vanish, cancelling with the top part of the fraction provided by the eigenconditions. This leaves us with a single term (say a^n) which ends up constructing the rest of the sequence and multiplying each term by a. Obviously the various terms in the denominator of our generating function are related to the eigenvalues of the linear algebra approach; they're the same! The b_i's in $\frac{c_0+c_1x+c_2x^2+...}{(1-b_0x)(1-b_1x)(1-b_2x)...}$ are in fact the same thing as the eigenvalues, and as such, finding the eigenvalues for our linear algebra method consists of factoring a big polynomial. Finding the corresponding eigenvectors (or starting conditions) for b_i (also called λ_i) merely consists of solving for the c's in $c_0 + c_1x + c_2x^2 + ... =$

$\frac{(1-b_0x)(1-b_1x)(1-b_2x)...}{(1-b_ix)}$. The reason the linear algebra method broke down was due to the repeated eigenvalue, which also appeared in the generating functions version as the repeated polynomial $(1-3x)^2$.

It is clear that the generating functions method works infinitely better, as taking a couple of derivatives is easy as crackers compared to finding eigenvalues, finding nullities after putting a matrix to the k-th power, and then finding generalized eigenvectors. As such, we will stop using the linear algebra method and focus more on the generating function approach, as well as its uses and parameters.

3.3 Parameters of Generating Functions

We will now discuss the various rules that can be used to shortcut generating functions, as well as a discussion on a new type of generating function called exponential generating functions (as opposed to ordinary power series generating functions) and their corresponding rules.

3.3.1 Rules of OPSGF's

Some of these rules are included for helpfulness during this brief review of the topic, while some are included for personal interest and consistency [mostly inspired by Wil94].

Rule 1

$$\{a_{n+h}\}_0^\infty \Rightarrow \frac{A(x) - a_0 - a_1x - a_2x^2 - ... - a_{h-1}x^{h-1}}{x^h}$$

We already saw an example of this sort of manipulation before, but now we generalize it:

$$\begin{aligned}\sum a_{n+h}x^n &= a_h + a_{h+1}x + ... \\ &= (a_h + a_{h+1}x + ...) * \frac{x^h}{x^h} \\ &= \frac{(a_hx^h + a_{h+1}x^{h+1} + ...)}{x^h} \pm \frac{a_0}{x^h} \pm \frac{a_1x}{x^h} \pm ... \pm \frac{a_{h-1}x^{h-1}}{x^h} \\ &= \frac{(a_0 + a_1x + ...) - a_0 - a_1x - ... - a_{h-1}x^{h-1}}{x^h} \\ &= \frac{A(x) - a_0 - a_1x - ... - a_{h-1}x^{h-1}}{x^h}\end{aligned}$$

Which shows how shifts affect an OPS generating function in general.

Rule 2

With $P(n)$ as a polynomial:

$$\{P(n)a_n\}_0^\infty \Rightarrow P\left(x\left(\frac{d}{dx}\right)\right)A(x)$$

Earlier on this was demonstrated for $P(n) = n^i$ and $a_n = 1$. Extending this doesn't require much else:

$$\begin{aligned}\sum_n P(n)a_n x^n &= \sum_n \left(\sum_i b_i n^i\right) a_n x^n \\ &= \sum_i b_i \sum_n n^i a_n x^n \\ &= \sum_i b_i \sum_n \left(x\frac{d}{dx}\right)^i a_n x^n \\ &= \sum_i b_i \left(x\frac{d}{dx}\right)^i \sum_n a_n x^n \\ &= P\left(x\left(\frac{d}{dx}\right)\right)A(x)\end{aligned}$$

This rule can be applied easily to say, find the generating function of $a_n = n^3$ as just $\left(x\left(\frac{d}{dx}\right)\right)^3 \frac{1}{1-x} = \frac{x^2+4x+1}{(1-x)^4}$.

