COMMENTS ON EGFS

> What is the black king in chess? This is a strange question, and the most satisfactory way to deal with it seems to be to sidestep it slightly... What matters about the black king is not its existence, or its intrinsic nature, but the role that it plays in the game.
>
> The abstract method in mathematics, as it is sometimes called, is what results when one takes a similar attitude to mathematical objects. This attitude can be encapsulated in the following slogan: a mathematical object *is* what it *does*.
>
> – Tim Gowers

> If we can somehow associate a binomial poset with generating functions of the form $\sum f(n)x^n/B(n)$, then we will have... provided some justification of the heuristic principle that ordinary generating functions are associated with the nonnegative integers, exponential generating functions with sets, Eulerian generating functions with vector spaces, and so on.
>
> – Richard Stanley

> (OGFs) are ideally suited for counting ordered structures like integer partitions, ordered trees, and words. But many combinatorial families are really *sets*, where the order is immaterial. For these the natural concept is that of an exponential generating function.
>
> – Doron Zeilberger

Rather than continuing the attempt to "explain" the difference between $\sum a_n x^n$ and $\sum a_n x^n/n!$, it is more important to think in the spirit of the quote by Gowers above, and ask what these generating functions *do* that distinguishes the heuristic principles associated with them.

1. Multiplication

If we multiply two ogfs for type A and type B structures, we get

$$\left(\sum a_n x^n\right)\left(\sum b_n x^n\right) = \sum \left(\sum_{k=0}^{n} a_k b_{n-k}\right) x^n .$$

Thus, the resulting coefficient arises from taking an ordered pair of a type A object and a type B object, where the underlying set on which the total combined object lives is already cut into the type A and type B parts.

On the other hand, doing the same for two egfs we get

$$\left(\sum a_n x^n/n!\right)\left(\sum b_n x^n/n!\right) = \sum \left(\sum_{k=0}^{n} \binom{n}{k} a_k b_{n-k}\right) x^n/n! .$$

Thus, the resulting coefficient arises from taking all possible combinations of type A structures and type B structures on a set of n elements, where the underlying set on which the total combined object lives is *not* already cut into type A and type B parts. What the $\binom{n}{k}$ factor *does* is to allow all possible such decompositions of the underlying set to host the two possible structures.

NOTE: It is very reasonable to ask the question "if egfs have to deal with sets, why is the ogf for $\binom{n}{k}$ as k varies the one that has a nice product form?" My best heuristic answer to this is that *the structure of being a* subset is not the same as *being a structure on a* set. In fact, the set of k-subsets can be viewed as a collection of ordered sequences of 0's and 1's, and hence falls into Zeilbergers ogf description.

2. Composition

If we compose two ogfs for type A and type B structures, where $b_0 = 0$, we get

$$A(B(x)) = \sum a_k (B(x))^k .$$

Thus, the summands on the right-hand side are counting ordered sequences of k type B objects, and then giving each of the ordered sequences a type A structure – this interpretation forces the type A structures to be structures on ordered sequences!

EXAMPLE: If type A is the trivial structure on an ordered string of length n, then the ogf for type A is x^n. If type B is the structure giving the number of trivially structured sets that are empty or contain one element, then the ogf for type B is $1 + x$ (here we allow $b_0 \neq 0$ since we are going to compose it with the finite ogf x^n.) Composing these ogfs gives the function $(1 + x)^n$, which is the ogf for the number of k element subsets of an n-set.

If we compose two egfs for type A and type B structures, where $b_0 = 0$, we get

$$A(B(x)) = \sum a_k (B(x))^k / k! .$$

As we have discussed, this allows the type B "blocks" to be unordered; thus, the type A structures allowed on these blocks "should be" structures that are on unordered sets of elements, because of the form that composing egfs produces.

For more details and examples, see Chapter 14 of van Lint and Wilson.