Ten Combinatorial Commandments [Draft]

Shaoshi Chen   Robert Dougherty-Bliss
Doron Zeilberger

July 13, 2024

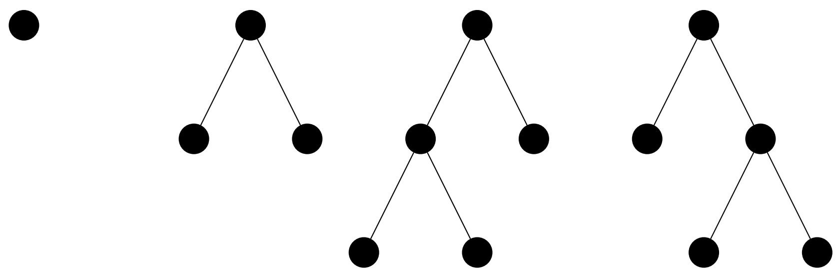
The *Catalan numbers*

$$C_n := \frac{1}{n+1}\binom{2n}{n}$$

are the most ubiquitous sequence in all of combinatorics. Their OEIS entry A108 contains over 10,000 words, and Stanley's famous *Enumerative Combinatorics* lists (as an exercise!) XXX combinatorial objects counted by the Catalan numbers =0mu plus 3mu[**stanley**].

Here we emulate both Stanley and another great mathematician, Carl Jacobi, who was known to say that one must always "seek a converse, turn a thought the other end to" =0mu plus 3mu[**invert**]. Rather than providing a collection of theorems about just one sequence, we will provide a collection of proofs about just one theorem. We hope that newcomers or novices to enumerative combinatorics will appreciate the many techniques and tricks laid out here over many proofs.

Our lone theorem is about *binary trees*. A binary tree is a tree where every node contains at most two children, one "left" and one "right." Binary trees can be organized in a variety of ways, but one popular way is to group them by the number of "final" nodes without descendants they contain, called *leaves*. Below are four unlabeled trees which contain one to three leaves.



In fact, these are the *only* such nonempty binary trees if we require that each node has exactly zero or one child. Such a binary tree is called "full." Our theorem is as follows.

Theorem 1. *The number B_n of full binary trees with $n+1$ leaves is C_n.*

1 Remy's bijective proof

Consider an arbitrary full binary tree. Pick any leaf. Delete this leaf, and replace its parent with the sibling of the leaf we picked. This is a bijection between two sets.

On the left, we have "binary tree with one leaf singled out." There are $(n+1)B_n$ of these. (Or whatever the correct indexing should be.) On the right, we have "binary tree with one leaf / internal node singled out, plus with a marker for whether this came from the 'right' or 'left' in our bijection." There are $2(2n+1)B_{n-1}$ of these. That gives the recurrence

$$(n+1)B_n = 2(2n+1)B_{n-1}.$$

(Check the indices here! They are almost definitely wrong right now.)

2 Young Tableaux

There's an easy bijection between Dyck paths and $2n$ Young Tableaux. Then, the hook length formula lets you count the number of these Young Tableaux. (This is a bit reversed from what you would normally do.)

3 Generating functions

Every full binary tree is either empty or contains a single node with two full binary trees as children. (Note that this include the case where the tree is just one node.) If the children of the root node are (L, R), then each distinct pair produces a different tree, and the number of leaves of the entire tree is the number in L plus the number in R. So, to get a tree with $n > 1$ leaves, we stitch together all possible left-right children such that the number of their leaves sums to n.

This amounts to the following proposition.

Proposition 1. *For $n \geq 0$,*

$$B_{n+1} = \sum_{k=1}^{n} B_k B_{n+1-k}.$$

4 Combinatorial grammars and generating functions

Linguists and computer scientists have developed methods to formally describe languages. Combinatorialists have recognized that many discrete objects can also be described by these methods, and in fact can lead directly to solutions to difficult enumeration problems. This approach is epitomized in Flajolet and Sedgewick's *Analytic Combinatorics* =0mu plus 3mu[**analytic**].

Consider a language which consists of words of the form a, aa, aaa, $aaaa$, and so on, as well as the empty word. This language is generated by the *context free grammar*

$$\alpha \to \epsilon \mid a\alpha.$$

With the understanding that α represents an arbitrary member of our language, and that ϵ is a symbol for the empty word, this states that each α is either ϵ or a followed by another occurrence of some α. The language of balanced parenthesis could be represented as

$$\alpha \to () \mid (\alpha) \mid \alpha\alpha,$$

which contains the words $()$, $(())()$, and $(())$.

This idea translates quite directly to full binary trees. A full binary tree is either empty, contains a root and no children, or contains a root and has exactly two children. If T stands for an arbitrary full binary tree and

r an arbitrary root, then the above description translates to the following grammar:

$$T \to r \mid r(T)(T)$$

where r stands for a root node. This says that a tree is either empty, or contains a node and two children.

The main benefit to the grammatical approach is their direct relation to generating functions.

XXX

$$f(x) = 1 + xf(x)^2$$

5 Algebraic equations imply recurrences

A generating function which satisfies a polynomial equation is called *algebraic*. Every algebraic generating function leads directly to a certain kind of recurrence.

[Explain how to go from algebraic equations to differential equations to recurrences. Mention gfun!]

6 Lagrange Inversion

The generating function equation satisfied by $f(x)$ is of a very special form. That is, it is of the form

$$u(x) = x\phi(u(x))$$

for some formal power series $\phi(x)$.

Reason as follows:

$$f(x) = 1 + xf(x)^2$$
$$f(x) - 1 = x((f(x) - 1) + 1)^2.$$

Now set $u(x) = f(x) - 1$, so that we have

$$u(x) = x(1 + u(x))^2.$$

And so...

