On the Foundations of Combinatorics

Daniel Nikpayuk

December 9, 2014

1 Preface

I have long sought "axiomatic" style foundations for combinatorics; it has been the hardest area for me to learn, and axioms would go a long way in building an understanding and intuition. After many years of searching, I have concluded that no such "satisfactory" foundations exist. The thing is, after a lot of research and exploration, in various areas of math, but also other areas like linguistics and computing science, a theory has been forming in my mind—both by intention as well as through luck—which I think is a first good step in the right direction. I will here and now say this is my motivation for writing the article you are about to read, but I will also add that part of the intention is infact a selfish one: By writing an article I come to understand the material better myself, not to mention it later provides reference material when it is done.

In anycase, the reason for my dissatisfaction with the existing pedagogical design of combinatorics should be addressed. It stems from combinatorical theory traditionally asking the four following questions:

1. Does a structure *exist*?
2. If the structure exists, is it *unique*?
3. If it exists and is not unique, *how many* such structures are there?
4. If it exists and is not unique, then of all possibilities is there an *optimal* one?

I don't disagree with any of these as to be included in the theory as important and recurring themes, but I do disagree with them to be the logic for which combinatorial theory is taught. To be so bold, I choose to study as my foundations the two words:

representation and *arrangement*.

which in and of themselves are nothing new: It's the emphasis then that I am resolved to change. I make analogy to calculus here in that it studies the words continuity and convergence, and it does so, not by using the logic of "limits" to develope the proofs, but by using the "epsilons" (ϵ) and "deltas" (δ) of Weierstraß. This is the other part of my intention for this writing: that of rigourous foundations to put combinatorics on par with the other long-standing and polished branches of mathematics.

A note should be mentioned about the writing style of this article itself (regardless of its content). For the most part, I try to maintain a formal style throughout, especially with the notations and definitions, but I do not consider this to be a formal article, so the words between the math that are mine will be of a more relaxed and conversational nature; for me it's more important that things flow, this way I enjoy it more, and as a result, it is usually the case the reader does as well... probably.

2 review

Though I call it review, I have made some small changes to some of the definitions, so it is worth reading over quickly. The definitions I have left unaltered are there because I make direct reference to them in the development of theory of this article.

2.1 set theory

I'm introducing some of my own notation here.

So to begin, let $c = (a, b)$ be some ordered pair, to represent a from the view point of c I write $c_{|1|}$ which indicates the *first coordinate* of c. Moreover, I write $c_{|2|}$ to represent b, the *second coordinate* of c.

The *difference* of two sets A, B written $A - B$ is defined to be a set containing exactly those elements of A that are not in B.

Next, let $f : A \to B$ be a mapping, to represent A with respect to f, I write $\mathrm{Dom}(f)$ and call it the *domain* of f. To represent B, it is written $\mathrm{Ran}(f)$ and is called the *range* of f. This much is pretty standard.

An *equivalence relation* of a set is a relation *on* that set which is: *reflexive*, *symmetric* and *transitive*.

The *product* of an *indexed system of sets* $\langle S_i \mid i \in I \rangle$ is the collection of all functions $f : I \to \bigcup_{i \in I} S_i$ such that $f(i) \in S_i$. This is a way of defining an "abstraction" of a cartesian product of sets.

An ordering of a set is said to be *partial* if some elements are not comparable, and is otherwise said to be *total* or *linear*. A subset of a partially ordered set which is linearly ordered is called a *chain*. Within the realm of ordered sets, the difference between a *minimum* element and a *minimal* element is that a minimum element is comparable to every other element (totally ordered set) and is still the *least* element, whereas a minimal element is a least element but isn't necessarily comparable with the all of the remaining (applicable to partially ordered sets).

Well Ordered Set A set S with an ordering imposed on it is said to be well ordered if:

1. the ordering is linear.
2. every nonempty subset $R \subseteq S$ has a least element.

2.2 graph theory

I've modified the common graph theoretic definitions slightly to suit my needs, the semantics are the same, its simply the wording that is different. It is worthwhile to look over, and in doing so one will note—possibly with some dissatisfaction—these versions aren't as "efficient" as the mainstream; the reasoning behind this is that it will smoothen things out when it comes time to introduce the definitions of the theory I am presenting in the following section.

Graph A graph G is a pair of sets (V, E) where V is called the vertex set (of vertices of G), and E is called the edge set of G, and is such that it consists of unordered pairs of vertices from V.

When two graphs G, H (or possibly more) are in a given context, to distinguish between their vertex and edge sets one writes V_G, E_G and V_H, E_H. I will infact use this notation when only one graph is under discussion.

I'm not going to worry or bother with "multisets", which would entail an explanation about loops and multiple edges, the graphs I'm interested in are *simple*.

Given an edge $e = \{v_1, v_2\} \in E_G$ of some graph G, the vertices $v_1, v_2 \in V_G$ that compose it are said to be *incident* to e. Moreover, with respect to each other, the vertices v_1, v_2 are said to *adjacent*.

If G is some graph with $v \in V_G$, then the *degree* of the vertex v, written as $\deg(v)$ is equal to the number of edges $e \in E_G$ which are incident to it.

Subgraph Let G be a graph, a graph H is said to be a subgraph of G if $V_H \subseteq V_G$ and $E_H \subseteq E_G$.

Here is an "inbetween" definition I'm further adding to smoothen things out:

Ordering of a Graph Let G be a graph, the ordering $\mathrm{Ord}(G)$ is the set such that for every edge $\{v_1, v_2\} \in E_G$ (with $v_1, v_2 \in V_G$), the *ordered* pairs (v_1, v_2) and (v_2, v_1) are exactly those elements in $\mathrm{Ord}(G)$.

The members of $\mathrm{Ord}(G)$ are called *directed edges*.

Path Let G be a graph, let $v_1, v_2 \in V_G$. A sequence $(p_1, p_2, \ldots, p_n)$ with $p_1, p_2, \ldots p_n \in \mathrm{Ord}(G)$ such that $(p_s)_{|2|} \neq (p_t)_{|2|}$ for any $s \neq t$ is called a path from v_1 to v_2 if the following conditions are satisfied:

1. $(p_1)_{|1|} = v_1$
2. $(p_i)_{|2|} = (p_{i+1})_{|1|} \quad$ for $\quad 1 \leq i < n$
3. $(p_n)_{|2|} = v_2$

In particular, a path is called *closed* if $v_2 = v_1$ and is otherwise called *open*.

Connected Graph A graph G is called connected if given any two vertices $v_1, v_2 \in V_G$ with $v_1 \neq v_2$, there exists a path from v_1 to v_2. A graph G that is not connected is called disconnected.

The removal of a vertex v from a graph G, written as $G - v$, is defined as the removal of v from the vertex set V_G and any incident edges of v from E_G. Similarily, the removal of an edge e from E_G (but not the vertices incident to it) will be written as $G - e$.

Disconnect Vertex Let G be a graph and v a vertex. If there exists some connected subgraph H with $v \in V_H$ such that $H - v$ is disconnected, then v is called a disconnect vertex of G.

Similarily, an edge e of G is called a *bridge* if its removal causes some connected subgraph H —to which it belongs— to become disconnected.

Here I introduce a two part "process" definition which is useful in the latter:

Open Partition Let G be a connected graph possessing a disconnect vertex v. Let $\{H_i\}_{i \in I}$ be the collection of subgraphs obtained by the removal of v, then $\{H_i\}_{i \in I}$ is called an open partition of G.

Closed Partition Let G be a connected graph possessing a disconnect vertex v. Let $\{H_i\}_{i \in I}$ be an open partition of G using v. If one were to unite v and its appropriate edge(s) back onto to each individual subgraph H_i written as H_i', then the collection $\{H_i'\}_{i \in I}$ is called a closed partition of G.

It's worth noting a closed partition is also a collection of subgraphs of the original. Both types of graph partitions aren't partitions in the set-theoretic sense, but they're pretty close; one being disjoint whose union is almost the full graph, and the other who's union is the whole, but isn't exactly disjoint... but close.

Cyclic Graph A graph G is said to be cyclic if it contains a closed path, in which case, such a path is also called a cycle. Furthermore, a graph G is said to be acyclic if it contains no cycles.

Tree A tree T is a connected acyclic graph.

A vertex l of a tree T is called a *leaf* if $\deg(l) = 1$, all vertices of T that are not leaves are called *nodes*.

The reader will see later on that I am only interested in trees possessing at least one node.

3 Basic Concepts

One of the main ideas I am presenting here is the representation of functions. I take an analogy to ring theory, in which splitting fields of integer polynomials in $\mathbb{Z}[x]$ allow one to express many irrational real numbers "algebraically". It is true that there are important trancendental real numbers like π which are also worthy of expression, but such a thing is another matter altogether. The point is, the first priority of combinatorics is to express only those functions that do not trancend "computation", the ones which mathematicians take for granted and yet regard as *grammatical*.