Rule 3

$$\left\{\sum_{n_1+n_2+n_3...=n} a_{n_1}b_{n_2}c_{n_3}...\right\}_0^\infty \Rightarrow A(x)B(x)C(x)...$$

This one makes sense after looking at what the coefficient of x^n in the function $A(x)B(x)C(x)...$ is. When we multiply these many polynomials together, we get an x^n term whenever the sum of the powers of some given x's is equal to n. As such we can see that the coefficient of *one* x^n coefficient in the multiplication of $A(x)B(x)C(x)$ may be something like $a_1b_3c_{n-4}$, but to find our new function we want to find the coefficients of all the x^n's. As such we can see the the coefficient of x^n is actually $\sum_{i+j+k=n} a_ib_jc_k$. Extending this logic, we see that the coefficient of $A(x)B(x)C(x)...$ is indeed $\sum_{i+j+k+...=n} a_ib_jc_k...$, and thus the former is the generating function for the latter.

It follows from this that

$$\{ \sum_{n_1+n_2+..n_k=n} a_{n_1}a_{n_2}...a_{n_k}\}_0^\infty \Rightarrow A^k(x)$$

3.3.2 Exponential Generating Functions

Until now we have talked exclusively about ordinary power series generating functions of the form $A(x) = \sum a_n x^n$. We now turn our attention to the family of similar, but different, exponential generating functions.

Definition: The exponential generating function of a sequence $\{a_n\}_0^\infty$ is a function A(x) such that:

$$A(x) = \sum_{n=0}^{\infty} \frac{a_n}{n!} x^n$$

Of course we immediately see an analogue to e^x; by setting $\{a_n\}$ to 1, we get that

$$\{1\} \Rightarrow \sum \frac{1}{n!} x^n = e^x$$

Or in other words, e^x is the exponential generating function for $\{1\}$, the same way $\frac{1}{1-x}$ is the OPSGF for $\{1\}$. The structure behind EGF's make the rules that apply to them occasionally more complex, and occasionally less complex (and thus preferable) to work with. Let's go over them and their justifications.

3.3.3 Rule of EGF's

Rule 1

$$\{a_{n+h}\}_0^\infty \Rightarrow \left(\frac{d}{dx}\right)^h A(x)$$

We look at $A(x) = \sum \frac{a_n}{n!} x^n$, our EGF for $\{a_n\}$, and take a single derivative, seeing the following:

$$\left(\frac{d}{dx}\right)\sum_{n=0}^{\infty}\frac{a_n}{n!}x^n = \sum_{n=1}^{\infty}\frac{na_n}{n!}x^{n-1}$$
$$= \sum_{n=1}^{\infty}\frac{a_n}{(n-1)!}x^{n-1}$$
$$= \sum_{n=0}^{\infty}\frac{a_{n+1}}{(n)!}x^n$$

Which is the EGF for $\{a_{n+1}\}$. If we continue this derivative-taking process h times we get our EGF for $\{a_{n+h}\}$ as $A^{(h)}(x)$.

Rule 2

$$\{P(n)a_n\}_0^{\infty} \Rightarrow P\left(x\left(\frac{d}{dx}\right)\right)A(x)$$

Just the same as the OPSGF vesion, taking the derivative then multiplying by x has the same effect of sticking an n out in front:

$$\sum_n P(n)\frac{a_n}{n!}x^n = \sum_n\left(\sum_i b_i n^i\right)\frac{a_n}{n!}x^n$$
$$= \sum_i b_i \sum_n n^i \frac{a_n}{n!}x^n$$
$$= \sum_i b_i \sum_n \left(x\frac{d}{dx}\right)^i \frac{a_n}{n!}x^n$$
$$= \sum_i b_i \left(x\frac{d}{dx}\right)^i \sum_n \frac{a_n}{n!}x^n$$
$$= P\left(x\left(\frac{d}{dx}\right)\right)A(x)$$

This is the same as the result of the OPSGF version, which makes this rule particularly unexciting.

