Notes on cycle generating functions

1. The cycle generating function of a species

A *combinatorial species* is a rule attaching to each finite set X a set of structures on X, in such a way that the allowed structures do not depend on the particular names of the elements of X. This concept is best illustrated by giving examples of structures that do and do not form species.

In the notes on exponential generating functions, one kind of structure we enumerated was that of an *alternating permutation* on the numbers $1, 2, \ldots, n$. Recall that this meant a permutation $a_1, a_2 \ldots, a_n$ of these numbers such that $a_1 < a_2 > a_3 < a_4 \cdots$. These turned out to be interesting structures, because the exponential generating function for odd-length alternating permuations is $\tan x$, and the exponential generating function for even-length ones is $\sec x$. However, these structures do NOT belong to a species. The reason is that the condition $a_1 < a_2 > a_3 \cdots$ depends on the relation "$<$," which is specific to numbers. One way to express this difficulty is that there is no meaning to the concept of an alternating permutation of a set X whose elements have no built-in order. Another, somewhat more precise, way to formulate the trouble is that it is not possible to permute the numbers in an alternating permutation and still have an alternating permutation.

For constrast, consider the structure of *labelled tree* on a vertex set X. The concept of a labelled tree on any vertex set whatsoever is meaningful, because it does not depend on the particular names of the vertices. Again, we can formulate this more precisely by saying that you can permute the vertices of a labelled tree and get another (maybe the same) labelled tree. Labelled trees DO form a species.

Other examples of species are the species of all linear orderings of a set X, or the species of all permutations of a set X, or the trivial species which has just one structure on each set X. As we shall see, the linear orderings of X and the permuations of X are two *different* species, even though each has the same number of structures on an n element set, namely $n!$, and even though the two species therefore have the same exponential generating function. In fact, all of the examples of structures that we have enumerated using exponential generating functions are species, with the sole exception of alternating permutations.

If F is a species, we will denote the set of F-structures on a set X by $F(X)$. Because F is a species, the group of permutations of X acts on $F(X)$. In particular, the symmetric group S_n acts on $F([n])$. Here, as usual, $[n]$ is an abbreviation for $\{1, 2, \ldots, n\}$.

Definition. The *cycle generating function* of a species F is the formal power series in a variable x and infinitely many variables $p_1, p_2, \ldots$ defined by the formula

$$Z_F(p_1, p_2, \ldots; x) = \sum_{n=0}^{\infty} \frac{x^n}{n!} \sum_{g \in S_n} |F([n])^g| p_1^{j_1(g)} p_2^{j_2(g)} \cdots p_n^{j_n(g)}.$$

Here $F([n])^g$ denotes the set of F-structures on $[n]$ which are fixed by the permutation g, and $j_k(g)$ is the number of k-cycles of g (acting on $[n]$).

To make things clear, let's work out the first few terms of Z_F for the species F of labelled, unrooted trees. For each n and each permutation $g \in S_n$, we need to figure out how many trees are fixed by g. It will be enough to do this for one permutation of each cycle type, since they all fix the same number of trees.

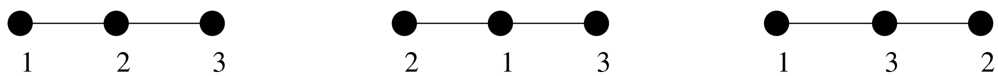
For $n = 0$, there is just one tree, and just one g, namely the identity permutation of the empty set, which has no cycles. The corresponding term in Z_F is 1.

For $n = 1$, there is again just one tree and one permutation, the identity, but now it has a single cycle of length 1, so its monomial is p_1. The corresponding term in Z_F is xp_1.

For $n = 2$, there is still just one tree. Now there are two permutations, the identity and (1 2), and of course they both fix the one tree. The corresponding term in Z_F is

$$\frac{x^2}{2}(p_1^2 + p_2).$$

For $n = 3$, there are three trees, shown here.