In order to represent such grammatical functions, it is useful to think of them as sentences. It is no big suprise then that the inspiration of my definitions come from the study of linguistics (the study of human language), in which sentences are studied in a branch of linguistics called syntax.

Now, within syntax, sentences aren't studied for their meaning (this area is called semantics), they are studied as objects of information storage and retrieval. As an example, one particular feature of sentences is their word order: the "Subject-Verb-Object" paradigm of sentence structure that language teachers are so fond of teaching

their favorite or most troublesome pupils.[1] As it turns out, informal tree diagrams—used to represent the various kinds of structures of sentences—are popular in the teaching of syntax. This is a good cause to make use of trees as well, since they belong to math in the first place (naturally though, it will be in a more formal and enjoyable setting). In particular, I am interested in using such trees for path-oriented representations of functions, with the application of combinatorial formula manipulations in mind.

Finally, when I say "path-oriented", although it is not really worthwhile to explain exactly what I mean here (you'll find out soon enough), I will say the concept is a practical one, and is related to the "pointer" concept used in programming languages such as C++.

Before I move onto the first definition, I will add that the clever reader will notice I use a lot of linguistics terminology, but I thought it over and it's okay. For starters, I'm not that clever with names, so why make up a bunch of crappy ones that no one will like. Secondly and lastly, both linguistics and mathematics have a lot of terminology rooted in the ancient greek language, so its not an unnatural fit to begin with. In anycase, enjoy:

Clause A clause C is a pair of sets (G, M) such that G is a graph, and $M : S \to A$ is a mapping where its domain is a subset of the vertex set of G, which is to say $S \subseteq V_G$.

I extend the notions of *subgraphs*, *open* and *closed partitions* from graphs to clauses, the definitions themselves should be apparent.

Although I don't make use of generic clauses in this article, I leave the door open for their study, which is something I will comment on later. In the meantime, I need to narrow the focus somewhat:

Sentence A clause $U = (T, M)$ is called a sentence if its graph T is a tree containing a node $n \in V_T$ such that:

$$\mathrm{Dom}(M) = V_T - \{n\}$$

In such a case, n is called the *root* of the sentence, and is given the special character r_U.

If the context is clear the root r_U may be written without its subscript: r. It is worth noting that although the root does not belong to the domain, there exists a unique path from the root to every other vertex that does belong (which is infact every other vertex of the tree). This motivates the following notation:

Given a sentence $U = (T, M)$ and a path p from its root r to some vertex b, I write

$$U/p[b]$$

to mean

$$M(b)$$

though considering that b is implied within p already, I may—depending on the purpose—simply write it as:

$$U/p$$

which is to say that U/p is the image of b under M. Such an image U/p of a sentence will be called an *arguement* of U. As well, the vertex b that is "incident" to the path p will be called the *branch* of p. Finally, the set of such paths p from the root to a branch will be called the *syntax* of U and denoted as $\mathrm{Syn}(U)$. I am not finished here though, if one inverts the order of the edges within the path p to obtain $\bar{p}$, one may write

$$[b]\bar{p}\setminus U$$

to mean the same thing as

$$U/p[b]$$

the difference here is a subtle change from *verb-orientation* to *object-orientation*, which is conceptually more useful for formula manipulations—more will be explained later. Finally, if the context is clear, I will "hide" the reference U and subscript the path p resulting in:

$$_p[b] \qquad \text{or} \qquad [b]_{\bar{p}}$$

such notational definitions will be particularily important throughout the remainder of this article: don't worry yet about how they're applied, this is what much of the article is about.

A sentence S is called *simple* if the root r_S is its only node.

[1] the author should point out so as to not discriminate, but as it turns out the vast majority of the worlds languages—in which there is a preferred word order within a sentence—happen to prefer the S-O-V ordering.

Base Let B be a sentence satisfying the following conditions:

1. B is simple.
2. the syntax $\mathrm{Syn}(B)$ is well ordered.

then B is called a base.

I take advantage of the fact that the syntax of a base is well ordered by representing the paths of it as r^k—which is to say $r^k \in \mathrm{Syn}(B)$—where r is its root (and only node), and k is the k^{th} path within the well ordering.

3.0.1 formal power series

Though it is not entirely apparent just yet, a weak-point of my theory admittedly is that of a heavy load of definitions without any noticable applications for quite a while. This is why I will here take a short detour by giving a proper foundational definition of the formal power series used in ring theory, in which an existing "lack of" has always bothered me.

Before I do though, I will introduce a special case of the notation I will be exploring further down the road. This notation I am introducing is a way of representing the syntax of a base B.

The first thing to note is that such a syntax $\mathrm{Syn}(B)$ is finite, which holds since we are intending to define polynomials which are themselves finite in degree within rings. Using the notation defined as above, the paths of a base are represented as $x^0, x^1, \ldots, x^n$ for some finite n, where x is the root. I should add I'm using the natural numbers $\mathbb{N}$ to "notate" the well ordering.

To get the big picture of "all at once" though, I *concatenate* these paths using the plus $(+)$ sign:

$$\mathrm{Syn}(B) := 1 + x + x^2 + \ldots + x^n$$

Now I cheated a little here, so I will explain: the '1' is shortform for x^0, as well, the x is shortform for x^1. Also, as stated before, if I want to display any of the arguements of B, I simply write them as:

$$B/x^k \qquad \text{or} \qquad x^k \backslash B$$

where k is some natural number from 0 to n.

With this I have enough for our definition.

Let R be some ring, we wish to define a collection of bases $\mathcal{B}$ possessing the same cardinality (size) as R such that they all contain the same syntax, one with the following form:

$$\mathrm{Syn}(B) := 1 + x + x^2 + \ldots + x^n$$

for $B \in \mathcal{B}$ and some fixed n. In particular define their arguements as:

$$B/x^k = a^k \quad \text{for some } a \in R \text{ with } 0 \leq k \leq n$$

and do this for each $B \in \mathcal{B}$ such that they span all a in R.

This is $\frac{1}{2}$ of the picture though, we still need coefficients. How does one obtain these you ask?... through algebra's favorite trick... an equivalence relation of course.

The thing to note is that the bases $B \in \mathcal{B}$ possess mappings $m : \mathrm{Dom}(B) \to R$ all with the same range, so given some other mapping $g : R \to R$, one may "act" on these bases by defining compositions:

$$g \circ m : \mathrm{Dom}(B) \to R$$

Thus we define an "action map" to create our equivalence relation, where our definition of two bases $B, B' \in \mathcal{B}$ being equivalent is:

$$B \equiv B' \iff g \circ m = g \circ m'$$

where m is the map of B and m' is the map of B'. In such a case, to represent an equivalence class, one writes:

$$[B] = c_0 + c_1 x + c_2 x^2 + \ldots + c_n x^n$$

where the c_k are called coefficients, possessing the following property:

$$c_k = g([B]/x^k) \quad \text{given } 0 \leq k \leq n$$

This is possible because each base in a class has by definition the same syntax and the same coefficients. In such a case this equivalence class is called a polynomial.

Finally, if one has many such polynomials of various "degrees", it is easy then to define addition and multiplication to obtain a polynomial ring $R[x]$... and the rest they say is history.

3.1 Morphism Relationships between Clauses

There are various structural relationships between sentences that span outwards starting with sentences that are the most similar and leading to ones that are the least. For generality though, I make the following definitions using clauses:

Equality Two clauses are equal if their associated graphs are equal and if their associated mappings are equal.

This one's sort of trivial, but needed for tests of *logical extensionality*. In order to look at the other existing relationships, I need the following definition first:

Graph Isomorphism Two graphs G and H will be called isomorphic if there exists a edge preserving (and therefore path preserving) bijection between their respective vertex sets.

This notion of isomorphism naturally extends to clauses themselves:

Clause Isomorphism Two clauses W, X are isomorphic, written $W \simeq X$ if the following conditions are satisfied:

1. there exists an isomorphism i from the graph of W to the graph of X.
2. $W/p = X/i(p)$ for every path $p \in \mathrm{Syn}(W)$.

Clause Homomorphism Two clauses W, X are homomorphic, written $W \sim X$ if their graphs are isomorphic.

This definition is weaker than an isomorphism in the sense that these clauses may not preserve the extensions (images) of their respective mappings.

3.2 Sentence Composition and Grammars

Every node of a tree is a disconnect vertex.

Composable Sentences Two sentences W, X are said to be composable if there exists paths $p \in \mathrm{Syn}(W)$ and $q \in \mathrm{Syn}(X)$ such that $W/p = X/q$.

Compound Sentence A sentence Y is called a compound sentence if there exists a path $p \in \mathrm{Syn}(Y)$ such that:

1. the closed partition of Y using the branch b of p generates two subsentences A, B where the root of A is the root of Y, and b is the root of B.
2. $Y/p \simeq A/p \simeq B$.

Futhermore, if W, X are composable sentences such that:

1. there exists a homomorphism i from A to W, with $A/q = W/i(q)$ for all $q \neq p$ (almost isomorphic).
2. $B \simeq X$.