Rule 3

$$\left\{ \sum_{n_1+n_2+n_3...=n} \frac{a_{n_1} b_{n_2} c_{n_3} ...}{n_1! n_2! n_3! ...} \right\}_0^\infty \Rightarrow A(x)B(x)C(x)...$$

with

$$\left[\frac{x^n}{n!}\right] A(x)B(x)C(x)... = \sum_{n_1+n_2+n_3...=n} \frac{n! a_{n_1} b_{n_2} c_{n_3} ...}{n_1! n_2! n_3! ...}$$

Or, in the case of only two functions A(x) and B(x):

$$\left[\frac{x^n}{n!}\right] A(x)B(x) = \sum_{n_1+n_2=n} \frac{n! a_{n_1} b_{n_2}}{n_1! n_2!} = \sum_{n_1} \binom{n}{n_1} a_{n_1} b_{n-n_1}$$

This comes about from multiplying the EGF's together:

$$\begin{aligned} A(x)B(x)C(x)... &= \left\{ \sum_{n_1} \frac{a_{n_1} x^{n_1}}{n_1!} \right\} \left\{ \sum_{n_2} \frac{a_{n_2} x^{n_2}}{n_2!} \right\} \left\{ \sum_{n_3} \frac{a_{n_3} x^{n_3}}{n_3!} \right\} ... \\ &= \sum_{n_1+n_2+n_3+...=n} \frac{a_{n_1} b_{n_2} c_{n_3} ...}{n_1! n_1! n_1! ...} x^n \end{aligned}$$

3.3.4 Moment Generating Functions

We will now cover a section that is important for probability calculations (and will be important later in a few sections), though we will only discuss the discrete version.

A moment μ_k of a probability distribution is defined as follows[GS97]:

$$\mu_k = E(X^k) = \sum x^k P(X = x)$$

You may notice that μ_1 is the mean of the probability distribution and $\mu_2 - \mu_1^2$ is the variance of the probability distribution. This fact will be used later on.

A moment generating function is so named because, just like our other generating functions, it an interesting sequence tacked along it. Unsurprisingly this sequence is the sequence μ_k of the various moments of the function. We define our moment generating function for an arbitrary probability distribution as follows:

Definition: The moment generating function of a random variable X is a function g(t) such that:

$$g(t) = E(e^{tx}) = \sum_j e^{tx_j} P(X = x_j)$$

But it is also equal to:

$$\begin{aligned} g(t) &= E\left(\sum \frac{t^n x^n}{n!}\right) = \sum \frac{t^n E(x^n)}{n!} = \sum \frac{\mu_n t^n}{n!} \\ &= 1 + \mu_1 t + \frac{\mu_2 t^2}{2!} + \frac{\mu_3 t^3}{3!} + \dots \end{aligned}$$

showing how moments appear in this generating function. How to extract a certain moment is simple[Gri]:

$$\begin{aligned} \frac{d^r}{dt^r}(g(t))\bigg|_{t=0} &= \frac{d^r}{dt^r}\left(E(e^{tx})\right)\bigg|_{t=0} = E\left(\frac{d^r}{dt^r}(e^{tx})\right)\bigg|_{t=0} = E\left(x^r(e^{tx})\right)\big|_{t=0} \\ &= E\left(x^r\right) \\ &= \mu_r \end{aligned}$$

Thus we have a method to find the various moments of a probability distribution; first we construct our g(t) by multiplying our $P(X = x)$ by e^{tx} and summing. We then take the $g(t)$, differentiate it k times. Finally we set $t = 0$ and go from there to find our μ_k.

3.3.5 Dirichlet Generating Functions

As an aside, there is another family of generating functions called the Dirichlet generating functions.

Definition: The Dirichlet generating function of a sequence $\{a_n\}_1^\infty$ is a function A(x) such that:

$$A(x) = \sum_{n=1}^{\infty} \frac{a_n}{n^x}$$

The most famous example of a DGF is the Riemann Zeta function:

$$\zeta(x) = \sum_n \frac{1}{n^x}$$

Which also happens to be the DGF for the sequence $\{1\}$.

