ON THE ELEGANT UTILITY OF GENERATING FUNCTIONS

DAKSH DHEER
(FIRST YEAR, HANSRAJ COLLEGE, 2021)

Abstract

"A generating function is a clothesline on which we hang up a sequence of numbers for display."

Herbert S Wilf

A **generating function** is a tool for encoding an infinite sequence of numbers by treating them as the coefficients of a power series. It is a bridge between discrete mathematics and continuous analysis.

$$\{a_0, a_1, a_2, a_3, \dots\} \longrightarrow \boxed{\text{Generating Function}} \longrightarrow P(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \dots$$

This paper attempts to introduce the concept of a generating function in a general way and explain how it used to solve certain problems algebraically, that is, the basis of its elegance and utility.

The paper starts off introducing the concept of a generating function, accompanied by illustrations to display its power.

For example, **how do we compute the 197^{th} number in the Fibonacci sequence without having to use the recursion and calculating all the previous terms?** There are many approaches to obtain the n^{th} Fibonacci number; this paper evaluates it using generating functions.

The paper would also attempt to answer, using generating functions, a seemingly innocuous question - **how many ways are there to make change for a hundred dollars? a hundred rupees?**

Well, this seemingly unrelated question is a question deep at the heart of the theory of integer partitions - a very interesting area, in its own right, of discrete mathematics, making use of combinatoric number theory.

The only prerequisite required to solve the above problem is acceptance of the following fact:

$$(x^p)(x^q) = x^{p+q}$$

Introduction

A **generating function** is a tool for encoding an infinite sequence of numbers by treating them as the coefficients of a power series. It a bridge between discrete mathematics and continuous analysis.

Herbert S Wilf puts it as

> A generating function is a clothesline on which we hang up a sequence of numbers for display.

Now, that is a marvelous quote because it is exactly how one should think about generating functions.
An alternate perspective can be to think of a generating function as a bag where all the important information about a sequence is stored and can be retrieved or 'generated' later on.

The whole power of generating functions is based around two simple facts:

(1) The product of x^p and x^q is x^{p+q}.
(2) Knowing the coefficients of a polynomial or a power series allows us to manipulate it and extrapolate information.

1. Getting started with Recursions

Consider the following recursive sequence, given that $a_0 = 3$:

$$a_{n+1} = 3a_n - 4 \tag{1}$$

Writing the first few terms, we get:
$a_1 = 5, \quad a_2 = 11, \quad a_3 = 29, \quad a_4 = 83, \quad a_5 = 245$, and so on ...
How do we solve this recursion to gain an explicit form? Well, we can look at it and notice that there is a rough correspondence of the terms of the sequence with powers of 3. Moreover, looking at the first few terms, one can easily guess and check that the sequence is given by:

$$a_n = 3^n + 2, \ \forall n \in \mathbf{N} \cup \{\mathbf{0}\}$$

Now, let us solve the same question using a generating function. We shall define the function as:

$$A(x) = \sum_{n \geq 0} a_n x^n$$

Our strategy is to evaluate $A(x)$ and expand it as a series to be able to read out the coefficients.

To evaluate $A(x)$, we consider $A(x) - 3xA(x)$ and use the relation given in (1)

$$\begin{aligned} A(x) - 3xA(x) &= (a_0 + a_1x + a_2x^2 + \dots) - 3(a_0x + a_1x^2 + a_2x^3 + \dots) \\ &= a_0 + x(a_1 - 3a_0) + x^2(a_2 - 3a_1) + x^3(a_3 - 3a_2) + \dots \\ &= 3 + x(-4) + x^2(-4) + x^3(-4) + \dots \end{aligned}$$

$$\begin{aligned}
A(x)(1-3x) &= 7 - \frac{4}{1-x} \\
A(x) &= \frac{7}{(1-3x)} - \frac{4}{(1-3x)(1-x)} \\
&= \frac{1}{1-3x} + \frac{2}{1-x} \\
&= (1 + 3x + 3^2x^2 + 3^3x^3 + \dots) + 2(1 + x + x^2 + x^3 + \dots) \\
&= 3 + (3^1+2)x^1 + (3^2+2)x^2 + (3^3+2)x^3 + (3^4+2)x^4 + \dots
\end{aligned}$$

Hence, we obtain:

$$A(x) = \sum_{n\geq 0} a_n x^n = \sum_{n\geq 0} (3^n + 2)x^n \implies a_n = 3^n + 2, \ \forall n \in \mathbf{N} \cup \{\mathbf{0}\}$$

One might wonder why we took such a tedious route to solve the same question. After all, we generally aim to reach an efficient solution, right?