Then Y is said to be the composition of W, X.

Any such branch b_p which allows for a compound sentence Y is called a *compound node*.

A sentence is said to be *non-trivial* or *non-simple* if it possesses a node which is not the root.

Decomposable Sentence A non-trivial sentence is said to be decomposable if every non-root node is a compound node.

Now for the main concept of this entire article:

Grammar A grammar $\mathcal{G}$ is a collection of *ordered* sentences with the following properties:

1. every non-trivial sentence $W \in \mathcal{G}$ is decomposable.
2. Let $W, X \in \mathcal{G}$ be composable sentences, with Y their composition, then $Y \in \mathcal{G}$.
3. the ordering of every simple sentence S is a well-ordering.

In such a case, the members of a grammar will be called *formulas* (formulae if you want to be fancy).

The last point says that every simple sentence is a base. It may be easily shown that a graph isomorphism preserves the well-ordering of two bases (given that it preserves edges).

I must admit, I have taken great pains to ensure this definition includes the possibility of formulas possessing syntaxes of *infinite size*, and possibly *infinite length*, for if I did not, this theory would simply be a "step-child" or servant to computing science; not that there's anything wrong with that, but one of my intentions (to mention it again) is to keep this on par with any other branch of mathematics. To further that arguement, it could be argued that there is no need for a definition such as a grammar, that algebra with group and ring theory and the such handles the "philosophy" and language of math already... I say to that the great strength, but also the great weakness of the algebras is in their defining only a finite number of "operations" formally; the weakness is in the fact that in practice it is common to *imbue* an additional operation, which is a "hack"—it gets the job done, but does not follow the philosophy of the discipline. Lastly, I make note that such a definition is very much related or possibly equivalent to ones used in formal logic (including model theory), but the direction and style is much more mathematical.

3.3 The Representation of Formulas

With the really hard definitions over (though there's more to come), I explore and exploit them here with aim to develope and familiarise the reader with the notation.

The grammatical functions I spoke of earlier are—within the framework of this article—nothing more than collections of formulas of a grammar... which would be the reason for my calling it a grammar. This section explores their representation through the representation of their syntaxes.

Let's first look at a simple *traditional* formula:

$$B * (D + E)$$

when one looks at this, one sees three objects B, D, E acted upon by the operations $*, +$. The thing is, there is more information implied here which many students of math take for granted, there is the object C being $C = (D + E)$ as well as the object A defined as $A = B * C$.

Such relationships are shown as:

with

which—again, within the framework of this article—may be thought of as "basic composable formulas":

such that when actually composed, their composition looks like this:

The objective then is to translate the traditional formula:

$$B * (D + E)$$

into the new formula, with new notation.

For starters, let

$$F :\equiv B * (D + E)$$

where ':≡' is worded as "informationally equivalent" meaning F is our new formula, but also the reference point of its arguements. The secret is to break down this formula into its basic composable components—that of its bases—which is possible due to our definition of a grammar. To reference the A and the B from F using the *main base*, we write:

$$\begin{aligned} F/1 &= A \\ F/\mu &= B \end{aligned}$$

where μ is the root of the main base but also of the entire formula (μ is lowercase mu). The clever reader who is beginning to understand where I'm going with this might be tempted to say that

$$^{*}F/\mu^2 = C$$

but due to the way a compound is defined,[2] it is actually:

$$F/\mu^2 = (D + E)$$

thus, if one wants to *point* to C, or for that matter D or E individually, they would write:

$$\begin{aligned} F/\mu^2 1 &= C \\ F/\mu^2 \sigma &= D \\ F/\mu^2 \sigma^2 &= E \end{aligned}$$

Here σ is the root of the "subordinate" base (σ is lowercase sigma). In linguistics this concept of "pointing" would be something similar to *modification* (an example of a modifier is that of turning the verb "teach" into the noun "teacher").

As an example of how these names would be spoken, take the middle equation

$$F/\mu^2\sigma = D$$

one would say:

$$F \text{ over [mu squared] } \textit{pass} \text{ sigma...equals } D$$

The idea is that a path

$$p$$

is represented by its component edges

$$p_1 p_2 \dots p_n$$

which conveniently are the roots of bases and may be notated as such. If you throw a formula F into the picture, the notation takes on an additional form:

$$F/p$$

is equivalent to

$$F/p_1 p_2 \dots p_n$$

Reiterating all of this in the terminology of linguistics, the $B * C$ is a clause, and $D + E$ is also one. The subject of $B * C$ is the resultant number A, the objects, direct, indirect, oblique or whatever are B, C. The computation $D + E$ is an embedded clause, and from my knowledge of linguistic syntax, it meets the specifications defined by a "complement clause." In anycase, it's subject is the number C and it's objects are D, E. At this point one must be wondering, but there is infact a method to the madness of this linguistics analysis: I have broken this formula down as a means of transition to introduce the following paradigm:

$$\underbrace{A}_{\text{subject}} \quad \underbrace{\text{is Computed}}_{\text{verb}} \quad \underbrace{\text{from } B}_{\text{object}} \quad \underbrace{\text{using } C.}_{\text{other object}} \tag{1}$$

where the "computational verb" is "multiplied". Also

$$\underbrace{C}_{\text{subject}} \quad \underbrace{\text{is Computed}}_{\text{verb}} \quad \underbrace{\text{from } D}_{\text{object}} \quad \underbrace{\text{using } E.}_{\text{other object}} \tag{2}$$

where the computational verb is "added". This may be used as an informal tool to translate an *operator* (verb) into the devised notation.

The above is the first step in representing grammatical formulas. The second step is in displaying the tree structure of the syntax "all at once" as was done with bases in the polynomial ring example earier on. This is of course done in the same way, by concatenating the paths with plus (+) signs. Going back to our formula:

$$F :\equiv B * (D + E)$$

[2] by the way, the pre-superscript asterisk is a linguistic convention communicating that what follows doesn't actually occur in observation

one structurally denotes the syntax of F as:

$$\mathrm{Syn}(F) := 1 + \mu + \mu^2 1 + \mu^2\sigma + \mu^2\sigma^2$$

Here's the trick though, this form is kind of ugly, so one may "factorize" as such:

$$\mathrm{Syn}(F) := 1 + \mu + \mu^2(1 + \sigma + \sigma^2)$$

Finally though, even this can look ugly sometimes, so for display purposes, I sometimes use a less formal "factorization" (I will mention it but I don't recommend it):

$$\mathrm{Syn}(F) := (1 + \mu + \mu^2)(\mu^2 + \sigma + \sigma^2)$$

where the bases themselves are seperated by parentheses. Thus, looking at the rightside base, the μ^2 and σ and σ^2 all belong to the subordinate base. Furthermore, if standing alone, this rightside would be written as $(1 + \sigma + \sigma^2)$ but since it's "subject" is the second "object" of the first clause, I make reference to that fact by replacing it (its 1 that is) with μ^2. This, with respect to linguistics, I figure would be some form of subject or object agreement as opposed to disambiguity by word order.

I'll add another example of translating standard formulas into these new formulas (because examples are good):

The clause or formula g being informationally equivalent to:

$$(b + (cd))e$$

would be translated to have its syntax as:

$$\mathrm{Syn}(g) = 1 + \mu(1 + \sigma + \sigma^2(1 + \mu + \mu^2)) + \mu^2$$

which I must admit is comparatively long to convey the same structural information (about 3 times the length), but as with anything in language, frequently used "phrases" tend to evolve shorter forms; once the "whole" is established, the context is set and shortforms may be invoked, but I'll get to that later.

In anycase, the components broken down are:

$$\begin{aligned} g/1 &= (b + (cd))e \\ g/\mu &= b + (cd) \\ g/\mu^2 &= e \\ \\ g/\mu\sigma &= b \\ g/\mu\sigma^2 &= cd \\ \\ g/\mu\sigma^2\mu &= c \\ g/\mu\sigma^2\mu^2 &= d \end{aligned}$$

The strength does not entirely show itself here, because this formula is not informatically dense, but other examples with info-dense data environments (jungles) like "sums of sums of products" can better demonstrate.

The analogy is with algorithmics. Powerful algorithims that can do the same work as simpler ones tend to use more resources and are less efficient than their counterparts when computing small values; they only really show their true strength when heavier computations are required.

Finally, one may have noticed already I tend to use σ as the root of any formula of addition and μ as the root for any formula having to do with multiplication. I will use such conventions through the remainder.

3.3.1 formulas with deep embedding

Just to work with this notation a bit more, to build some understanding, I ask the question: What does one do when the pathname of a given formula becomes way too long?

$$f/\sigma\sigma\sigma^2\mu^4\sigma^2\sigma^3\mu\mu^2\sigma$$

its drawback is that it is no longer as user-friendly, though admittedly, if a formula structure is that complicated, you're kinda in trouble anyways. In situations like this, I use the arrow notation:

$$f/\!\!\rightarrow \sigma$$

where

$$\text{“}\rightarrow\text{”} = \sigma\sigma\sigma^2\mu^4\sigma^2\sigma^3\mu\mu^2$$

moreover, if you want to specify the number of branchpoints, you can do this:

$$f/\underset{8}{\rightarrow} \sigma$$

3.3.2 recursion

This example is a bit early as far as the definitions are concerned, not enough of them have been given for this to be rigourous (relative to this article), yet enough have been given for it to be understood, and I am very much aware of the need for examples.

