

Generating Functions

Franklin



Outline

Overview

Counting

Binary Trees

Enumeration

Path Length



Section 1

Overview



Introduction

Suppose we have a sequence a_0, a_1, a_2, \dots

The **generating function** for this sequence is

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_kx^k + \dots ,$$

a polynomial or power series whose coefficients are the elements of the sequence.

Generating functions enable us to use manipulations to learn more about the sequence.



Examples

1, 6, 15, 20, 15, 6, 1	$1 + 6x + 15x^2 + 20x^3 + 15x^4 + 6x^5 + x^6 = (1 + x)^6$
1, 1, 1, 1, ...	$1 + x + x^2 + x^3 + \dots = \frac{1}{1 - x}$
0, 1, 1, 2, 3, 5, 8, 13, ...	$x + x^2 + 2x^3 + 3x^4 + 5x^5 + 8x^6 + 13x^7 + \dots$



Size

Often, the terms x^k of a generating function correspond to something of *size* k .

Consider the generating function from the Binomial Theorem:

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

The x^k term has coefficient $\binom{n}{k}$. Given a set with n elements, this coefficient is the number of ways to choose a subset of size k .



Section 2

Counting



Coin Tosses

Q: Flip 4 coins. How many ways are there to get 2 heads?

The answer is $\binom{4}{2} = 6$, but let's see how generating functions can be used.

Consider a single coin – it either gives us 1 tail (no heads), or 1 head.

There's only one way each of these can occur, so the generating function for a single coin is $1 \cdot x^0 + 1 \cdot x^1 = 1 + x$.

Here x represents a head – the coefficient of x^1 is the number of ways to get 1 head with a single coin. The "size" in this problem is the total number of heads – the x^k term will correspond to getting k heads.



Combining Generating Functions

Now let's combine 2 coins by multiplying their generating functions:

$$(1 + x)(1 + x) = 1 \cdot 1 + 1 \cdot x + x \cdot 1 + x \cdot x = 1 + 2x + x^2.$$

What does multiplying mean here?

- $1 \cdot 1 = x^0$ corresponds to TT (no heads)
- $1 \cdot x = x^1$ corresponds to TH (1 head)
- $x \cdot 1 = x^1$ corresponds to HT (1 head)
- $x \cdot x = x^2$ corresponds to HH (2 heads)

Each way to get k heads contributes a term of x^k to the product. So the coefficient of x^k in the product is the total number of ways to get k heads!

For 2 coins, there are 2 ways to get 1 head (coefficient of x in $1 + 2x + x^2$).



More Coins

If we have 4 coins, simply multiply the generating functions for each:

$$(1+x)(1+x)(1+x)(1+x) = (1+x)^4 = 1 + 4x + 6x^2 + 4x^3 + 1.$$

The coefficient of x^2 (6) is our answer – the number of ways to get 2 heads.

More generally, if we flip n coins, the number of ways to get k heads is the coefficient of x^k in $(1+x)^n$, which is $\binom{n}{k}$.



Changemaking

Q: How many ways can we make a dollar with (unlimited) pennies, nickels, and dimes?

The generating function for the amounts we can make in pennies is

$$1 + x + x^2 + x^3 + \cdots .$$

Here our "size" is the total money – the coefficient of x^k is the number of ways to make k cents.

For pennies, the coefficients are all 1, because we can make k cents in exactly one way: use k pennies.



Changemaking

For nickels and dimes, the generating functions are

$$1 + x^5 + x^{10} + \cdots, \quad 1 + x^{10} + x^{20} + \cdots.$$

Putting it all together, our generating function using pennies, nickels, and dimes is

$$\begin{aligned} F(x) &= (1 + x + x^2 + \cdots)(1 + x^5 + x^{10} + \cdots)(1 + x^{10} + x^{20} + \cdots) \\ &= \frac{1}{1-x} \cdot \frac{1}{1-x^5} \cdot \frac{1}{1-x^{10}}. \end{aligned}$$

The answer for making a dollar is the coefficient of x^{100} . This is nontrivial to find, but there are ways [Gas14], which we won't go into here.