The identity element in S_3 obviously fixes all of them, contributing $3p_1^3$. The element $g = (1\ 2)$ fixes one tree, the last one shown above, contributing a monomial p_1p_2. There are two more transpositions, (1 3) and (2 3), contributing a total of $3p_1p_2$. The 3-cycles (1 2 3) and (1 3 2) do not fix any trees, so they contribute nothing. The resulting third term in Z_F is

$$\frac{x^3}{6}(3p_1^3 + 3p_1p_2) = \frac{x^3}{2}(p_1^3 + p_1p_2).$$

Before we compute the fourth term, let us notice something interesting about the third term. The *automorphism group* $\mathrm{Aut}(T)$ of a tree T on labelled vertices $\{1, \ldots, n\}$ is defined to be the set of permutations $g \in S_n$ that fix T. Informally speaking, these are the symmetries of T. The automorphism group of any F-structure on a set X is defined similarly. So, for instance, the automorphism group of the last tree shown above is the two-element subgroup $G = \{(1)(2)(3), (1\ 2)(3)\}$ of S_3.

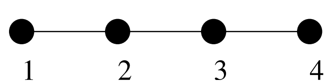
Given any group G of permutations, we may assign each element $g \in G$ weights $j_1(g), j_2(g), \ldots$, where $j_k(g)$ is the the number of k-cycles in g, as before, and form the ordinary generating function $\sum_{g\in G} p_1^{j_1(g)} p_2^{j_2(g)} \cdots p_n^{j_n(g)}$. The *cycle index* of G is defined to be $1/|G|$ times this generating function. In the example above, with $G = \{(1)(2)(3), (1\ 2)(3)\}$, the cycle index is

$$C_G(p_1, p_2, \ldots) = \frac{1}{2}(p_1^3 + p_1p_2).$$

Notice that the third term in Z_F, which we just computed, is equal to x^3 times this cycle index. In this case we got the group G from the last tree, but since the others are isomorphic to it, their automorphism groups have the same cycle index, although they are different groups.

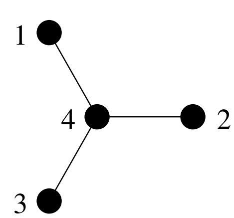
All this suggests that it might be more efficient to evaluate the terms of Z_F, not by considering each permutation g and looking at the trees that it fixes, but by considering each isomorphism type of tree, and looking at the permutations that fix it, that is, at its automorphism group. Let's take this approach to computing the fourth term of the cycle generating function Z_F for trees.

There are two isomorphism types of trees on four vertices. A typical tree of the first type is a "path"



,

and there are 12 trees isomorphic to and including this one. A typical tree of the second type is a "star"



,

and there are 4 trees of this type. A "path" tree has two automorphisms: the automorphism group of the one shown above consists of the identity and the permutation $g = (1\ 4)(2\ 3)$. Each of the twelve "path" trees therefore contributes $p_1^4 + p_2^2$ to the total $\sum_{g\in S_4} |(\text{trees})^g| p_1^{j_1}(g) p_2^{j_2(g)} \cdots$, for a net contribution of

$$12(p_1^4 + p_2^2).$$

The automorphism group of a "star" tree is the symmetric group on the labels of the three outer vertices, the center vertex being always fixed. For the one shown above it is the subgroup of S_4 which is a copy of S_3, with all its elements fixing 4. Each "star" tree therefore contributes $p_1^4 + 3p_1^2p_2 + 2p_1p_3$, for a net contribution of

$$4(p_1^4 + 3p_1^2p_2 + 2p_1p_3).$$

Hence the fourth term of Z_F is

$$x^4\left(\frac{1}{2}(p_1^4 + p_2^2) + \frac{1}{6}(p_1^4 + 3p_1^2p_2 + 2p_1p_3)\right).$$

Note that this is x^4 times the sum of the cycle indices for the automorphism groups of the two types of trees. This is not an accident; we will prove the general result below.

Returning to our example and writing out all terms computed so far, we have for the species F of labelled trees,

$$Z_F = 1 + xp_1 + \frac{x^2}{2}(p_1^2 + p_2) + \frac{x^3}{6}(3p_1^3 + 3p_1p_2) + \frac{x^4}{24}(16p_1^4 + 12p_2^2 + 12p_1^2p_2 + 8p_1p_3) + \cdots .$$

Let's make a couple of additional observations about this result. First of all, every tree is obviously fixed by the identity element, so the term involving p_1^n has coefficient equal to the number of labelled trees, which we know should be n^{n-2}. This agrees with the coefficients you see above: 1, 1, 1, 3, 16. We can extract just these terms by setting $p_1 = 1$ and every other $p_k = 0$. Then we get

$$Z_F(1, 0, 0, \ldots; x) = 1 + x + \frac{x^2}{2} + 3\frac{x^3}{6} + 16\frac{x^4}{24} + \cdots ,$$

which is the usual exponential generating function for labelled trees.