Well, it is true that the generating functions approach proved to be quite long here, but this was just an example to illustrate how it is used.

We cannot always guess and check a recursion to obtain a solution, which is what our next example will be.

We shall find an explicit, closed form for the n^{th} Fibonacci number.

2. The Fibonacci Sequence

The Fibonacci Sequence is given by the recursion:

$$F_{n+1} = F_n + F_{n-1} \qquad (n \geq 1; F_0 = 0; F_1 = 1)$$

Consider the generating function:

$$\begin{aligned}
G(x) &= \sum_{n\geq 0} F_n x^n = 0 + 1x + \sum_{n\geq 2} F_n x^n \\
&= x + \sum_{n\geq 2} (F_{n-1} + F_{n-2})x^n = x + \sum_{n\geq 2} F_{n-1}x^n + \sum_{n\geq 2} F_{n-2}x^n
\end{aligned}$$

Re-indexing the infinite sums, we get the following:

$$\begin{aligned}
G(x) = \sum_{n\geq 0} F_n x^n &= x + \sum_{n\geq 1} F_n x^{n+1} + \sum_{n\geq 0} F_n x^{n+2} \\
&= x + x\sum_{n\geq 0} F_n x^n + x^2 \sum_{n\geq 0} F_n x^n \\
&= x + xG(x) + x^2 G(x)
\end{aligned}$$

Note: We changed the first sum to go from $n \geq 1$ to $n \geq 0$, because $F_0 = 0$.

Hence, we get:

$$G(x) - xG(x) - x^2G(x) = x \implies G(x) = \frac{x}{1-x-x^2} \tag{2}$$

We can just use the MacLaurin series expansion of (2) to obtain the coefficients.
The Mathematica code is:

```
Series[x/(1 - x - x^2), {x, 0, 50}]
```

However, our goal is to find an explicit form, so we shall now apply Partial Fraction Decomposition on the right side of (2):

$$1 - x - x^2 = 0 \implies x = \frac{1 \pm \sqrt{5}}{2}$$

Now, for compactness, let us put $\phi_1 = \frac{1+\sqrt{5}}{2}$ and $\phi_2 = \frac{1-\sqrt{5}}{2}$.

$$\begin{aligned} 1 - x - x^2 &= -(x^2 + x - 1) = -(x - \phi_1)(x - \phi_2) \\ &= -\phi_1\phi_2 \left(\frac{1}{\phi_1}x + 1\right)\left(\frac{1}{\phi_2}x + 1\right) \end{aligned}$$

Now, $\phi_1\phi_2 = \left(\frac{1+\sqrt{5}}{2}\right)\left(\frac{1-\sqrt{5}}{2}\right) = -1$ and also, we have $\frac{1}{\phi_1} = -\phi_2$ and $\frac{1}{\phi_2} = -\phi_1$

Hence, we get the following:

$$1 - x - x^2 = -(-1)(-\phi_2 x + 1)(-\phi_1 x + 1) = (1 - \phi_1 x)(1 - \phi_2 x)$$

From here, it is easy to check that

$$\frac{x}{1 - x - x^2} = \frac{\frac{1}{\sqrt{5}}}{1 - \phi_1 x} - \frac{\frac{1}{\sqrt{5}}}{1 - \phi_2 x}$$

Hence, finally, we obtain the following:

$$G(x) = \frac{1}{\sqrt{5}}\left(\frac{1}{1 - x(\frac{1+\sqrt{5}}{2})} - \frac{1}{1 - x(\frac{1-\sqrt{5}}{2})}\right) \tag{3}$$

Then, letting $c = \frac{1}{\sqrt{5}}$, equation (3) transforms into the following:

$$G(x) = c\left(\frac{1}{1 - \phi_1 x} - \frac{1}{1 - \phi_2 x}\right)$$

$$= (c + c\phi_1 x + c\phi_1^2 x^2 + c\phi_1^3 x^3 + \dots) - (c + c\phi_2 x + c\phi_2^2 x^2 + c\phi_2^3 x^3 + \dots)$$

Thus, we have:

$$G(x) = \sum_{n \geq 0} F_n x^n = \sum_{n \geq 0} c(\phi_1^n - \phi_2^n)x^n \implies F_n = c(\phi_1^n - \phi_2^n)$$

Hence, the expression for the n^{th} Fibonacci number is:

$$F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n, \ \forall\ n \in \mathbf{N} \cup \{\mathbf{0}\}$$

3. Counting Money Change

Let us use generating functions now to tackle a more *practical* problem in real life.