The following recursive equation:

$$a_n = b_n + c_n a_{n-1}$$

given some initial value a_0 has a combinatorially "optimized" closed form which I will derive here:

The strategy that works for me, for finding closed forms, is to substitute (assuming one can) its predecessor into the equation:

$$a_{n-1} = b_{n-1} + c_{n-1} a_{n-2}$$

thus becoming:

$$a_n = b_n + c_n(b_{n-1} + c_{n-1} a_{n-2})$$

Following this, one then rearranges until one comes up with a form in which the main base is the same as the original, and in particular is such that the recursive object a_k occurs in the same place:

$$\begin{aligned} a_n &= b_n + c_n(b_{n-1} + c_{n-1} a_{n-2}) \\ &= b_n + (c_n b_{n-1} + c_n(c_{n-1} a_{n-2})) \\ &= b_n + (c_n b_{n-1} + (c_n c_{n-1}) a_{n-2}) \\ &= (b_n + c_n b_{n-1}) + (c_n c_{n-1}) a_{n-2} \end{aligned}$$

which is done in sequence using the distributive law, the associative law, and again the associative law.

One may have noticed I have not used my notation yet in these manipulations. This recursion is infact a good example to show both its strengths and weaknesses. We start with the weaknesses, in which the above manipulation, assuming it is referenced as f, would end up looking as such (take a brief look and then keep reading):

$$\begin{aligned} \mathrm{Syn}(f) &= 1[a_n] + \sigma[b_n] + \sigma^2(1 + \mu[c_n] + \mu^2[a_{n-1}]) \\ &\equiv 1[a_n] + \sigma[b_n] + \sigma^2(1 + \mu[c_n] + \mu^2(1[a_{n-1}] + \sigma[b_{n-1}] + \sigma^2(1 + \mu[c_{n-1}] + \mu^2[a_{n-2}]))) \\ &\equiv 1 + \sigma[b_n] + \sigma^2(1 + \sigma(1 + \mu[c_n] + \mu^2[b_{n-1}]) + \sigma^2(1 + \mu[c_n] + \mu^2(1 + \mu[c_{n-1}] + \mu^2[a_{n-2}]))) \\ &\equiv 1 + \sigma[b_n] + \sigma^2(1 + \sigma(1 + \mu[c_n] + \mu^2[b_{n-1}]) + \sigma^2(1 + \mu(1 + \mu[c_n] + \mu^2[c_{n-1}]) + \mu^2[a_{n-2}])) \\ &\equiv 1 + \sigma(1 + \sigma[b_n] + \sigma^2(1 + \mu[c_n] + \mu^2[b_{n-1}])) + \sigma^2(1 + \mu(1 + \mu[c_n] + \mu^2[c_{n-1}]) + \mu^2[a_{n-2}]) \end{aligned}$$

Such a sequence of manipulations has no place in good mathematics, the whole idea after all is to simplify the process, to organise the manipulations and make them less informationally confusing.

The trick then is to use the traditional *object-oriented* notation to get a clue as to the "rules" of manipulation (as above), then to verify this using the newer *verb-oriented* notation. The strength here is in being able to divide and conquer the verification. Another way of thinking about this is to break this recursive equation down into its recursive components to solve, just as one may find a derivative of a suitable multivariable function by finding its respective partial derivatives.

Going back to the formula f in the above, we first look at the object

$$\mu^2\sigma^2 \setminus f$$

which is infact:

$$a_{n-1}$$

though to accentuate the fact that I'm more interested in the object for my manipulation, I choose the form:

$$[a_{n-1}]_{\mu^2\sigma^2}$$

whereby I have removed the formula reference f (since the context is clear), and subscripted the path structure. To reiterate, this says that a_{n-1} is the second term of a product μ^2 which itself is the second term of a sum σ^2.

Although I've already defined this notation, I never really explained it. Here the square brackets [ ] represent closure, I make analogy to both calculus with its closed intervals, as well as ring theory which uses them for polynomial rings which are not fields—$\mathbb{Z}[x]$ as opposed to $\mathbb{Q}(x)$. By closure I mean that which is contained within is a either a leaf, or a base in which its arguements are all leaves. An example of this is

$$[5]_1 + [2]_\sigma + [3]_{\sigma^2}$$

which is the same as $5 = 2 + 3$.

One more example before I move on:

$$\text{Syn}(g) = 1 + \sigma + \sigma^2[1 + \mu + \mu^2]$$

one should not assume out of context that g/σ is an object, it might infact be a subordinate sentence, one of the strengths of this notation is in its ability to conceal information which isn't of immediate use. On the other hand, since the $1 + \mu + \mu^2$ is enclosed in square brackets as opposed to parenthesis, the convention is that its arguements are all objects. Finally, if a sentence is enclosed in parenthesis, I tend to let it mean either closed or open, but maybe I should reserve it strictly for open and use angle brackets $\langle\ \rangle$ for the ambiguous case...I haven't really stabilized that in my mind just yet.

To get back to it then, within our established formula, we first look at

$$[a_{m-1}]_{\mu^2\sigma^2}$$

for $1 < m \leq n$, which when applying the substitution as before becomes

$$[a_{m-2}]_{\mu^2\sigma^2\mu^2\sigma^2}$$

and again as before, we apply the distributive law providing us with

$$[a_{m-2}]_{\mu^2\mu^2\sigma^2\sigma^2}$$

Before continuing on, I will explain the logical thought process of what I just did to obtain this distributive form. The second and third edges in $\mu^2\sigma^2\mu^2\sigma^2$ (before the distribution) respectively are σ^2 followed by μ^2 which says to me, without even having to know the details of the rest of the sentence is that I have sum of two things which are part of a product of two more things. Thus I may distribute through leaving a sum of two things where each is a product. Since the second edge is σ^2 before the distribution, it means our object in question is the second object in the sum, and therefore retains its second place status within the sum afterwards. Finally, since the third edge is μ^2 before the distribution, it says the first object of the multiplication is the one that multiplies through, thus staying on the first or left side, further entailing that our a_{m-2} becomes the second object in the multiplication after the distribution resulting in:

$$[a_{m-2}]_{\mu^2\mu^2\sigma^2\sigma^2}$$

as above. This seems somewhat difficult a way of thinking at first, but it does get easier and even natural with practice, though I must admit it favours people with strong auditory manipulation skills. If you are not convinced, I should remind you after all, though you take it for granted now, when you were a child you had some difficulty adjusting to this distributive law when it was first introduced to you (I'm counting on the assumption that if you didn't find it difficult then, you shouldn't find this difficult now).

Next, we reassociate the products, since the second term of the first product is itself a product of two terms, we reassociate the $f/\mu^2\mu$ term to become $^\sim f/\mu\mu^2$ (the symbol $^\sim f$ is my way of acknowledging that the reference has changed, without actually changing it), that is, we changed the first term in the first product to a product, simplifying the second term to:

$$[a_{m-2}]_{\mu^2\sigma^2\sigma^2}$$

we apply this same thinking to the bases of addition ending up with

$$[a_{m-2}]_{\mu^2\sigma^2}$$

To reiterate what has been done, we started with

$$[a_{m-1}]_{\mu^2\sigma^2}$$

a formula of a sum who's second term is a product, and ended up again with a sum who's second term is a product

$$[a_{m-2}]_{\mu^2\sigma^2}$$

only this time our a_{m-1} object has become a_{m-2}. This is to say, we may substitute, distribute, and twice associate as many times as is required, and we will inevitably end up with a formula h to which we know the structural outline:

$$\mathrm{Syn}(h) = 1 + \sigma + \sigma^2(1 + \mu + \mu^2[a_0])$$

The focus then changes to finding out the structures and objects which comprise:

$$h/\sigma$$

and of

$$h/\sigma^2\mu$$

which is the easier of the two, and thus is the one with which we will begin.