Second, suppose we set every $p_k = 1$. Then the cycle index of each automorphism group becomes 1, and we get

$$Z_F(1, 1, 1, \ldots; x) = 1 + x + x^2 + x^3 + 2x^4 + \cdots ,$$

which is the *ordinary* generating function for *unlabelled* trees. Here you begin to see why exponential generating functions count labelled structures and ordinary generating functions count unlabelled ones: each kind of generating function is a special value of the cycle generating function, which incorporates them both.

2. Properties of the cycle generating function

We now turn to some general properties of the cycle generating function $Z_F(p_1, p_2, \ldots; x)$. We have already seen some of these properties in the example in Section 1. To establish the main properties, we will look at the formula for Z_F from two different vantage points.

First of all, we can view Z_F as the exponential generating function for the sums

$$\sum_{g\in S_n} |F([n])^g| p_1^{j_1(g)} p_2^{j_2(g)} \cdots p_n^{j_n(g)}.$$

This sum in turn is an ordinary generating function: a weighted enumerator for pairs (f, g) consisting of an F-structure f and a permutation g that fixes f, with the weight of the pair keeping track of the cycle structure of g, and given by $p_1^{j_1(g)} p_2^{j_2(g)} \cdots p_n^{j_n(g)}$. Hence Z_F is the mixed exponential/ordinary generating function for structures consisting of a pair (f, g), weighted by the cycle monomial of g. Now if we set $p_1 = 1$ and $p_k = 0$ for all other k, we are setting the weight to zero for all pairs in which g is not the identity element, and to 1 for pairs $(f, 1)$. This gives the exponential generating function for the structures f, since to choose a pair $(f, 1)$ is just to choose f. We summarize these observations as a theorem.

Theorem 1. *The special value*

$$Z_F(1, 0, 0, \ldots; x)$$

of the cycle generating function for a species F is equal to the usual exponential generating function for F-structures.

Next, let us rewrite the formula for Z_F by summing first over structures f, and then over permutations g that fix f, that is, over $g \in \operatorname{Aut}(f)$. We have

$$Z_F = \sum_n \frac{x^n}{n!} \sum_{f \in F([n])} \sum_{g \in \operatorname{Aut}(f)} p_1^{j_1(g)} p_2^{j_2(g)} \cdots p_n^{j_n(g)}.$$

Now for different isomorphic structures f, that is, for different structures f in the same orbit of the action of S_n on $F([n])$, the sum $\sum_{g \in \operatorname{Aut}(f)} p_1^{j_1(g)} p_2^{j_2(g)} \cdots p_n^{j_n(g)}$ is the same. For each orbit of F-structures, this sum appears as many times as the size of the orbit. By the orbit-stabilizer theorem, the size of the orbit is given by $|\operatorname{orb(f)}| = |S_n|/|\operatorname{Aut}(f)|$, since $\operatorname{Aut}(f)$ is, by definition, the stabilizer of f in S_n. Hence we have

$$Z_F = \sum_n \frac{x^n}{n!} \sum_{\operatorname{Orb}(S_n, F([n]))} \frac{n!}{|\operatorname{Aut}(f)|} \sum_{g \in \operatorname{Aut}(f)} p_1^{j_1(g)} p_2^{j_2(g)} \cdots p_n^{j_n(g)},$$

which is equal to

$$Z_F = \sum_n x^n \sum_{\operatorname{Orb}(S_n, F([n]))} C_{\operatorname{Aut}(f)}(p_1, p_2, \ldots).$$

Here $\operatorname{Orb}(S_n, F([n]))$ denotes the set of orbits of S_n acting on the F-structures on $[n]$, and $C_{\operatorname{Aut}(f)}(p_1, p_2, \ldots)$ is the cycle index of the automorphism group of any structure f belonging to the chosen orbit.

Now an orbit of S_n on $F([n])$ is the same thing as an *unlabelled* F-structure, so we have arrived at a new interpretation of Z_F: it is the *ordinary* generating function for the cycle indices of the automorphism groups of *unlabelled* F-structures. From this we deduce a second theorem giving a special value of Z_F.