Biased Coin Tosses

Q: There are n coins C_1, C_2, \dots, C_n . For each k , coin C_k is biased so that, when tossed, it has probability $\frac{1}{(2k+1)}$ of falling heads. If the n coins are tossed, what is the probability that the number of heads is odd? (Putnam 2001)

Consider coin C_1 . It has a $\frac{1}{3}$ chance of being heads, and a $\frac{2}{3}$ chance of being tails.

The "size" in this problem is the total number of heads. For coin C_1 , we can write the *probability* generating function $f_1(x) = \frac{1}{3}x + \frac{2}{3}$.

In other words, C_1 has a $\frac{1}{3}$ chance of contributing 1 to the total number of heads, and a $\frac{2}{3}$ chance of contributing 0.



Combining the Coins

The generating function for coin C_k is $f_k(x) = \frac{1}{2k+1}x + \frac{2k}{2k+1}$.

Then the probability generating function for all coins is the product of the individual generating functions for each coin:

$$F_n(x) = \left(\frac{1}{3}x + \frac{2}{3}\right) \left(\frac{1}{5}x + \frac{4}{5}\right) \cdots \left(\frac{1}{2n+1}x + \frac{2n}{2n+1}\right).$$

Given k , this product produces an x^k term for every possible way to choose k factors to take an x term from, corresponding to choosing k coins to land heads. The probabilities for each coin are multiplied, producing the probability that k specific coins land on heads.

The overall coefficient of x^k is then the sum of all these probabilities over all combinations of k heads. This is the probability of getting k heads!



The Generating Function

We have

$$\begin{aligned} F_n(x) &= \left(\frac{1}{3}x + \frac{2}{3}\right) \left(\frac{1}{5}x + \frac{4}{5}\right) \cdots \left(\frac{1}{2n+1}x + \frac{2n}{2n+1}\right) \\ &= \frac{(x+2)(x+4)\cdots(x+2n)}{3 \cdot 5 \cdots (2n+1)} \\ &= p_0 + p_1x + p_2x^2 + \cdots + p_nx^n, \end{aligned}$$

where p_k is the probability of getting k heads.

Expanding the generating function and finding each coefficient is difficult. But we actually only want the probability that the total number of heads is odd:

$$p_1 + p_3 + p_5 + \cdots .$$



The Generating Function

A generating function is still a function of a variable. So what happens if we plug things in?

If we plug in $x = 1$, we get

$$F_n(1) = p_0 + p_1 + p_2 + \cdots + p_n = \frac{(1+2)(1+4)\cdots(1+2n)}{3 \cdot 5 \cdots (2n+1)} = 1.$$

That seems useful, but we only want the sum of every other coefficient. Let's try $x = -1$:

$$\begin{aligned} F_n(-1) &= p_0 - p_1 + p_2 - \cdots \pm p_n \\ &= \frac{(-1+2)(-1+4)\cdots(-1+2n)}{3 \cdot 5 \cdots (2n-1)(2n+1)} = \frac{1}{2n+1}. \end{aligned}$$



Filtering Oddities

We have the sum and alternating sum of the coefficients. Subtracting them will cancel the terms we don't want:

$$\begin{aligned} F_n(1) - F_n(-1) &= 2p_1 + 2p_3 + 2p_5 + \cdots \\ &= 1 - \frac{1}{2n+1} = \frac{2n}{2n+1}. \end{aligned}$$

So $p_1 + p_3 + p_5 + \cdots = \frac{n}{2n+1}$. That's our answer – the probability of getting an odd number of heads upon rolling the coins C_1, \dots, C_n .



Roots of Unity Filter

Plugging in 1 and -1 is a common way to filter out every 2nd term.

This technique can be extended to filter out every k th term, by plugging in the k th roots of unity (complex solutions to $x^k = 1$).

If you're interested, I highly recommend this 3Blue1Brown video:

<https://youtu.be/bOXCLR3Wric>.

It nicely presents a solution using generating functions and the roots of unity filter to the following problem:

Find the number of subsets of $\{1, 2, 3, 4, 5, \dots, 2000\}$, the sum of whose elements is divisible by 5.



