Notes on exponential generating functions (continued).

9. The exponential generating function for rooted labelled trees

In this section we consider the problem of enumerating unordered rooted trees on a set of n labelled vertices. This is a typical structure enumeration problem which we can attack using exponential generating functions. We have already seen how to enumerate binary trees using the product principle. For unordered trees we typically need the composition principle.

Let $t(n)$ denote the number of unordered rooted labeled trees on an n-element set, and let

$$T(x) = \sum_{n=0}^{\infty} t(n) \frac{x^n}{n!}$$

be the corresponding exponential generating function. It will turn out that things work best if we agree that $t(0) = 0$, that is, we do not count the empty tree as a rooted tree.

We begin as we did for binary trees, by viewing a tree as a product structure, in which one vertex is chosen to be the root, and the rest are arranged into a collection of subtrees whose roots are the children of the main root. In other words, the vertices other than the one chosen as the root are given the structure of a "forest" of rooted trees. Next we observe that a forest is really a composite structure, consisting of a partition of the vertices, with a structure of rooted tree on the vertices in each block, and a trivial structure on the set of blocks.

By the composition principle, the exponential generating function enumerating forests is equal to

$$e^{T(x)}.$$

Note that our agreement not to count the empty tree is convenient here, since it makes $T(0) = 0$, as we require when applying the composition principle. Now the product structure describing a root and a forest is (one-element set)$\times$(forest), so by the product principle we have

$$T(x) = xe^{T(x)}.$$

This identity determines $T(x)$. To actually compute any fixed number of terms, one can use the method of successive approximation. Begin with

$$T(x) = x + \cdots,$$

which is correct through the order x term. Substituting this into the right hand side in the above identity, we will get $e^{T(x)}$ correct through the order x term, and hence we will get $xe^{T(x)}$ correct though the order x^2 term. In this way we find

$$T(x) = xe^{(x+\cdots)} = x(1 + x + \cdots) = x + x^2 + \cdots.$$

Repeating this, we get

$$\begin{aligned} T(x) &= xe^{(x+x^2+\cdots)} = xe^x e^{x^2} \cdots \\ &= x(1 + x + x^2/2 + \cdots)(1 + x^2 + \cdots) \\ &= x(1 + x + 3x^2/2 + \cdots) \\ &= x + x^2 + 3x^3/2 + \cdots. \end{aligned}$$

This can be continued indefinitely, the next step giving

$$T(x) = x + 2x^2/2! + 9x^3/3! + 64x^4/4! + \cdots.$$

This agrees with what we knew from Cayley's tree enumerator or the matrix-tree theorem to be the answer, namely $t(n) = n^{n-1}$. Note however that the exponential generating function approach allows us to arrive at the identity $T(x) = xe^{T(x)}$ in a direct fashion, whereas to get the Cayley or matrix-tree generating functions, we first needed a kind of amazing guess to discover the answer, and then a tricky argument to prove it.

10. Lagrange inversion

From the computation of $T(x)$ above we might readily guess the formula $t(n) = n^{n-1}$ if we hadn't known it before, but it is still not apparent why the formal power series

$$T(x) = \sum_{n=1}^{\infty} n^{n-1} \frac{x^n}{n!}$$

should be the solution of the equation

$$T(x) = xe^{T(x)}.$$

Here I will briefly discuss one way in which this result can be obtained. We can rewrite the identity for $T(x)$ as

$$T(x)e^{-T(x)} = x,$$

which says that $T(x)$ is the *inverse function* of xe^{-x}, in the same way that $\arcsin x$ is the inverse function of $\sin x$, or e^x is the inverse function of $\log x$. There is a classical formula of Lagrange to find the coefficients of the Taylor series of an inverse function. Since we are concerned here with formal series, we only allow series with zero constant term, as these are the only formal series for which it makes sense to speak of a formal inverse function. Then the *inversion formula of Lagrange* can be written as follows.

Theorem 1. *Let $xG(x)$ be the functional composition inverse of $xF(x)$. Then*

$$[x^n]G(x) = [x^n]\frac{F(x)^{-n-1}}{n+1},$$

where the symbol $[x^n]$ denotes the coefficient of x^n in the expression that follows it.

Note that the formula gives the x^n coefficient of $G(x)$ as the x^n coefficient of another series which *depends on n*. It does not give a closed form for $G(x)$, which would be impossible in general.