Theorem 2. *The special value*

$$Z_F(1, 1, 1, \ldots; x)$$

of the cycle generating function for a species F is equal to the ordinary generating function for unlabelled *F-structures.*

Proof. This follows from the preceding observations and the fact that for any group of permutations G, the cycle index satisfies $C_G(1, 1, 1, \ldots) = 1$. The latter fact holds because $C_G(p_1, p_2, \ldots)$ is by definition $1/|G|$ times a sum of $|G|$ monomials in the variables p_k. □

3. More examples

In this section we will compute the cycle generating functions for some of the most important species.

Example. Let F be the *trivial* species, with only one F-structure on $[n]$ for every n. Then of course there is only one unlabelled F-structure, and its automorphism group is all of S_n. It follows that the cycle generating function for the trivial species is

$$Z_{\text{triv}} = \sum_{n=0}^{\infty} x^n C_{S_n}(p_1, p_2, \ldots), \tag{1}$$

the ordinary generating function for the cycle indices of the symmetric groups S_n. Of course this is only useful if we can calculate $C_{S_n}(p_1, p_2, \ldots)$. However, Z_{triv} can also be viewed as a mixed ordinary/exponential generating function for permutations, in which each permutation counts with a weight monomial $p_1^{j_1} p_2^{j_2} \cdots$ for a permutation with j_1 1-cycles, j_2 2-cycles, and so on. Without this extra detail, we would count permutations as composite structures (trivial) $\circ$ (cycle), giving the exponential generating function

$$e^{-\log(1-x)} = e^{x+x^2/2+x^3/3+\cdots}.$$

But it is easy to modify this to keep track of the cycle lengths: in the k-th term of the exponent, we just introduce a factor p_k. Hence we have the formula

$$Z_{\text{triv}} = e^{p_1 x + p_2 x^2/2 + p_3 x^3/3 + \cdots}. \tag{2}$$

By expanding and comparing term by term with equation (1) above, we can use this formula to compute the cycle indices $C_{S_n}(p_1, p_2, \ldots)$ for arbitrary n.

Let's look briefly at the special values given by the theorems in Section 2. First, if we set $p_1 = 1$ and all other $p_k = 0$, we get

$$Z_{\text{triv}}(1, 0, 0, \ldots; x) = e^x.$$

This agrees with the familiar exponential generating function for the trivial species. Next, if we set all $p_k = 1$, we get

$$e^{-\log(1-x)} = \frac{1}{1-x}.$$

This is the *ordinary* generating function for the *unlabelled* trivial species. There is one trivial unlabelled structure for each n, and we have obtained, as we expect, the geometric series with all coefficients equal to 1.

Example: The species of *linear orderings*. The automorphism group of a linear ordering is trivial, since any permutation of the elements changes the order. All the linear orderings of $[n]$ form a single S_n orbit, which is to say, there is just one unlabelled linear ordering for each n. Using the interpretation of Z_F as an ordinary generating function for cycle indices of automorphism groups, we immediately obtain the formula

$$Z_{\text{lin}} = \sum_{n=0}^{\infty} x^n p_1^n = \frac{1}{1 - p_1 x}.$$

The first special value is

$$Z_{\text{lin}}(1, 0, 0, \ldots; x) = \frac{1}{1-x},$$

which is the exponential generating function for linear orderings. The second special value is

$$Z_{\text{lin}}(1, 1, 1, \ldots; x) = \frac{1}{1-x}.$$

Of course this comes out to the same thing because Z_{lin} only depends on p_1 and x. However, the result is now to be interpreted as the ordinary generating function for unlabelled linear orderings. Since there is just one of these for each n, we get the geometric series.

Example: The species of *permutations*. Here we must be clear about how S_n acts on the set of permutations of $[n]$, which is to say, on itself. It acts by permuting the numbers in the cycle notation for a permutation. In group-theoretic terms, this means that g sends σ to $g\sigma g^{-1}$. In other words, S_n acts on itself by conjugation. Now let's see how many permutations σ are fixed by g. We have

$$g\sigma g^{-1} = \sigma \quad \text{if and only if} \quad g\sigma = \sigma g,$$

that is, if and only if σ commutes with g. Note that in particular, g fixes σ if and only if σ fixes g. This is an extremely useful fact. For the number of elements σ fixed by g is equal to the number that fix g, that is, to the stabilizer of g in the action of S_n on itself by conjugation. By the orbit-stablizer theorem, this number is equal to $n!/|C(g)|$, where $C(g)$ is the conjugacy class of g, consisting of all permutations with the same cycle structure as g. Using this, we find

$$Z_{\text{perm}} = \sum_{n=0}^{\infty} \frac{x^n}{n!} \sum_{g \in S_n} \frac{n!}{|C(g)|} p_1^{j_1(g)} p_2^{j_2(g)} \ldots p_n^{j_n(g)}.$$