This ends the section dedicated to the various rules of both OPSGF's and EGF's. The following sections will contain examples of the applications of generating functions, both OPS and E.

4 Applications of generating functions

This section will be dedicated to various examples that employ generating functions to solve.

4.1 The Fibonacci Sequence

$$a_{n+2} = a_{n+1} + a_n \qquad (n \geq 0;\ a_0 = 0;\ a_1 = 1)$$

Presented at the beginning of this paper, we will now deal with the question "what is the nth term in the Fibonacci Sequence?"

Going around in our usual way, we get this:

$$\begin{aligned} \frac{A(x) - a_0 - a_1 x}{x^2} &= \frac{A(x) - a_0}{x} + A(x) \\ \frac{A(x) - x}{x^2} &= \frac{A(x)}{x} + A(x) \\ A(x) &= \frac{x}{1 - x - x^2} \end{aligned}$$

Solving with partial fractions the usual way, we get a closed form representation of a_n:

$$a_n = [x^n]A(x) = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

Equivalently we could have approached this with an EGF perspective and solved for $[\frac{x^n}{n!}]A(x)$ where:

$$A''(x) = A'(x) + A(x) \qquad (A(0) = 0, A'(0) = 1)$$

and we would have gotten the same thing.

4.2 The Binomial Distribution

The binomial distribution describes an experiment with n identical and independent trials, each with a chance p of success. The experiment describes the probability $P(X = x)$ that the number of successful trials of those n independent will be x. The probability of the sum being x is $P(X = x) = \binom{n}{x} p^x q^{n-x}$, and the possible values for x range from 0 all the way up to n (so we sum from

0 to n). To start we take our $P(X=x)=\binom{n}{x}p^x q^{n-x}$ and multiply it by it by e^{tx}, remembering that $p+q=1$:

$$\begin{aligned} g(t) &= \sum_{x=0}^{n} e^{tx}\binom{n}{x}p^x q^{n-x} \\ &= \sum_{x=0}^{n}\binom{n}{x}(pe^t)^x q^{n-x} \\ &= (pe^t+q)^n \end{aligned}$$

$$\begin{aligned} g'(0) &= \left. npe^t(pe^t+q)^{n-1}\right|_{t=0} = np(p+q)^{n-1} = np = \mu_1 \\ g''(0) &= \left. npe^t(pe^t+q)^{n-2}(npe^t+q)\right|_{t=0} = (np)^2 + npq = \mu_2 \end{aligned}$$

$$\begin{aligned} E(X) &= \mu_1 = np \\ V(X) &= \mu_2 - (\mu_1)^2 = (np)^2 + npq - (np)^2 = npq \end{aligned}$$

Exactly as expected, but with much less work (forgoing the otherwise required discussion of considering each trial separately.

4.3 The Poisson Distribution

The Poisson distribution is another example of a discrete probability distribution. The only parameter of the Poisson distribution is λ, which is the expected number of events given that you wait a certain amount of time. The probability of x events happening in a given time is $P(X=x)=\frac{\lambda^x e^{-\lambda}}{x!}$. The possible range of values for X range from 0 to ∞, so we sum appropriately.

$$\begin{aligned} g(t) &= \sum_{x=0}^{\infty} e^{tx}\frac{\lambda^x e^{-\lambda}}{x!} \\ &= e^{-\lambda}\sum_{x=0}^{\infty}\frac{(e^t\lambda)^x}{x!} \\ &= e^{-\lambda}e^{\lambda e^t} \end{aligned}$$

$$\begin{aligned} g'(0) &= \lambda \\ g''(0) &= \lambda^2 + \lambda \end{aligned}$$

$$\begin{aligned} E(X) &= \lambda \\ V(X) &= \lambda^2 + \lambda - \lambda^2 = \lambda \end{aligned}$$

Which covers the Poisson Distribution, with the absolutely expected mean and variance (given that the Poisson is set up by calling the mean and variance λ).