We first define t_s to be equal to the object $\sigma^2\mu$ after the s^{th} sequence of manipulations, for example $t_0 = c_n$. We now seek t_{s+1} which is determined by applying the sequence of manipulations one more time:

$$\begin{aligned}
[t_s]_{\mu\sigma^2} &\subseteq (\ [t_s]_{\mu} + [c_{n-1-s}]_{\mu\sigma^2\mu^2}\ )_{\sigma^2} \\
&\supseteq (\ (\ [t_s]_{\mu} + [c_{n-1-s}]_{\mu\mu^2}\ )_{\sigma^2}\ )_{\sigma^2}
\end{aligned}$$

$$\begin{aligned}
(\ (\ [t_s]_{\mu} + [c_{n-1-s}]_{\mu\mu^2}\ )_{\sigma^2}\ )_{\sigma^2} &\equiv (\ (\ [t_s]_{\mu} + [c_{n-1-s}]_{\mu^2}\ )_{\mu\sigma^2}\ )_{\sigma^2} \\
&\equiv [t_s c_{n-1-s}]_{\mu\sigma^2\sigma^2} \\
&\equiv [t_s c_{n-1-s}]_{\mu\sigma^2} \\
&\equiv [t_{s+1}]_{\mu\sigma^2}
\end{aligned}$$

here $\subseteq$ means the leftside formula is "contained in" or "equivalent to" the rightside, which is appropriate for a substitution. Also the $\supseteq$ says the leftside or an equivalent contains the rightside. The remaining $\equiv$ is to indicate equivalence.

To summarize back into the regular notation, the above says:

$$t_{s+1} = t_s(c_{n-1-s})$$

This same type of trick may now be applied to determine h/σ. The thing is, I have no excuse to continue using this foreign verb-oriented notation of mine to finish the derivation. I didn't really even need it for this last step. The only time I really needed it was to verify the main base structure is "preserved" under the appropriate "substitution" "distribution" "association" "association" sequence. With that given, I simply could have done this to begin with:

$$a_n = u_s + t_s a_{n-1-s}$$

where $u_0 = b_n$, $t_0 = c_n$. With this, and the additional equation:

$$a_{n-1-s} = b_{n-1-s} + (c_{n-1-s})a_{n-2-s}$$

we have:

$$\begin{aligned}
a_n &= u_s + t_s(b_{n-1-s} + c_{n-1-s}a_{n-2-s}) \\
&= u_s + (t_s b_{n-1-s} + t_s(c_{n-1-s}a_{n-2-s})) \\
&= u_s + (t_s b_{n-1-s} + (t_s c_{n-1-s})a_{n-2-s}) \\
&= (u_s + t_s b_{n-1-s}) + (t_s c_{n-1-s})a_{n-2-s} \\
&= u_{s+1} + t_{s+1}a_{n-2-s}
\end{aligned}$$

thus, leaving us with:

$$t_{s+1} = (c_{n-1-s})t_s$$

and

$$u_{s+1} = (b_{n-1-s})t_s + u_s$$

These recurrences are much more easily solved. Starting with t_s, when expanded, one obtains the product:

$$t_s = \prod_{0 \leq j \leq s} c_{n-j}$$

and since $a_1 = b_1 + c_1 a_0$, this may be applied up to the point that the last term in the product (that of c_{n-s}) is no smaller than c_1, which to say $n - s = 1$, or $s = n - 1$ which determines our object:

$$t_{n-1} = \prod_{0 \leq j \leq n-1} c_{n-j}$$

Now, we turn our attention to the u_s, which expands into the sum:

$$u_s = \sum_{0 \leq k \leq s} (b_{n-k})t_{k-1}$$

Some might object to this as being unacceptable... except that

$$t_{-1} = 1$$

since an empty product is defined to be the multiplicative identity. Anycase, substituting $s = n - 1$ as was done for t_s, and expanding the t_{k-1} itself leaves us with:

$$u_{n-1} = \sum_{0 \leq k \leq n-1} b_{n-k} \prod_{0 \leq j \leq k-1} c_{n-j}$$

Finally, when we apply everything we've learned to:

$$a_n = u_{n-1} + t_{n-1} a_0$$

we derive our optimal closed form:

$$a_n = a_0 \prod_{0 \leq j \leq n-1} c_{n-j} + \sum_{0 \leq k \leq n-1} b_{n-k} \prod_{0 \leq j \leq k-1} c_{n-j}$$

where in the special case of $a_0 = b_0$ simplifies to

$$a_n = \sum_{0 \leq k \leq n} b_{n-k} \prod_{0 \leq j \leq k-1} c_{n-j}$$

although some might consider the following a little better:

$$a_n = \sum_{0 \leq k \leq n} b_k \prod_{k+1 \leq j \leq n} c_j$$

3.4 Globalization of Formulas

Formulas within a grammar themselves are useful, but a larger "perspective" in the long-run is the way to go, one does this by looking at the "product" of a collection of formulas, which makes sense if you realise that a collection of formulas may be thought of as a subset of an indexed system of sets.

Here we arrive at the second most important concept of this article, which is the first real step in "categorizing" grammatical functions.

Operator An operator F of a grammar $\mathcal{G}$ is a collection of formulas of $\mathcal{G}$ such that for any $f, f' \in F$, if $T_f, T_{f'}$ are the respective associated trees, then:

$$T_f = T_{f'}$$

A consequence of this is that the formulas of an operator share the same syntax. Thus it is a reasonable thing to say that the common syntax of the formulas is the syntax of an operator, written $\mathrm{Syn}(F)$. Furthermore, the union $\bigcup_{f\in F}\{f/p\}$ of arguements of some path p spanning the formulas of F, is said to be the p^{th} *perspective* of F. Furthermore, any such perspective given p will be written as:

$$F/p$$

which is to say

$$F/p := \bigcup_{f\in F}\{f/p\}$$

3.4.1 Morphisms between Operators

There are various relationships that span outwards starting will most similar to least when looking at operators, beginning with:

Operator Equality Two operators $(S, M), (T, N)$ are equal if $S = T$ and $M = N$.

Operator Isomorphism Let $F = (S, M), G = (T, N)$ be two operators with isomorphic syntaxes. If the following conditions are satisfied:

1. for any formula $f \in F$ there exists an isomorphic formula $g \in G$.
2. for any formula $g \in G$ there exists an isomorphic formula $f \in F$.

then the operators F and G are said to be isomorphic. A formula $g \in G$ that is isomorphic to some formula $f \in F$ is said to be the isomorphic complement of f, but it may also be said that they are adjacent.

Isomorphic operators mostly show up as suboperators.

Bimorphism Let $F = (S, M), G = (T, N)$ be two operators with isomorphic syntaxes. Let $i : S \to T$ be such an isomorphism. For every path $p \in \mathrm{Syn}(F)$, if $F/p = G/i(p)$ then the operators F and G are said to be bimorphic.

This says two operators have the same syntaxes and the same perspectives, which is slightly weaker than an isomorphism of operators.

Operator Homomorphism Two operators F, G are homomorphic if their syntaxes are isomorphic.

3.4.2 operators with several perspectives

As stated earlier, I've been using σ to reference addition, though admittedly that was a bit of a cheat since we haven't even reached the section dealing with the proper foundations of such an operation. Regardless, it is still worthwhile to give examples, so I will go on using it anyways.

Let's say we have an operator F corresponding to addition:

$$\mathrm{Syn}(F) = 1 + \sigma + \sigma^2$$

where each formula is an instance of the operation: If $f \in F$ where $f/\sigma = 2$ and $f/\sigma^2 = 3$, then $f/1$ must equal 5 which is to say that $5 = 2 + 3$ (though I've used this example already).

If this is the case, then what is the interpretation of?

$$\mathrm{Syn}(G) = 1 + \sigma + \sigma^2 + \sigma^3 + \sigma^4 + \sigma^5 + \sigma^6$$

assuming G is an operator that is a generalisation of F.

Luckily, along with multiplication as well as many of the operators of interest, the associative law holds for addition and therefore the interpretation is in practice never ambiguous. But, from a definition standpoint, this simply means each formula of G is "a sum with 6 non-identity arguements", where the numbers (arguements) of this particular formula are currently hidden from anyone looking to interpret them. I suppose if I wanted to be very thorough, when applying this to actual operations, I would have to prove the associative law from scratch. I would start with formulas whose non-identity arguements are all occurences of the same value, such formulas are trivially associative... but... I'm not going to do that. I will however append to the already made notation, to account for

operators with many perspectives. The degree of some simple formula f is the highest order arguement (analogous to polynomials); it is written with the "double-bar" notation, with the previous example this is:

$$f/|\sigma| = 6$$

Next, if one is working with a formula h such that:

$$h/|\mu| = 30$$

then... for starters, it is implied the root operation (if not only operation) is a long string of multiplications, in particular there are 30 numbers multiplied together. In a situation like this, one gets the opportunity to point to more than one arguement at a time, for example if:

$$\begin{aligned} h/\mu^{12} &= 7 \\ h/\mu^{13} &= 3 \end{aligned}$$

then

$$h/(\mu^{12} + \mu^{13}) = 7 \times 3$$

3.5 Combinatorial Functions

Mathematical functions which may be represented by a *grammatical operator* will be called *combinatorial functions*.

Combinatorial Pair Let f be a formula with a bridge e. The ordered pair (n, i) is called the combinatorial pair generated by e if n, i are the subformulas generated by removing e.

There is a lot of math that uses a formulaic representation of a function up until the point when things get tough (contradictory in nature), when that happens it falls back on the set theoretic definition. This is allowed because the algebra of the formula allows for a proof of extension, a proof that one has been working with the same function in both cases. The framework of this article takes the following approach.