At this point, we will not discuss the proof of Lagrange's formula, but take it for granted and apply it in our situation. Since $T(x)$ is the functional composition inverse of xe^{-x}, we take $F(x) = e^{-x}$. Then $G(x) = T(x)/x$, so the coefficient of x^n in $G(x)$ is actually the coefficient of x^{n+1} in $T(x)$, which is $t(n+1)/(n+1)!$. According to the formula, this is given by

$$t(n+1)/(n+1)! = [x^n]\frac{e^{(n+1)x}}{n+1} = \frac{(n+1)^n}{n!(n+1)}.$$

Hence

$$t(n+1) = (n+1)^n,$$

or, replacing n with $n-1$,

$$t(n) = n^{n-1}.$$

11. Variations on tree enumeration

We can use the method of the previous section to count rooted labeled trees with various kinds of additional structure on the children of each vertex. We will always view a tree as a product structure (one-element root)$\times$(forest), and a forest as a composite structure. In general the outer structure on the trees in the forest may be non-trivial, depending on what type of trees we want to count.

Example: Unordered rooted labeled binary trees. These are unordered trees in which every vertex has at most two children. Thus each forest is to be a forest of at most two trees. The generating function for the trivial structure of "set with at most two elements" is $1 + x + x^2/2$, so

$$U(x) = x(1 + U(x) + U(x)^2/2),$$

where $U(x)$ is the generating function for unordered rooted labeled binary trees. Note that this is a quadratic equation which can be solved exactly for $U(x)$.

Example: Unordered rooted labeled strictly binary trees. This means each vertex has exactly two children or none. For the generating function we get

$$V(x) = x(1 + V(x)^2/2).$$

Example (just to show the lengths to which you can take this method): Unordered rooted labeled trees in which every vertex either has only leaves or no leaves as children. As before, we analyze such a tree as a product structure consisting of a root and a forest. The forest structure for this one is a bit tricky. Either the forest is all leaves, that is, it is a forest of one-vertex trees, or else it is a forest of trees of our same type again, all of which have at least two vertices. Let $Z(x)$ be the exponential generating function enumerating our trees. By definition we don't count the empty tree, and there is one tree on a one-element set, contributing a term x to $Z(x)$. The remaining terms of $Z(x)$ enumerate the trees with more than one element, so their generating function is $Z(x) - x$. By the addition principle and the composition principle, the generating function enumerating forests of the type we want is $e^x + e^{Z(x)-x}$. Here the term e^x enumerates the forests consisting of one-vertex trees (a forest of one-vertex trees is just a trivial structure) and the other term enumerates the forests of trees with more than one vertex. The identity giving $Z(x)$ is then

$$Z(x) = x(e^x + e^{Z(x)-x}).$$

12. Trees, permutations, and functional digraphs

Let A be a finite set and $f\colon A \to A$ be any function from A to itself—not necessarily a permutation. We can form a directed graph whose vertices are the elements of A, with an edge directed from x to $f(x)$ for each element $x \in A$. This digraph will have the property that every vertex has out-degree equal to 1. Conversely, any such digraph is the graph of a unique function f, namely the function mapping each vertex x to the vertex at the other end of the unique edge directed out of x. Note that loops are allowed, since we may have elements with $f(x) = x$. A digraph in which every vertex has out-degree 1 is called a *functional digraph*.

In particular, if A has n elements, then there are n^n functions from A to itself, and hence n^n functional digraphs with vertex set A. If we regard these as structures on A, then they are enumerated by the exponential generating function

$$F(x) = \sum_n n^n \frac{x^n}{n!}.$$

This suggests a possible way of proving that the number of unordered rooted labelled trees on n vertices is given by $t(n) = n^{n-1}$, purely combinatorially, without using the Lagrange inversion formula. Namely, the identity

$$T(x) = \sum_{n=1}^{\infty} n^{n-1} \frac{x^n}{n!},$$

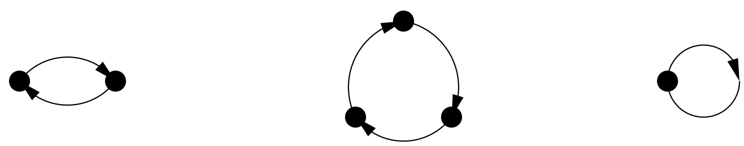
which we would like to prove, is equivalent to

$$xT'(x) = \sum_{n=1}^{\infty} n^n \frac{x^n}{n!} = F(x) - 1. \tag{1}$$

Here we subtracted 1 on the right hand side so as not to count the empty functional digraph. We must do this because the constant term on the left-hand side is zero.