Section 3

Binary Trees



Subsection 1

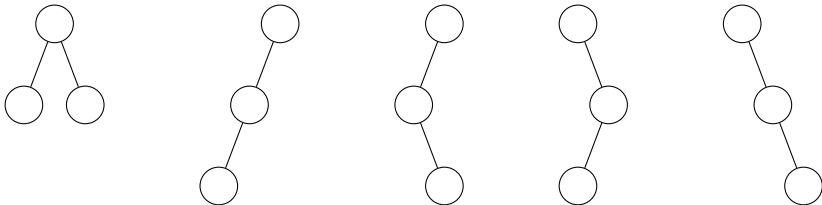
Enumeration



Enumerating Binary Trees

Q: How many different binary trees have n nodes? [SF96]

- For $n = 0$, there is 1 tree – the empty tree
- For $n = 1$, there is 1 tree – a single node
- For $n = 2$, there are 2 trees – a node with a left child, or a node with a right child
- For $n = 3$, there are 5 trees:



The Generating Function

Let \mathcal{T} be the set of all binary trees. Given a tree $t \in \mathcal{T}$, let $|t|$ be the number of nodes in t . We want the number of trees in \mathcal{T} with n nodes (trees of "size" n).

Consider the generating function

$$T(x) = \sum_{t \in \mathcal{T}} x^{|t|}.$$

In other words, each tree contributes a single term to the sum, which is determined by the number of nodes in that tree.

For example, the 3-node trees on the previous slide would each contribute an x^3 term to $T(x)$. Since there are exactly 5 trees with 3 nodes, the coefficient of x^3 in $T(x)$ is $5x^3$.



The Generating Function

So we have the generating function

$$T(x) = \sum_{t \in \mathcal{T}} x^{|t|} = 1 + x + 2x^2 + 5x^3 + \cdots = \sum_{n \geq 0} T_n x^n.$$

The coefficients T_n represent the number of binary trees with n nodes. We want to find an expression for T_n .

To use this generating function, we exploit the recursive structure of binary trees.



Recursive Structure

A binary tree t is either:

- Empty with 0 nodes, or
- A single node with a left subtree t_L and a right subtree t_R (binary trees) – with $1 + |t_L| + |t_R|$ nodes total
 - ▶ In this case, each tree t uniquely corresponds to a pair of trees (t_L, t_R)

Using this, we can write our generating function as follows:

$$T(x) = \sum_{t \in \mathcal{T}} x^{|t|} = 1 + \sum_{t_L \in \mathcal{T}} \sum_{t_R \in \mathcal{T}} x^{|t_L| + |t_R| + 1}$$



A Functional Equation

Then,

$$\begin{aligned}T(x) &= 1 + \sum_{t_L \in \mathcal{T}} \sum_{t_R \in \mathcal{T}} x^{|t_L|+|t_R|+1} \\&= 1 + x \sum_{t_L \in \mathcal{T}} x^{|t_L|} \sum_{t_R \in \mathcal{T}} x^{|t_R|} \\&= 1 + xT(x)^2.\end{aligned}$$

Solving for $T(x)$ using the quadratic formula,

$$T(x) = \frac{1 \pm \sqrt{1 - 4x}}{2x}.$$

Since $T(0) = 1$, we take the minus sign.



Binomial Expansion

We need to expand our expression for $T(x)$ to extract coefficients.

According to an extended form of the binomial theorem,

$$(1 + y)^{1/2} = 1 + \binom{1/2}{1}y + \binom{1/2}{2}y^2 + \binom{1/2}{3}y^3 + \cdots = \sum_{k=0}^{\infty} \binom{1/2}{k}y^k.$$

where we can generalize the definition of $\binom{n}{k}$ to real n as follows:

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



Extracting Coefficients

Using this expansion with $y = -4x$,

$$\begin{aligned} T(x) &= \frac{1 - \sqrt{1 - 4x}}{2x} \\ &= \frac{1 - \left(1 + \binom{1/2}{1}(-4x) + \binom{1/2}{2}(-4x)^2 + \binom{1/2}{3}(-4x)^3 + \dots\right)}{2x} \end{aligned}$$

After some simplification, we find that the coefficient of x^n is

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

This is the number of binary trees with n nodes! Note: The numbers T_n are called the *Catalan numbers*.