In this formula, the $n!$ in numerator and denominator cancel, and $|C(g)|$ is exactly the number of terms in the sum with a given monomial $p_1^{j_1(g)} p_2^{j_2(g)} \ldots p_n^{j_n(g)}$. Hence each such monomial occurs with a net coefficient exactly equal to x^n. In short, we have

$$Z_{\text{perm}} = \sum_{n=0}^{\infty} x^n \sum_{|\lambda|=n} p_{\lambda_1} p_{\lambda_2} \cdots p_{\lambda_l}.$$

This is the ordinary generating function for all partitions λ, counted with weight $x^n p_1^{j_1} p_2^{j_2} \cdots$, where n is the size of λ and j_k is the number of parts equal to k in λ. Using our familiar techniques for partition enumeration, we can rewrite this as

$$Z_{\text{perm}} = \prod_{k=1}^{\infty} \frac{1}{1 - p_k x^k}.$$

The first special value is

$$Z_{\text{perm}}(1, 0, 0, \ldots; x) = \frac{1}{1 - x}.$$

This is the exponential generating function for permutations. As expected it is the same as the exponential generating function for linear orderings. Note, however, that the full cycle generating functions Z_{lin} and Z_{perm} are not the same. This shows very clearly the important point that linear orderings and permutattions are *two different species.*

The second special value is

$$Z_{\text{perm}}(1, 1, 1, \ldots; x) = \prod_{k} \frac{1}{1 - x^k}.$$

This is the familiar ordinary generating function for partitions. It should also be the ordinary generating function for unlabelled permutations. Is this correct? Well, an "unlabelled" permutation is entirely determined by its cycle structure, that is, two permutations are in the same orbit if they have the same cycle lengths. The cycle lengths form a partition of n, so unlabelled permutations are nothing but partitions, and the formula checks.

4. Decorated F-structures

We have seen that the two special values $Z_F(1, 0, 0, \ldots; x)$ and $Z_F(1, 1, 1, \ldots; x)$ have enumerative meaning, but what about the full cycle generating function itself? The answer to this comes from the Polya-Redfield theorem. Remember that Z_F is the ordinary generating function for the cycle indices of automorphism groups of unlabelled structures. Now suppose we have a set of "colors"

A, with some symbolic weights assigned, and enumerated by an ordinary generating function $G(y)$. We use the notation $p_k[G(y)] = G(y^k)$. Then the Polya-Redfield theorem tells us that

$$C_{\mathrm{Aut}(f)}[G(y)] \underset{\mathrm{def}}{=} C_{\mathrm{Aut}(f)}(G(y), G(y^2), \ldots)$$

is the ordinary generating function for C-colorings of the set $[n]$, up to symmetries given by the automorphism group of f. In other words, it is the ordinary generating function for unlabelled structures isomorphic to f, with their vertices "decorated" with colors from the set A. Summing this over all F-structures, with a factor x^n to keep track of the size of the vertex set, gives

$$Z_F[G(y)] = Z_F(G(y), G(y^2), \ldots; x),$$

and this is the ordinary generating function for all unlabelled vertex-decorated F-structures, with a weight $x^n y^k$ if it has n vertices and the coloring contributes weight y^k. More generally, the generating function for the colors can be a weight enumerator $G(y_1, y_2, \ldots, y_r)$ in any number of variables, and then for p_k we substitute $p_k[G(\mathbf{y})] = G(y_1^k, y_2^k, \ldots, y_r^k)$.

The simplest possibility, but also in some ways the most general, is that the set A consists of r colors each with its own symbolic variable y_i. Then the enumerator of the colors is $y_1 + y_2 + \cdots + y_r$, and

$$p_k[y_1 + y_2 + \cdots + y_r] = y_1^k + y_2^k + \cdots + y_r^k.$$

This quantity is called the k-th *power-sum* of the variables y_i. This is the reason I have used the letter p for the variables p_k, to stand for "power-sum." We can summarize the above considerations as a theorem.