We will use exponential generating function principles to find an identity satisfied by $F(x)$, and use this to prove that $F(x) = xT'(x)$.

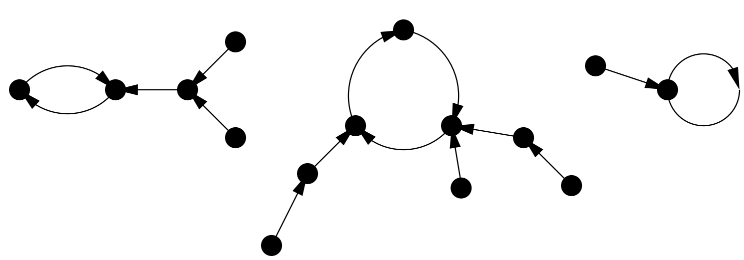
To get started, let us consider two special types of functional digraphs. The first type is the digraph of a permutation. Here the graph simply displays the cycle structure of the permutation, with the edges directed around each cycle, as indicated in the example shown here.



The second special type of functional digraph to consider is a rooted tree, with a loop on the root, and the rest of the edges directed toward the root. Then f is the function mapping each non-root vertex in the tree to its parent, and the root to itself.

Trees and permutations represent extreme cases of functional digraphs. At one extreme, in a permutation digraph, every vertex belongs to a cycle. At the other, in a tree, only one vertex, the root, belongs to a cycle. The general functional digraph can be described as a mix of permutations and trees. Suppose we start at a vertex x and follow directed edges from x to $f(x)$ to $f^2(x) = f(f(x))$ and so on. Eventually, since our vertex set is finite, we must return to a vertex already visited. At that point we fall into a cycle and we will continue around and around it forever. Consequently we can say this about the structure of a functional digraph: some vertices belong to cycles, and the rest of the vertices have paths leading to these vertices.

Now let x be a vertex in a cycle and consider the set X of all vertices y such that y does not belong to any cycle, and $f^k(y) = x$ for some k. Together with x, the vertices in X form a tree directed into x as the root, since in the graph on $X \cup \{x\}$, every path leads eventually to x. Now every vertex not belonging to a cycle belongs to one such tree for some vertex x that does belong to a cycle. In this way, given a functional digraph on a set A, we get a partition of A into rooted trees, and an arrangement of the roots of the trees into a bunch of cycles, that is, into a permutation. Here is a picture typical of the situation.



In this functional digraph, there are six trees, and the permutation on their roots is the one shown in the previous figure. Note that two of the trees in this example are one-element trees consisting of a root only.

What we have described is an equivalence of the structure "functional digraph" with the composite structure of (permutation)$\circ$(rooted tree). Indeed, a directed rooted tree with all edges directed into the root is the same as an undirected rooted tree, since the tree itself determines the directions, and the structure of permutation that we have on the set of roots we can equally well think of as being on the set of trees themselves, with each tree represented by its root. The generating function for permutations is $\frac{1}{1-x} = 1/(1-x)$, and that for rooted trees is what we have denoted $T(x)$. Hence the generating function $F(x)$ for functional digraphs is related to $T(x)$ by

$$F(x) = 1/(1 - T(x)).$$

To

To prove (1) we have now only do to a little bit of calculus and algebra. Our starting point is the two identities

$$T(x) = xe^{T(x)}, \quad F(x) = 1/(1 - T(x)),$$

both of which we obtained directly from exponential generating function counting principles. Differentiating both sides of the first identity gives

$$T'(x) = e^{T(x)} + xe^{T(x)}T'(x),$$

and hence

$$xT'(x) = xe^{T(x)} + x(xe^{T(x)})T'(x).$$

Using $xe^{T(x)} = T(x)$, this simplifies to

$$xT'(X) = T(x) + xT(x)T'(x),$$

or

$$(1 - T(x))xT'(x) = T(x).$$

Therefore

$$xT'(x) = T(x)/(1 - T(x)) = 1/(1 - T(x)) - 1 = F(x) - 1,$$

which is what we wanted to show.