Then apply the formula $[x^n]u(x) = \frac{1}{n}[x^{n-1}]\phi(x)^n$.

Easy Lagrange reference: https://sites.math.rutgers.edu/~zeilberg/mamarim/mamarimPD

7 Elementary calculus

Easy from the binomial theorem:

$$\sum_n \binom{2n}{n} x^n = \frac{1}{\sqrt{1-4x}}.$$

Then integrate both sides and divide by x:

$$\sum_n \frac{1}{n+1}\binom{2n}{n} x^{n+1} = \int \frac{1}{\sqrt{1-4x}}\, dx.$$

Can check that the difference is constant.

8 Dyck words

The grammar leads directly to the definition of Dyck paths. We can also mention the more direct bijection between binary trees and Dyck paths, which is a description of the (prefix?) depth-first traversal of a tree.

From here, we need to count the number of Dyck paths of a certain length. But we want to do so without reference to the above methods, such as solving the generating function equation, using Lagrange inversion, and so on.

[This may become multiple sections.]

Obvious recurrence method Think of Dyck paths as walks from $(0,0)$ to (n,n) which stay below the identity line and consist of steps right and up. We will first give a more general formulation, where $F(a,b)$ the number of such walks from $(0,0)$ to (a,b). This has a very obvious description:

$$\begin{aligned} F(a,b) &= F(a-1,b) + F(a,b-1) \\ F(a,0) &= 1 \\ F(a,a+1) &= 0. \end{aligned}$$

From here, we can verify the closed form answer $F(a,b) = (a-b+1)(a+b)!/(a+1)!b!$ quite easily. But we should also mention how you come up with this. (One idea is to guess that the answer is hypergeometric, then fit a formula given the data.)

Reflection proof The number of all walks from $(0,0)$ to (n,n) is exactly $\binom{2n}{n}$. There is a fairly standard proof that you can "reflect" bad paths to get good ones, and vice versa.

Motzkin lemma Take dyck words of semilength n and consider the $2n+1$ cyclic shifts that they generate. These are, taken together, just the set of $2n+1$ lists with $n+1$ copies of 1 and n copies of -1, of which there are $\binom{2n+1}{n}$. Motzkin's lemma says that these cyclic shifts are all distinct, and it should be easy to show that they are all distinct. So, the number of dyck words of semilength n is the number of equivalence classes, which is $\binom{2n+1}{n}\frac{1}{2n+1}$.

9

10 Summation methods

As proven in previous sections, the number of full binary trees with n leaves satisfies the identity

$$B_{n+1} = \sum_{k=1}^{n} B_k B_{n-k}.$$

Previously, we continued from here by deriving the generating function of B_n. That is, in some sense, the more difficult choice. A simpler method is to just check that C_n satisfies the same recurrence. If so, and if they have the same initial conditions, then we would be done. In other words, we would be done if we could prove the following proposition.

Proposition 2.

$$\frac{1}{n+2}\binom{2(n+1)}{n+1} = \sum_{k=1}^{n} \frac{1}{(k+1)(n-k+1)}\binom{2k}{k}\binom{2(n-k)}{n-k}.$$

Without leaving the world of summation, there are roughly two ways to prove this identity. The first, following the summation bible *Concrete Mathematics* =0mu plus 3mu[**conc**], would be to manipulate the sum and apply various transformations until a reasonable identity applied. The second is to pull a rabbit out of a hat.

Proof. Call the summand $F(n,k)$ and define the function

$$G(n,k) := \frac{(n+1-2k)(2k-2n-1)(k+1)}{n^2+3n+2} F(n,k).$$

When defined, these functions satisfy the identity

$$G(n,k+1) - G(n,k) = F(n,k),$$

which can (and should!) be verified by a computer. It follows that

$$\sum_{k=1}^{n-1} F(n,k) = G(n,n) - G(n,0) = \frac{3n}{(n+1)(n+2)} \binom{2n}{n}.$$

Adding $F(n,n)$ to both sides yields

$$\sum_{k=1}^{n} F(n,k) = \frac{2(n+2)}{(n+1)(n+2)} \binom{2n}{n}.$$

The expression on the right is C_{n+1}. $\square$

The function $G(n,k)$ from the above proof appeared as if by magic. It is the result of *Gosper's algorithm* for hypergeometric summation =0mu plus 3mu[**gosper**]. A sequence $a(n)$ is *hypergeometric in n* provided that $a(n+1)/a(n)$ is a fixed rational function in n. Given a hypergeometric sequence $a(n)$, Gosper's algorithm determines whether there exists a hypergeometric $s(n)$ such that

$$s(n+1) - s(n) = a(n),$$

and constructs it if it exists. This is equivalent to asking whether $\sum_{k=1}^{n} a(k)$ is hypergeometric, which in very loose terms is like asking whether this sum has a "closed form."

Our summand $F(n,k)$ happens to be hypergeometric in k (where the base field consists of all rational functions in n). When we execute Gosper's algorithm on $F(n,k)$ we get precisely the $G(n,k)$ referenced above.

It is worth mentioning that we got lucky. Gosper's amazing algorithm does not always produce a closed form, because not every indefinite sum has a closed form. This is especially true for hypergeometric sequences of more than one variable. There are very powerful generalizations of Gosper's algorithm that produce *recurrences* for all hypergeometric sums, and indeed much more general classes of sums. We will not go into detail here, but see =0mu plus 3mu[**ab**; **dfinite**; **computer**] and the innumerable references there.

[Todo: More on Zeilberger, WZ, companion identities? -R]

11 Constant term integrals