Subsection 2

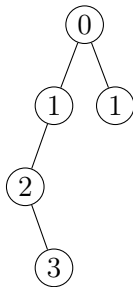
Path Length



Path Length

Given a binary tree t , its **path length** $\pi(t)$ is the sum of the lengths of the paths from the root to each node in the tree.

For example, this binary tree has path length $0 + 1 + 1 + 2 + 3 = 7$.



If we divide the path length by the number of nodes, we get the average distance from the root to a node in the tree. This is useful for analyzing algorithms that involve searching for nodes in a tree.



Path Length in Random Trees

Q: What is the expected path length of a random binary tree with n nodes? [SF96]

Here, "random" means that each of the T_n binary trees with n nodes has an equal probability of being selected (*random binary Catalan trees*).

- This is often used in compiler design for parse trees for expression evaluation. However, other models of random trees exist.

In other words, over all binary trees with n nodes, what is the average path length?

We'll answer this question using a similar approach to enumerating binary trees.



The Generating Function

Recall that \mathcal{T} is the set of all binary trees.

Define the *cumulative* generating function

$$C_T(x) = \sum_{t \in \mathcal{T}} \pi(t) x^{|t|} = \sum_{n \geq 0} C_n x^n.$$

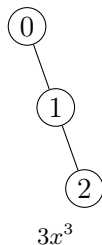
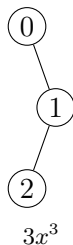
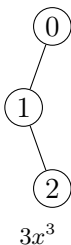
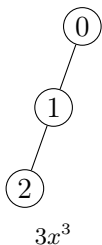
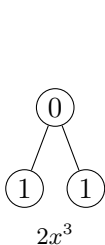
In the middle is the generating function written summing tree-by-tree.
On the right, all the like terms are collected, and we sum over each term.

The coefficients C_n tell us the total path length over all trees with n nodes.



Example

Here are the five 3-node trees from earlier, and their contribution to $C_T(x)$:

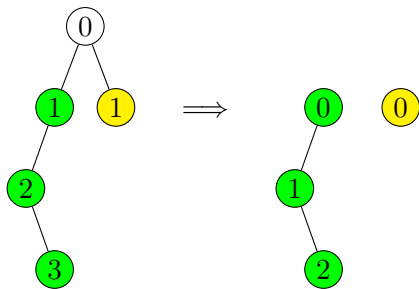


The coefficient corresponds to path length; the exponent of x corresponds to the number of nodes.



Recursive Definition of Path Length

Consider the tree t from earlier; its subtrees t_L and t_R are colored.



When t_L is split off from t , the "new root" is one level lower.

So the depths of the nodes in t_L are one less than the depths of the same nodes of in t .

The same is true for t_R .

So $\pi(t) = 0 + \pi(t_L) + |t_L| + \pi(t_R) + |t_R|$.



Using Recursion

Now,

$$\begin{aligned} C_T(x) &= \sum_{t \in \mathcal{T}} \pi(t) x^{|t|} \\ &= \sum_{t_L \in \mathcal{T}} \sum_{t_R \in \mathcal{T}} (\pi(t_L) + \pi(t_R) + |t_L| + |t_R|) x^{|t_L| + |t_R| + 1} \end{aligned}$$

Note that there's no constant term ($1 + \dots$) since the only tree with 0 nodes (the empty tree) has path length 0, so the coefficient of x^0 is 0.