Combinatorial Relation Let F be an operator and e an edge of its tree. Define the relation C_F called the combinatorial relation of F to be such that it contains only the combinatorial pairs of the formulas $f \in F$ generated by the edge e.

Combinatorial Function An operator F is called a combinatorial function if its combinatorial relation is a set-theoretic function.

Identity Let F be a combinatorial function. For any formula $f \in F$, with (n_f, i_f) its combinatorial pair, the second coordinate and subformula i_f is called the identity of f with respect to F. In such a case the identity is written as $\mathrm{id}(f)$. Furthermore, the collection of all such identities is written as $\mathrm{id}(F)$.

This definition is motivated by the notion of the "subject" of a sentence. In terms of computing, this would translate to the output, or return value or whatever of the computation. In practice, the identity is more often than not a subformula with path 1 and arguement $f/1$, in which case the path 1 is called the *identity path* while the arguement $f/1$ is called the *identity arguement*. Furthermore, it is usually the case that one is only interested in the identity arguement, and thus, when the context is clear, will simply be refered to as the identity.

Formula manipulations as with most "actions" within math are best defined descriptively as relationships between formulas (as opposed to prescriptively as rules):

Formula Allomorphism Two combinatorial formulas f and g are said to be allomorphic if their identities are isomorphic.

a *combinatorial formula* is one that is part of a combinatorial function. This of course generalises:

Operator Allomorphism Two combinatorial functions F and G are said to be allomorphic if:

1. for all $f \in \mathrm{id}(F)$ there exists an allomorphic formula $g \in \mathrm{id}(G)$.

2. for all $g \in \mathrm{id}(G)$ there exists an allomorphic formula $f \in \mathrm{id}(F)$.

Here we arrive at the third most important concept of this article:

Rule Let F, G be allomorphic operators. Let $R \subseteq \mathrm{Syn}(F) \times \mathrm{Syn}(G)$ be a relation between the syntaxes of the two operators. If every allomorphic pair $f \in F, g \in G$ is such that for any $(p, q) \in R$ it follows that

$$f/p = g/q$$

then the relation R is said to be a rule from F to G. In such a case the rule R will be written $F\langle G\rangle$.

It should be noted that if H is some operator isomorphic to G, then some *equivalent* rule exists from F to H. Furthermore, rules form an equivalence relation on operators of a grammar.

3.5.1 an example of a generic clause

The following is a visualization of the clause structure of a 3×3 matrix:

As far as the expanations go, I've thus far explained a lot, and in pretty great detail. So I'm leaving this one for the reader to fill in the blanks. I'll give one clue though, the branches of the "syntax" are intersections of the horizontal and vertical paths. This standard representation again has everything to do with combinatorics, the question then is, can it be proven as optimal?

3.5.2 dummy variables

This is a bit of a detour, it's a "bone to pick" so to speak regarding the concept of a dummy variable. Looking at many of the successes of defining concepts within mathematics, one realises they've been made by *describing* a structure rather than *prescribing* one, a notion I hinted upon earlier. Now, if a theory is big enough, for the purpose of human learning, concepts and terminology need to be standardised. That is to say one can never (or vary rarely) get rid of prescribing altogether; the next best thing then is to streamline it or minimise it.

In anycase, one could argue the notion of a "rule" is just a conceptual swapping with that of a "dummy variable", and possibly even more prescribing has actually been done. I disagree, not that I'll argue the point with some clever retort, frankly I don't care that much. I will say that if you give it a bit of thought, you should be able to think of some good reasons why this is a better thing.

3.5.3 combinatorial generating functions

The way to approach combinatorial power series is the same as with the formal power series of rings, that of using an "action map". The difference is the initial mapping, the formula, which differs by mapping to some other kind of combinatorial set (as opposed to a ring).

For example, if you needed to build a team of 3 problem solving mathematicians, and you had 2 available from the University of Alberta (maybe they owed you a favour), and 3 from the University of British Columbia, how many ways are there to build such a team? That's not all though, one further restriction is that 2 of your available mathematicians from UBC refuse to work with each other. Now, one uses an operator to map the possible choices for each university. For UofA:

$$\mathrm{Syn}(f) = 1 + x + x^2 \quad \text{where } f/x^k = \text{ a choice of } k \text{ mathematicians}$$

given this, you can use an action map to make an equivalence class:

$$[f] = 1 + 2x + x^2$$

where the coefficients here are the total number of choices of a collection of k mathematicians given x^k. Also, lets say for the university of UBC the polynomial ends up being:

$$[f] = 1 + 3x + 2x^2 + 0x^3$$

The combinatorial solution to determine all possibilities would be to multiply

$$(1 + 2x + x^2)(1 + 3x + 2x^2)$$

and if one does so, they obtain

$$(1+2x+x^2)(1+3x+2x^2) = 1+5x+9x^2+7x^3+2x^4$$

It should be stated that DeMorgan's laws of set union and intersection need to be taken into account, but this is not an introduction to applied combinatorics: thus there are 7 (x^3) ways to build your team.

Up to this point I've been using ϕ^0 for the identity of an operator with root ϕ. For the case of generating functions though, it would be prudent to have the identity as another power, to free up the 0. The way to achieve this is to use ω, which is the first or smallest infinite ordinal (i.e. it is the set of natural numbers themselves), in terms of addition, with ordinal numbers, one may do two things: $\omega + 1$ or $1 + \omega$, where $\omega + 1 = \{0, 1, 2, 3, \ldots, \omega\}$ and $1 + \omega = \{\omega, 0, 1, 2, 3, \ldots\}$. The difference is in the order of ω with respect to the remaining. In the second case this is to say $\omega < n$, $n = 0, 1, 2, 3, \ldots$ this is a good idea for at least two reasons: Firstly, it frees up 0, but secondly, it retains an intuitive sense when looking at infinite power series, but I leave that for a later day.

4 Representation

Back in the day, the notion of a "function" was at a stage such that it's definition was that of "a rule that maps each value 'x' to exactly one value 'y'". This definition over time changed out of necessity, which happened for a few reasons (not to give a history lesson). Two of these reasons notably being that it firstly was not precise, and secondly was not sufficient. In anycase, the suffiency of it is a problem that can't be fixed, it is what it is, which is something that is not applicable to many of the functions mathematicians find fascinating. Regardless of this matter, the precision, the "what is a rule?", can now be dealt with in the scope of this grammar theory.

This precision begins with a representative:

Representative Let x be a combinatorial function. An operator f is said to be a representative of x if there is exists a rule between the two. In which case such a relationship of representation is written $f(x)$.

Notice this definition can possibly be symmetric, but in practice one uses the more "complex" operator to represent the simpler one.

This doesn't mean much without examples, so let

$$\mathrm{Syn}(x) := 1 + \phi$$

be the combinatorial operator such that for any formula $t \in x$ (by the way ϕ is lowercase phi), the following property holds:

$$\mathrm{id}(t) = (t/\phi)^2$$

this is to say x is equivalent to a set theoretical function defined to as: $\{(a, a^2) \mid a \in G\}$ given some algebraic set G of course. Now let

$$\mathrm{Syn}(f) = 1 + \mu + \mu^2$$

be a representative of x. Then there must exist a rule between these allomorphic operators. Take

$$\{(1,1),\ (\phi,\mu),\ (\phi,\mu^2)\}$$

In such a case, this is all that is needed, but what does this really mean?

For starters, the $f(x)$ notation may be written as:

$$\begin{aligned} f(x) &= (1+\mu+\mu^2)(x) \\ &= 1(x) + \mu(x) + \mu^2(x) \end{aligned}$$

But lets say we want to get more specific than that, what if the arguements $f/1,\ f/\mu,\ f/\mu^2$ are leaves, endpoints, input (in this case they are), then the previous may be conventionally denoted as:

$$f(x) = 1[x] + \mu[x] + \mu^2[x]$$

which is simply a subtle change from parenthesis to square brackets; I remind the reader this is an analogy to calculus with the (open) and [closed] interval notations, which again, is further used in ring theory for fields $\mathbb{Q}(\sqrt{2})$ as opposed to (strictly) integral domains $\mathbb{Z}[i]$. Anyways, since μ is multiplication, a more compact form of this is:

$$f(x) = x^2$$

though in the scope of this theory, such traditional notational writings are object-oriented shortform.

The following subsubsections more or less translate some of the basic rules mathematical manipulations are based on. I will note I have intentionally presented the following in an auditory-friendly way: What I mean by that is the relations are defined with the structural form of this grammatical notation, which when one is trying to understand the meaning, sounding it out is easier to get the bigger picture. The thing is, tree diagrams work wonders for this sort of thing visual, mathematicians largely are visual creatures after all, so I will give an example of such a bit later, but one should also get practice to build auditory skills.