The question, **What are the number of ways in which one can make change for 2000 rupees?**

Well, to get started, let us think about the number of ways we can make change for 5 rupees.
Obviously, we can only use 1 rupee, 2 rupee or 5 rupee coins.

To do so, we can take the following routes:

(1) Five 1 rupee coins 1 1 1 1 1

(2) Three 1 rupee coins and One 2 rupee coin 1 1 1 2

(3) Two 2 rupee coins and One 1 rupee coin 2 2 1

(4) One 5 rupee coin 5

Hence, there are **4 ways** to do this.

Now, we can see that coins put together side by side all add up to 5, but how do *mathematize* this?

If we consider the coins as mathematical entities, it looks like they're being multiplied, meanwhile, in reality the value they hold comes from their sum.

Is there a way we can link a mathematical product to a sum? Well, here comes that old proposition we talked about earlier, about the power of generating functions: exponentiation!

We can consider the coin values as exponents and then 'multiply' the coins and obtain the sum via exponents.

Using our new definition, let us restate what we did above. We have to obtain the amount of 5 rupees, so we need to use coins of smaller denominations. This is also implies we cannot go above the amount, for example, we cannot ever use three 2 rupee coins. Hence, we'll need, at most, two 2 rupee coins.

Hence, we can use the following number of coins:

1 rupee coins: 0, 1, 2, 3, 4, 5 $\implies$ Consider $p(x) = 1 + x + x^2 + x^3 + x^4 + x^5$
2 rupee coins: 0, 1, 2 $\implies$ Consider $g(x) = 1 + x^2 + x^4$

5 rupee coins: 0, 1 $\implies$ Consider $f(x) = 1 + x^5$

These polynomials represent the possible number of coins.

To obtain our answer, we multiply them out and consider the coefficient of x^5:

$$pgf(x) = x^{14}+x^{13}+2x^{12}+2x^{11}+3x^{10}+4x^9+3x^8+4x^7+3x^6+4x^5+3x^4+2x^3+2x^2+x+1$$

Now that we have gathered a foot hold, let us move on to the original question: evaluating the number of such combinations for 2000 rupees.

Proceeding as above, we need to find the coefficient of x^{2000} in the polynomial below:

$$P(x) = \left(\sum_{i=0}^{2000} x^i\right)\left(\sum_{i=0}^{1000} x^{2i}\right)\left(\sum_{i=0}^{400} x^{5i}\right)\left(\sum_{i=0}^{200} x^{10i}\right)\left(\sum_{i=0}^{100} x^{20i}\right)$$
$$\left(\sum_{i=0}^{40} x^{50i}\right)\left(\sum_{i=0}^{20} x^{100i}\right)\left(\sum_{i=0}^{4} x^{500i}\right)\left(\sum_{i=0}^{1} x^{2000i}\right)$$

The required coefficient, computed using a Computer Algebra System (CAS), is 14170471581.

The Mathematica code is:

```
Series[Sum[x^(i), {i, 0, 2000}]*Sum[x^(2 i), {i, 0, 1000}]*
  Sum[x^(5 i), {i, 0, 400}]*Sum[x^(10 i), {i, 0, 200}]*
  Sum[x^(20 i), {i, 0, 100}]*Sum[x^(50 i), {i, 0, 40}]*
  Sum[x^(100 i), {i, 0, 20}]*Sum[x^(500 i), {i, 0, 4}]*
  Sum[x^(2000 i), {i, 0, 1}], {x, 0, 2001}]
```

Hence, there are exactly **14170471581 ways** to make change for 2000 rupees.

The formula we used above only works for rupees 2000 and below; if we compute the whole 18000 degree polynomial, and see the coefficients for the terms after 2000, they will be inaccurate (see Supplementary Material (4)).