Theorem 3. *The special value*

$$Z_F[y_1 + y_2 + \cdots] = Z_F(p_1(\mathbf{y}), p_2(\mathbf{y}), \ldots; x),$$

where $p_k(\mathbf{y}) = p_k[y_1 + y_2 + \cdots]$ is the k-th power-sum, is the ordinary generating function for unlabelled F-structures decorated with colors on the vertices, and enumerated with weight $x^n y_1^{c_1} y_2^{c_2} \cdots$, where n is the number of vertices and c_i is the number that receive color i.

An important point about this theorem is that the "special" value involved is actually a fully general value. In the theory of symmetric functions one shows that when there are infinitely many variables y_i, the power-sums $p_k(\mathbf{y})$ are in effect independent variables. The interpretation of Z_F as an ordinary generating function for decorated unlabelled F-structures therefore involves no loss of information.

Example: If we take the trivial structure on a set of n elements, decorate the elements with colors $1, \ldots, r$, and consider this as an unlabelled structure, we have a distribution of n identical items to r distinct recipients. This is just a multiset of n elements from the r colors. We assign color i a variable y_i. Then the ordinary generating function for multisets of the colors, counted with a monomial $x^n y_1^{k_1} \cdots y_r^{k_r}$, where n is the size of the multiset and k_i is the number of copies of color i in it, should be given by

$$Z_{\mathrm{triv}}(p_1(\mathbf{y}), p_2(\mathbf{y}), \ldots; x).$$

Using formula (2) for Z_{triv}, we see that this is equal to

$$\Omega[x(y_1 + \cdots + y_r)],$$

where we define

$$\Omega = e^{p_1 + p_2/2 + p_3/3 + \cdots},$$

and the square bracket notation means as usual that p_k goes to $p_k[x(y_1+\cdots+y_r)] = x^k(y_1^k+\cdots+y_r^k) = x^k p_k(\mathbf{y})$. Now notice that in general we have $p_k[X+Y] = p_k[X]+p_k[Y]$ for any quantities X, Y. Hence the exponent in the definition of Ω is additive, and Ω itself is multiplicative:

$$\Omega[X+Y] = \Omega[X]\Omega[Y].$$

We can use this compute $\Omega[xy_1+\cdots+xy_r]$ by computing $\Omega[xy_i]$ and multiplying. However

$$\Omega[xy_i] = e^{-\log(1-xy_i)} = \frac{1}{1-xy_i},$$

so

$$\Omega[x(y_1+\cdots+y_r)] = \prod_{i=1}^{r} \frac{1}{1-xy_i}.$$

This agrees with the generating function for multisets that we would get by familiar ordinary generating function methods.

5. Plethysm

Our final topic will be the analog for cycle generating functions of the product and composition principles for exponential generating functions. It turns out that the stray variable x in the cycle generating function is an unnecessary nuisance and it is best to get rid of it before proceeding further. We define the *reduced* cycle generating function for a species F to be

$$\widehat{Z}_F(p_1,p_2,\ldots) = Z_F(p_1,p_2,\ldots;1).$$

There is no real loss of information in working with $\widehat{Z}_F$ instead of Z_F. To see this, note that in the x^n term of Z_F, every monomial $p_1^{j_1(g)}p_2^{j_2(g)}\ldots p_n^{j_n(g)}$ has degree n, if we agree to consider p_k as having degree k. If we substitute $p_k \mapsto x^k p^k$ in such a monomial, the effect is to multiply it by x^n. This shows that

$$Z_F(p_1,p_2,\ldots;x) = \widehat{Z}_F(xp_1,x^2p_2,\ldots),$$

so we can always recover Z_F if we know $\widehat{Z}_F$.

The product principle for cycle generating functions is the same as the familiar one for exponential generating functions.

Theorem 4. *If $F = GH$ is a product structure, then*

$$\widehat{Z}_F = \widehat{Z}_G\widehat{Z}_H$$

(and hence also $Z_F = Z_G Z_H$).

Proof. Recall that a GH structure on $[n]$ consists of an ordered partition of $[n]$ into subsets A_1 and A_2, with a G structure on A_1 and an H structure on A_2. To decorate the GH structure is the same thing as to decorate the G structure on A_1 and the H structure on A_2 independently. When we drop the labels, that is, look at S_n orbits, it no longer matters which elements belong to A_1 and which to A_2: an unlabelled decorated GH structure just consists of an unlabelled decorated G structure and an unlabelled decorated H structure. These can be chosen independently, and the product principle is now reduced to the usual one for ordinary generating functions.