4.0.1 commutation

The commutative law is the easiest one, so I start with it. Staying within the realm of addition and multiplication, let $F, c \in \mathcal{G}$ for some grammar $\mathcal{G}$, with

$$\text{Syn}(F) = 1 + \sigma + \sigma^2 \quad \text{and} \quad \text{Syn}(c) = 1 + \sigma + \sigma^2$$

then the commutative property holds between them if:

$$c\langle F\rangle = \{(1,1), (\sigma, \sigma^2), (\sigma^2, \sigma)\}$$

but this isn't a very flattering view of things, certainly not in the long run. The solution to this problem of aesthetics takes some lead in though.

For starters, since $c(F)$ simply means "c currently representing F", it may be written as:

$$c(F) := \begin{cases} F/1 & \text{for} \quad c(F)/1 \\ F/\sigma^2 & \text{for} \quad c(F)/\sigma \\ F/\sigma & \text{for} \quad c(F)/\sigma^2 \end{cases}$$

but this isn't sufficient for our purposes, it is idea to do the same thing with the $c\langle F\rangle$ notation:

$$c\langle F\rangle := \begin{cases} F/1 & \text{for} \quad c(F)/1 \\ F/\sigma^2 & \text{for} \quad c(F)/\sigma \\ F/\sigma & \text{for} \quad c(F)/\sigma^2 \end{cases}$$

this may seem trivial, but as an example of what this means, one may now display the following:

$$c\langle F/\sigma\rangle = c(F)/\sigma^2$$

This is notationally similar to that of a function, which will come in handy later on. I should add, though it looks similar, one must not forget this is a *relation*, the main point being the corresponding values may not be unique. Nevertheless, in the following I will exclude the case of the identity because part of the definition of a rule is that the operators are allomorphic, there's no need to explicitly state the identities are equal.

4.0.2 association

Let

$$\text{Syn}(F) = 1 + \mu + \mu^2(1 + \mu + \mu^2) \quad \text{and} \quad \text{Syn}(a) = 1 + \mu(1 + \mu + \mu^2) + \mu^2$$

then the associative property is described as:

$$a\langle F\rangle := \begin{cases} F/\mu^2\mu^2 & \text{for} \quad a(F)/\mu^2 \\ F/\mu^2\mu & \text{for} \quad a(F)/\mu\mu^2 \\ F/\mu & \text{for} \quad a(F)/\mu\mu \end{cases}$$

notice that any path in single form $\mu,\ \mu^2$ becomes double $\mu\mu,\ \mu^2\mu^2$ (symmetry applies here also, anything double becomes single), and the asymmetric form inverts $\mu\mu^2 \to \mu^2\mu$.

4.0.3 distribution

I've explained the two previous in terms of addition and multiplication, because they're familiar, but the whole point is to take what were theorems and turn them into definitions, given this, let:

$$\mathrm{Syn}(F) = 1 + \theta + \theta^2(1 + \omega + \omega^2)$$

and

$$\mathrm{Syn}(d) = 1 + (\omega + \omega^2)(1 + \theta + \theta^2)$$

Here θ is some arbitrary operator (θ is lowercase theta), it is analogous to multiplication, and ω is more-or-less suppose to be addition (ω is lowercase omega).

The rule for the distributive property is defined as:

$$d\langle F\rangle := \begin{cases} F/\theta^2\omega^2 & \text{for} \quad d(F)/\omega^2\theta^2 \\ F/\theta^2\omega & \text{for} \quad d(F)/\omega\theta^2 \\ F/\theta & \text{for} \quad d(F)/\omega\theta \\ F/\theta & \text{for} \quad d(F)/\omega^2\theta \end{cases}$$

With actual addition and multiplication this is:

$$d\langle F\rangle := \begin{cases} F/\mu^2\sigma^2 & \text{for} \quad d(F)/\sigma^2\mu^2 \\ F/\mu^2\sigma & \text{for} \quad d(F)/\sigma\mu^2 \\ F/\mu & \text{for} \quad d(F)/\sigma\mu \\ F/\mu & \text{for} \quad d(F)/\sigma^2\mu \end{cases}$$

4.0.4 differentiation

Differentiation does tend to be a bit more complicated, so in this situation I will at the end present a tree diagram for contrast with formulas.

There are a few different laws: the first, using the mainstream style notation, would be:

$$D(f + g) = D(f) + D(g)$$

I use D for differentiation, I know $\frac{d}{dx}$ is the mainstream ideal, but I find it gets in the way:

$$D\langle F\rangle := \begin{cases} F/\sigma & \text{for} \quad D(F)/\sigma\delta \\ F/\sigma^2 & \text{for} \quad D(F)/\sigma^2\delta \end{cases}$$

Here I use δ as the derivative operator (δ is lowercase delta).

Next:

$$D(fg) = D(f)g + fD(g)$$

which translates to:

$$D\langle F\rangle := \begin{cases} F/\mu & \text{for} \quad D(F)/\sigma^2\mu \\ F/\mu^2 & \text{for} \quad D(F)/\sigma\mu^2 \\ F/\mu & \text{for} \quad D(F)/\sigma\mu\delta \\ F/\mu^2 & \text{for} \quad D(F)/\sigma^2\mu^2\delta \end{cases}$$

which finally, with the tree diagram, is also known as:

Lastly:

$$D(f \circ g) = D(g)D(f) \circ g$$

which becomes:

$$D\langle F\rangle := \begin{cases} F/\kappa & \text{for} \quad D(F)/\mu^2\kappa\delta \\ F/\kappa^2 & \text{for} \quad D(F)/\mu\delta \\ F/\kappa^2 & \text{for} \quad D(F)/\mu^2\kappa^2 \end{cases}$$

Here I use κ to be the composition operator (κ is lowercase kappa).

4.1 power operators

One thing I haven't really talked about so far is "Sigma" (Σ) and related notations. For example:

$$\sum_{1\leq k\leq n} k = 1+2+3+\ldots+n$$

or

$$\prod_{1\leq k\leq n} k = 1\cdot 2\cdot 3\cdot \ldots \cdot n$$

such things I call *macro*s, they are "power operators", which is to say they are operators who's perspectives themselves are operators. In computing, a related concept would be a Universal Turing Machine.

Power operators are generalizations of a special kind of operator, which possess *global* properties of association, and usually commutation. Addition is the obvious example:

$$1+(2+3) = 1+2+3 = (1+2)+3$$

It has an absorption quality described by the middle formula, which has no left or right parenthesis. It doesn't stop there though, one may add as many terms as one likes, and continue to simplify by removing the parentheses. This is what is meant by *Globally Associative*. A similar concept applies when a formula is commutative, which is to say it is *Globally Commutative*.

Though I should give formal definitions, I will leave these concepts as intuitive at this time, I haven't worked with enough examples to be confident of a particular version.

4.2 recursive operators

The following recursive equation was given early as an example:

$$a_n = b_n + c_n a_{n-1}$$

at the time, I stated that it is in need of a proper interpretation before one can properly handle it. Hence the following definition:

Recursive Operator Let F be a non-empty combinatorial function. If there exists a path $p \in \mathrm{Syn}(F)$ such that for every $f \in F$ the following holds:

$$\text{there exists an } f' \in F \text{ with } f'/p = f/1$$

then F is said to be recursive.

Actually this definition probably needs some tweaking, but I'm kinda lazy at this point into the article, and what has been stated is sufficient.

4.3 The Structor Concept

Structor Space An operator F is called a structor space if its formulas can be treated as vectors to form a vector space, in which case the formulas will be called structors.

This is to say that the arguements of the formulas may be operated on componentwise, which is acceptable given they all share the same syntax of the operator. As a side note, I have blended the words "structure" and "vector" to get "structor". In intuitive terms, a formula of an operator would be the vector, it does have a component-wise structure after all. Furthermore, the operator to which the formula belongs would be a *structor space*.

This isn't the whole story though, not all operators of a given grammar appreciate becoming vector spaces, they're difficult and otherwise not worth the effort. Part of the reason for this is that they do infact want to be of use, but they'd much prefer to do it wearing the outfit of another algebra like a group or a ring.

In anycase, having an actual algebra of formulas becomes a very interesting thing. Take the Stirling Numbers of the second kind for example:

$$\left\{ n \atop m \right\}$$

which is defined to be a counting number to represent the number of ways to partition a set of n elements into m subsets, given that $0 \leq m \leq n$.

Let T be an operator with formulas of the form:

$$\mathrm{Syn}(T) = 1 + \theta + \theta^2$$

If all formulas are such that

$$\begin{aligned} T/1 &= \left\{ {n \atop m} \right\} \\ T/\theta &= n \\ T/\theta^2 &= m \end{aligned}$$

a "natural definition" of component-wise addition is unfortunately not particularily reasonable to make, given the restrictions on n, m. Here's where the *but* comes in: If the operator possesses more formulas than just those that correspond to the Stirling numbers, as example the operator such that $T/1,\ T/\theta,\ T/\theta^2$ varies over all natural numbers, then it's easy. The key to the usefulness of such a thing is the operator of *only* Stirling numbers that we're interested in is still a subset of this extension, and more importantly, is such that only one value of $T/1$ corresponds to each value of $T/\theta,\ T/\theta^2$, that is, it's combinatorial.