This expands into 4 parts. The first is

$$\sum_{t_L \in \mathcal{T}} \sum_{t_R \in \mathcal{T}} \pi(t_L) x^{|t_L| + |t_R| + 1} = x \sum_{t_L \in \mathcal{T}} \pi(t_L) x^{|t_L|} \sum_{t_R \in \mathcal{T}} x^{|t_R|} = x C_T(x) T(x).$$



Expansion

The second part is similarly

$$\sum_{t_L \in \mathcal{T}} \sum_{t_R \in \mathcal{T}} \pi(t_R) x^{|t_L|+|t_R|+1} = x \sum_{t_L \in \mathcal{T}} x^{|t_L|} \sum_{t_R \in \mathcal{T}} \pi(t_R) x^{|t_R|} = xT(x)C_T(x).$$

The third part is (recall $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$, so $T'(x) = \sum_{t \in \mathcal{T}} |t| x^{|t|-1}$)

$$\sum_{t_L \in \mathcal{T}} \sum_{t_R \in \mathcal{T}} |t_L| x^{|t_L|+|t_R|+1} = x^2 \sum_{t_L \in \mathcal{T}} |t_L| x^{|t_L|-1} \sum_{t_R \in \mathcal{T}} x^{|t_R|} = x^2 T'(x) T(x).$$

Similarly, the fourth part is

$$\sum_{t_L \in \mathcal{T}} \sum_{t_R \in \mathcal{T}} |t_R| x^{|t_L|+|t_R|+1} = x^2 \sum_{t_L \in \mathcal{T}} x^{|t_L|} \sum_{t_R \in \mathcal{T}} |t_R| x^{|t_R|-1} = x^2 T(x) T'(x).$$



The Functional Equation

Putting it all together,

$$\begin{aligned}C_T(x) &= \sum_{t \in \mathcal{T}} \pi(t) x^{|t|} \\&= \sum_{t_L \in \mathcal{T}} \sum_{t_R \in \mathcal{T}} (\pi(t_L) + \pi(t_R) + |t_L| + |t_R|) x^{|t_L| + |t_R| + 1} \\&= 2xC_T(x)T(x) + 2x^2T(x)T'(x).\end{aligned}$$

Earlier we found that $T(x) = \frac{1 - \sqrt{1 - 4x}}{2x}$. Solving for $C_T(x)$,

$$C_T(x) = \frac{2x^2T(x)T'(x)}{1 - 2xT(x)} = \frac{1}{x} \left(\frac{x}{1 - 4x} - \frac{1 - x}{\sqrt{1 - 4x}} + 1 \right).$$



Coefficients

The coefficient of x^n is the total path length over all binary trees with n nodes.

We want the average path length, so we divide the coefficient by the number of trees with n nodes: T_n .

After some messy expansions, the average path length comes out as

$$\frac{(n+1)4^n}{\binom{2n}{n}} - 3n - 1 = n\sqrt{\pi n} - 3n + O(\sqrt{n}).$$



Other Applications of Generating Functions

- General trees, permutations, strings/regular expressions, many other combinatorial structures for analyzing algorithms
- Symbolic method (analytic combinatorics)
 - ▶ To get the generating function $T(x)$ for binary trees, we can actually just write this, since a tree is either empty or a node and two subtrees:

$$\mathcal{T} = \{\square\} + \{\bullet\} \times \mathcal{T} \times \mathcal{T} \implies T(x) = 1 + xT(x)^2.$$

- Snake Oil method: evaluate terrible sums like $f_n = \sum_k \binom{n+k}{2k} 2^{n-k}$ by throwing them in as coefficients of a generating function and swapping the order of summation
- Solving recurrences (e.g. finding a formula for Fibonacci numbers)



Questions?



A generating function is a device somewhat similar to a bag. Instead of carrying many little objects detachedly, which could be embarrassing, we put them all in a bag, and then we have only one object to carry, the bag.

— George Polya ([1954](#))



Bibliography I



Philippe Flajolet and Robert Sedgewick.

Analytic Combinatorics.

Cambridge University Press, USA, 1 edition, 2009.



William Gasarch.

How many ways can you make change: Some easy proofs, 2014.



Michel Goemans.

Generating functions.

<https://math.mit.edu/~goemans/18310S15/generating-function-notes.pdf>, 2015.



Robert Sedgewick and Philippe Flajolet.

An introduction to the analysis of algorithms.

Addison-Wesley Longman Publishing Co., Inc., USA, 1996.



Bibliography II



Herbert S. Wilf.

Generatingfunctionology.

A. K. Peters, Ltd., USA, 2006.