4.4 The Negative Binomial Distribution

The negative binomial distribution measures the probability that you will wait a certain number of trials (each with success probability p) before getting r successes. This distribution is closely related to the binomial distribution in that while the binomial distribution counts probabilities of getting certain numbers of successes given a fixed number of trials, the negative binomial does the inverse (or maybe the 'negative') and instead counts the probabilities of getting certain numbers of trials given a fixed number of successes. In this situation the possible values of X range from r to ∞.

$$\begin{aligned}
g(t) &= \sum_{x=r}^{\infty} e^{tx}\binom{x-1}{r-1}p^r(1-p)^{x-r} \\
&= p^r\sum_{x=r}^{\infty}(e^t)^{x-r+r}\binom{x-1}{r-1}(1-p)^{x-r} \\
&= (pe^t)^r\sum_{x=r}^{\infty}\binom{x-1}{r-1}((1-p)e^t)^{x-r} \\
&= (pe^t)^r\sum_{x=r}^{\infty}\binom{x-1}{r-1}((1-p)e^t)^{x-r} * \frac{(1-(1-p)e^t)^r}{(1-(1-p)e^t)^r} \\
&= \frac{(pe^t)^r}{(1-(1-p)e^t)^r} * \sum_{x=r}^{\infty}\underbrace{\binom{x-1}{r-1}((1-p)e^t)^{x-r}(1-(1-p)e^t)^r}_{\text{A neg. bin. summed from r to }\infty\text{; equals one.}} \\
&= \frac{(pe^t)^r}{(1-(1-p)e^t)^r} * 1 \\
&= \left(\frac{pe^t}{1-(1-p)e^t}\right)^r
\end{aligned}$$

$$\begin{aligned}
g'(0) &= \frac{r}{p} \\
g''(0) &= \frac{r^2+r(1-p)}{p^2}
\end{aligned}$$

$$\begin{aligned}
E(X) &= \frac{r}{p} \\
V(X) &= \frac{r^2+r(1-p)}{p^2} - \frac{r}{p}^2 = \frac{r(1-p)}{p^2}
\end{aligned}$$

Which once again lines up with what we expect. The geometric distribution is a special case of the negative binomial distribution with r=1. Accordingly, the geometric distribution has $E(X) = \frac{1}{p}$ and $V(X) = \frac{(1-p)}{p^2}$, catching two distributions with one MGF.

This marks the end of the probability distributions we will look at using MGFs, and the end of the content in this paper.

5 Conclusion

Though generating functions are not omnipotent, they come pretty darn close. They can be used to tackle (and quite handily) any simple linear recurrence relationships, and can be used to derive some neat sequences without much effort. They trump the classic linear algebra by quite a bit, as they need none of the complicated JCF shown in this paper. Overall the generating function technique is a powerful one that every aspiring sequentialist should have under their belt. Ideally anyone who has fully read and understood this paper will have all the tools they need to tackle generating functions in the future, and will be able to approach them with some amount of familiarity.

References

[Wil94] Herbert S. Wilf. *Generatingfunctionology*. 2nd ed. Boston: Academic Press, 1994. ISBN: 0127519564.

[GS97] Charles M. Grinstead and J. Laurie Snell. *Introduction to probability*. 2nd rev. ed. Providence RI: American Mathematical Society, 1997. ISBN: 0821807498.

[OS06] Peter J. Olver and Chehrzad Shakiban. *Applied linear algebra*. Upper Saddle River NJ: Prentice Hall, 2006. ISBN: 0131473824.

[Gri] Gleb Gribakin. *Chapter 6: Moment Generating Functions*.

[Lina] Jeff Lindquist. *How to Find Bases for Jordan Canonical Forms Version 2.0*.

[Linb] Jeff Lindquist. *How to Find Jordan Canonical Forms*.