Thus, to obtain a formula that works for any n, we need to consider the following polynomial:

$$G(x) = \left(\sum_{i=0}^{\infty} x^i\right)\left(\sum_{i=0}^{\infty} x^{2i}\right)\left(\sum_{i=0}^{\infty} x^{5i}\right)\left(\sum_{i=0}^{\infty} x^{10i}\right)\left(\sum_{i=0}^{\infty} x^{20i}\right)$$
$$\left(\sum_{i=0}^{\infty} x^{50i}\right)\left(\sum_{i=0}^{\infty} x^{100i}\right)\left(\sum_{i=0}^{\infty} x^{500i}\right)\left(\sum_{i=0}^{\infty} x^{2000i}\right)$$

This polynomial is a product of 9 infinite geometric series!
Hence, we can write:

$$G(x) = \left(\frac{1}{1-x}\right)\left(\frac{1}{1-x^2}\right)\left(\frac{1}{1-x^5}\right)\left(\frac{1}{1-x^{10}}\right)\left(\frac{1}{1-x^{20}}\right)$$
$$\left(\frac{1}{1-x^{50}}\right)\left(\frac{1}{1-x^{100}}\right)\left(\frac{1}{1-x^{500}}\right)\left(\frac{1}{1-x^{2000}}\right)$$

One can expand this polynomial out through its Maclaurin series to obtain the coefficients up to any terms!

The Mathematica code for the same is:

```
Series[(1/(1 - x))*(1/(1 - x^2))*(1/(1 - x^5))*(1/(1 - x^10))*(1/(1 -
  x^20))*(1/(1 - x^50))*(1/(1 - x^100))*(1/(1 - x^500))*(1/(1 -
  x^2000)), {x, 0, 2001}]
```

The 2001 at the end can be replaced by any number to obtain the required coefficient.
Implicitly, the n^{th} coefficient is given by: $\frac{d^n}{dx^n}G(1)$.

4. Partitions of Positive Integers

After solving that, let us take one final problem from the branch of Partition in Number Theory: **What is the number of ways in which you can partition a natural number into positive integral parts?**
For example, the number 5 can be partitioned in seven ways, viz:

(5) $(4+1)$ $(3+2)$ $(2+2+1)$ $(3+1+1)$ $(2+1+1+1)$ $(1+1+1+1+1)$

There is a problem here: there exists no explicit formula to calculate such number of ways for arbitrary n. But, we can make a generating function and move forward from there.

A partition is determined uniquely by the number of 1s, 2s and so on, and thus we write one factor for each integer and consider coefficients in the product:

$$(1+x+x^2+x^3+\dots)(1+x^2+x^4+x^6+\dots)\dots(1+x^k+x^{2k}+x^{3k}\dots)$$

$$= \prod_{k=i}^{\infty}\sum_{i=0}^{\infty} x^{ik} = \prod_{k=1}^{\infty}\frac{1}{1-x^k}$$

We can evaluate this series' coefficients to obtain the number of partitions for any n.

The Mathematica code is:

```
Series[Product[1/(1 - x^i), {i, 10}], {x, 0, 50}]
```

Supplementary Material

(I) **Why is convergence of the infinite geometric series not of concern?**

Well, the answer is: it does not really matter.
Remember that we our using x as a dummy variable - it does not hold any significance; it is just used a mathematical entity to model the function whose properties we want to exploit.
More simply, we are only concerned with the coefficients and not the variable itself, i.e., we can take x to be any value between -1 and 1 and all our steps would be justified. In the end, it is gonna cancel out anyway.

(II) **Mathematica codes** (click on the numbers to go to the relevant section):
 (a) Generating function for the Fibonacci Sequence: 2
 (b) Number of ways to make change, accurate up to 2000 terms: 3
 (c) Number of ways to make change, accurate up to n terms: 3
 (d) Partition of Positive Integers: 4

(III) **Fibonacci Partial Fraction Decomposition:** This calculation, skipped in the presentation, is given above the third equation. Click on (3) to go there.

(IV) **Why is the first formula only valid upto 2000 terms?**
We noted above that the polynomial $P(x)$ we derived will give accurate coefficients only up till the first 2000 terms. To understand why this is true, consider the number of ways to make change for 2020 rupees.
Proceeding similarly, we know that our first polynomial would contain factors from x^1 all the way up to x^{2020}, but the first polynomial we used in the 2000 rupee case only had terms from x^1 to x^{2000}. Similar reasoning follows for the other polynomials, and hence, our $P(x)$ will undercount all the values greater than 2020.
We, thus, derived the infinite polynomial later to overcome this issue.

References

[1] Herbert S. Wilf. 2006. *Generatingfunctionology.* A. K. Peters, Ltd., USA.
[2] Paul Zeitz. 2016. *The Art and Craft of Problem Solving.* New York: Wiley.
[3] Mathologer, Explaining the bizarre pattern in making change for a googol dollars (infinite generating functions)
[4] Polya, G. (1956). *On Picture-Writing.* The American Mathematical Monthly, 63(10), 689-697.