What this actually shows is that

$$\widehat{Z}_F[Y] = \widehat{Z}_G[Y]\widehat{Z}_H[Y]$$

for any Y. But if we take $Y = y_1+y_2+\cdots$, then $p_k[Y] = p_k(\mathbf{y})$, and these power-sums are algebraically independent for all k. Hence the identity evaluated at Y implies the full identity that we wanted to prove. $\square$

An interesting feature of this proof is that by using the interpretation of Z_F as an ordinary generating function for decorated unlabelled structures, we have reduced the product principle to the one for ordinary generating functions. At the same time, since the exponential generating function for F structures is a special value of Z_F, this result implies the product principle for exponential generating functions.

To give the cycle generating function version of the composition principle, we define a new operation called plethysm.

Definition Let G and H be formal power series in the variables p_1, p_2,..., and assume that the constant term of H is zero. The *plethysm* $G * H$ of H into G is defined to by

$$G * H = G(p_1 * H, p_2 * H, \ldots),$$

where

$$p_k * H = H(p_k, p_{2k}, p_{3k}, \ldots)$$

In other words, to plethystically substitute H into G, we substitute $p_k * H$ for each variable p_k in G, and to compute $p_k * H$, we substitute p_{jk} for each variable p_j in H.

Lemma 1. *The plethysm is related to the bracket operation by the identity*

$$G[H[Y]] = (G * H)[Y].$$

Proof. To compute $H[Y]$ we replace each variable p_j in H by $p_j[Y]$, which is defined as the result of replacing every symbol in Y by its j-th power. To compute $G[H[Y]]$ we replace each variable p_k in G by $p_k[H[Y]]$. This is obtained by replacing every symbol in $H[Y]$ by its k-th power. However, we could have gotten the same result by replacing each variable p_j in H by $p_{jk}[Y]$ in the first place. In other words, $p_k[H[Y]] = (p_k * H)[Y]$ according to our definition of $p_k * H$. Now we substitute this for each p_k in G to obtain $(G * H)[Y]$. □

Of course, the reason we defined plethysm as we did was to make this lemma work. Thus the lemma is largely just a matter of notation. Now we are ready for a real theorem.

Theorem 5. *If $F = G \circ H$ is a composite structure, then*

$$\widehat{Z}_F = \widehat{Z}_G * \widehat{Z}_H.$$

Proof. We will prove that

$$\widehat{Z}_F[Y] = (\widehat{Z}_G * \widehat{Z}_H)[Y]$$

for any Y. This is sufficient to prove the general identity, for the same reason as in the proof of the product principle. By the Lemma, the right-hand side above is equal to

$$\widehat{Z}_G[\widehat{Z}_H[Y]].$$

We must verify that this is a generating function for the same kind of decorated structures as $\widehat{Z}_F[Y]$.

We can interpret $\widehat{Z}_G[\widehat{Z}_H[Y]]$ as the generating function for unlabelled G structures with their vertices decorated by "colors" whose ordinary generating function is $\widehat{Z}_H[Y]$. However, the latter is the generating function for decorated unlabelled H-structures. Therefore $\widehat{Z}_G[\widehat{Z}_H[Y]]$ enumerates structures consisting of an unlabelled G-structure on a set whose elements are in turn decorated with unlabelled decorated H-structures.

Now let's see what an unlabelled decorated $G \circ H$ structure on a set X looks like. We are to choose a partition Π of X, a an H-structure on each block of Π, and a G-structure on the set of blocks of Π. Moreover, we are to decorate the vertices of X, which is the same thing as decorating the vertices of the various H-structures in the composite structure independently. Removing labels, we are left with a multiset of unlabelled decorated H-structures, with a G-structure on the elements of the multiset. But this is the same thing as an unlabelled G-structure on elements decorated by unlabelled decorated H-structures, which is what $\widehat{Z}_G[\widehat{Z}_H[Y]]$ enumerates. $\square$

Example: The cycle generating function for the species of set partitions can be computed using plethysm. Our species is a composite (trivial) $\circ$ (non-empty set). Previously, we used this to get the exponential generating function e^{e^x-1} for set partitions. Here, in place of e^x we need to use the cycle generating function for a trivial species, and in place of functional composition, we need to use plethysm.

In the notation introduced above, we have

$$\widehat{Z}_{\text{triv}} = \Omega = e^{p_1+p_2/2+\cdots},$$

and subtracting the constant term, which counts the trivial structure on the empty set, we have

$$\widehat{Z}_{\text{non-empty}} = \Omega - 1.$$

Hence

$$\widehat{Z}_{\text{partition}} = \Omega * (\Omega - 1).$$

Let's compute some terms of this, keeping track of everything up to degree 4. We have

$$\begin{aligned}
p_1 * (\Omega - 1) &= \Omega - 1 = e^{p_1+p_2/2+p_3/3+p_4/4+\cdots} - 1 \\
p_2 * (\Omega - 1) &= e^{p_2+p_4/2+\cdots} - 1 \\
p_3 * (\Omega - 1) &= e^{p_3+\cdots} - 1 \\
p_4 * (\Omega - 1) &= e^{p_4+\cdots} - 1.
\end{aligned}$$

Then

$$\begin{aligned}
\Omega * (\Omega - 1) &= e^{p_1*(\Omega-1)+p_2*(\Omega-1)/2+p_3*(\Omega-1)/3+p_4*(\Omega-1)/4+\cdots} \\
&= e^{p_1*(\Omega-1)} e^{p_2*(\Omega-1)/2} e^{p_3*(\Omega-1)/3} e^{p_4*(\Omega-1)/4} \cdots .
\end{aligned}$$

Expanding the various exponentials, substituting for $p_k * (\Omega - 1)$ from the chart above, collecting terms, and putting back the x^n factors for clarity, we find after some work:

$$\begin{aligned}
Z_{\text{partition}}(p_1, p_2, \ldots; x) = {} & 1 + x\,p_1 + \frac{x^2}{2}\left(2\,{p_1}^2 + 2\,p_2\right) + \frac{x^3}{6}\left(5\,{p_1}^3 + 9\,p_1\,p_2 + 4\,p_3\right) \\
& + \frac{x^4}{24}\left(15\,{p_1}^4 + 42\,{p_1}^2\,p_2 + 21\,{p_2}^2 + 24\,p_1\,p_3 + 18\,p_4\right) + \cdots .
\end{aligned}$$

Looking at the x^3 term, this tells us for instance that there are 5 labelled partitions of a 3 element set (the coefficient of the term involving p_1^3), and 3 unlabelled ones (setting all $p_k = 1$ in the x^3 term). You can easily check by hand that this is correct.

Now let's apply our two standard specializations to this. Since we are using reduced cycle generating functions, in order to recover the exponential generating function for set partitions we should set $p_1 = x$ and $p_k = 0$ for all $k > 1$. You can check from the definition of plethysm that making this substitution in $F * G$ is equivalent to first applying it to each of F and G, then composing the resulting functions $F(x)$ and $G(x)$. In other words, the plethysm principle for cycle generating functions reduces under this substittution to the composition principle for exponential generating functions. In particular, our cycle generating function $\Omega * (\Omega - 1)$ for set partitions reduces to the exponential generating function e^{e^x-1} that we knew before.

On the other hand, to get the ordinary generating function for unlabelled structures we want to set $p_k = x^k$. From the definitions we see that this substitution sends $F * G$ to $F(G(x), G(x^2), \ldots)$, where $G(x)$ is shorthand for $G(x, x^2, \ldots)$, that is, for the ordinary generating function for the unlabelled structures counted by G. In the present example, $\Omega - 1$ is the cycle generating function for the trivial non-empty species, so setting $p_k = x^k$ in $\Omega - 1$ gives the ordinary generating function $G(x) = x/(1-x)$ for the unlabelled trivial non-empty species.

Now let's calculate

$$\Omega(G(x), G(x^2), \ldots) = e^{x/(1-x)+(1/2)x^2/(1-x^2)+\cdots}.$$

The expression in the exponent is

$$\begin{aligned} \sum_{n=1}^{\infty} \frac{1}{n} \sum_{k=1}^{\infty} x^{kn} &= \sum_{k=1}^{ow} \sum_{n=1}^{\infty} \frac{x^{kn}}{n} \\ &= \sum_{k=1}^{\infty} \ln\left(\frac{1}{1-x^k}\right). \end{aligned}$$

Exponentiating this gives

$$\Omega(G(x), G(x^2), \ldots) = \prod_{k=1}^{\infty} \frac{1}{1-x^k}$$

as the ordinary generating function for unlabelled set partitions. Since an unlabelled partition of an n element set is merely an integer partition of n, this agrees with the ordinary generating function for integer partitions, as expected.