Take the two Stirling numbers:

$$\left\{ {4 \atop 3} \right\} = 6 \quad \text{and} \quad \left\{ {5 \atop 2} \right\} = 15$$

Then

$$21 = 6 + 15 = \left\{ {4 \atop 3} \right\} + \left\{ {5 \atop 2} \right\} \neq \left\{ {4+5 \atop 3+2} \right\} = \left\{ {9 \atop 5} \right\} = 6951$$

The trick is to say: let S be a combinatorial suboperator of T, if we have some formula $f \in S$, we can always find a unique formula $g \in T$ such that $f + g \in S$ under the component-wise addition. Back to the Stirling numbers example:

$$\text{If} \qquad \left\{ {4 \atop 3} \right\} = 6 \quad \text{corresponds to} \quad \left\{ {6 \atop 4 \atop 3} \right\}$$

this of course is the structor, but if this is the case, then one can easily find another Stirling structor to complete the job:

$$\left\{ {6 \atop 4 \atop 3} \right\} + \left\{ {6945 \atop 5 \atop 2} \right\} = \left\{ {6951 \atop 9 \atop 5} \right\}$$

where

$$\left\{ {6951 \atop 9 \atop 5} \right\} \quad \text{corresponds to} \quad \left\{ {9 \atop 5} \right\} = 6951$$

What's the point of all this you ask? the point is there are many circumstances in which one may treat formula's as structors yet not explicitly state the value of one or some of the formula's arguements. Such a thing is less time consuming and less stressful in having to worry about boundary constraints in applications; this is leading to the notation I am now to introduce:

4.3.1 structor notation

Take

$$\mathrm{Syn}(\boldsymbol{f}), \mathrm{Syn}(\boldsymbol{g}) = 1 + \sigma + \sigma^2$$

define $\boldsymbol{f} + \boldsymbol{g}$ to be such that

$$\mathrm{Syn}(\boldsymbol{f} + \boldsymbol{g}) = 1 + \sigma + \sigma^2$$

$$(\boldsymbol{f} + \boldsymbol{g})/\sigma^j = \boldsymbol{f}/\sigma^j + \boldsymbol{g}/\sigma^j$$

with $0 \leq j \leq 2$. First of all, I am actually mixing notations here in this last equation, the '+' sign on the left is a grammatical concatenation while the one on the right is an algebraic operation. Truthfully, the mixing of notations will be the case for the remainder, I do this because it does not provide ambiguity so as long as I use the

'/' to modify the operator into an object. Next, this example actually works out "naturally" in its definition of the addition of full formulas, I use it to point out that sometimes things are infact quite simple.

Now, notation-wise, to emphasise (or disambiguate) from non-structor formulas, instead of using the '/' symbol, I flip it around: '\'. For me, instead of saying "f divides", the notation says that whatever is on the rightside of '$f\backslash\ldots$' is such that it "extends f", is "above f", is "more than f", is "at the very least all of f".

For the record, this notation is stabilised in my mind just yet.

I could explain my way into the structor notation, but I'll go straight to an example, let

$$\mathrm{Syn}(F) = 1 + \sigma + \sigma^2 + \sigma^3$$

be an operator defining a structor space with the natural component-wise addition, then with:

$$f\backslash\boldsymbol{\sigma} - 3|\boldsymbol{\sigma^3} + 2|$$

one would say this is a formula g from F such that

$$g/\sigma = f/\sigma - 3$$

but also

$$g/\sigma^3 = f/\sigma^3 + 2$$

The '|' bars seperate these facts. Also, here, the f is the "neutral element" of this algebra, a starting point (a reference). I should finally add that nothing is actually said about the remaining arguements of g, that of $g/1$, g/σ^2, yes, in most situations this is ambiguous,[3] but as I pointed out above with an operator containing the Stirling Numbers, such ambiguity at times may be removed. Truthfully the safe way to play it is to assume by default that the notation represents all relationships between f and g unless otherwise specified (for example, stating beforehand that f belongs to a combinatorial suboperator). As an auxillary to all of this, if anyone is wondering, I make use of boldtype because it's fun.

A strange consequence of this way of notating is:

$$f\backslash\boldsymbol{\sigma^3} + 2|\boldsymbol{\sigma} - 3|$$

means the exact same thing, one need not worry about "word order" because the pathnames give reference. An advantage of this fact is:

$$f \backslash \left| \begin{array}{c} \boldsymbol{\sigma} - 3 \\ \boldsymbol{\sigma^3} + 2 \end{array} \right|$$

which is another formula, is such that it again means the same thing. Here an extra | at the beginning is added only because it makes things look cleaner. If one doesn't have either enough horizontal or vertical spacing one could make this notation to look like a matrix I suppose, it probably won't work perfectly though.

4.3.2 arithmetic series

With this next example, I'm pretty much going over the long established stuff, but as always, it's good to go over what's new with what's familiar when possible. Arithmetic series are the most obvious first choice:

$$1 + 2 + 3 + \ldots + (n-1) + n = \frac{1}{2}n(n+1)$$

To begin the proof, let:

$$\mathrm{Syn}(F) = 1 + \sigma + \ldots + \sigma^n \qquad \text{such that} \qquad F/\sigma^k = k \quad \text{with} \quad k \neq 0$$

but also, let

$$\mathrm{Syn}(p) = 1 + \sigma + \ldots + \sigma^n$$

be a representative $p(F)$ called "the inverted sum" of F, such that:

$$p/\sigma^k = F/\sigma^{n+1-k} = n + 1 - k \quad \text{with} \quad k \neq 0$$

[3] To be fair, ambiguity already exists within mathematical notation, for example $(2, \pi)$ is this an ordered pair? or an open interval of real numbers?

from the structor point of view:

$$\begin{aligned} 2(F/1) &= (2F)/1 \\ &= (F+F)/1 \\ &= (F+p(F))/1 \end{aligned}$$

but

$$\begin{aligned} (F+p(F))/\sigma^k &= F/\sigma^k + p(F)/\sigma^k \\ &= k + (n+1-k) \\ &= n+1 \end{aligned}$$

thus

$$\begin{aligned} F/1 &= \frac{1}{2}(F+p(F))/1 \\ &= \frac{1}{2}(F+p(F))/(\sigma + \ldots + \sigma^n) \\ &= \frac{1}{2}[(n+1) + \ldots + (n+1)] \\ &= \frac{1}{2}n(n+1) \end{aligned}$$

Okay Okay, the applications thus far have been pretty lame, with either a lack of rigour or triviality, but finally, I am ready to introduce something a bit better.

5 A Final Example

If mathematics allows for formal structures (power series), it should also allow for formal manipulations.

Time and again I have seen the "exact" same proof repeated to me though it was obvious the first time around, that of the binomial theorem.

By having seen it more than once, I mean:

$$(a+b)^n = \sum_{0 \le k \le n} \binom{n}{k} a^{n-k} b^k \tag{3}$$

$$D^n(fg) = \sum_{0 \le k \le n} \binom{n}{k} D^{n-k}(f) D^k(g) \tag{4}$$

$$(x+y)^{\underline{n}} = \sum_{0 \le k \le n} \binom{n}{k} x^{\underline{n-k}}\, y^{\underline{k}} \tag{5}$$

Here, D^n is the n^{th} derivative, and $x^{\underline{n}}$ is a "falling factorial power", which is to say

$$x^{\underline{n}} = (x-0)(x-1)(x-2)(\ldots)(x-[n-1])$$

It's a product of n things starting at x and subtracting 1 each time.

In anycase, the mechanical manipulation aspect of the proof is the same over every time, and that is the aim of this example. In particular, I will use the combinatorial style of proof which is my favourite, and is intuitive in this framework given the notion of paths of a syntax. I will also assume a commutative law exists, it's not much of a binomial theorem without it.

The first thing to do, would be to translate the left side of these equations into a more appropriate form starting with $(a+b)^n$. Let's simplify: a^n, this is "the identity exponentiated from a using n. Thus

$$\begin{aligned} \text{Syn}(f) &= 1 + \epsilon + \epsilon^2 \\ f/1 &= a^n \\ f/\epsilon &= a \\ f/\epsilon^2 &= n \end{aligned}$$

Obviously here I use ϵ as exponentiation (ϵ is lowercase epsilon).

Translating further reveals:

$$\begin{aligned} \mathrm{Syn}(f) &= (1+\epsilon+\epsilon^2)(\epsilon+\sigma+\sigma^2) \\ f/1 &= (a+b)^n \\ f/\epsilon &= a+b \\ f/\epsilon^2 &= n \\ \\ f/\epsilon\sigma &= a \\ f/\epsilon\sigma^2 &= b \end{aligned}$$

The same may be done with differentiation:

$$\mathrm{Syn}(g) = (1+\delta+\delta^2)(\delta+\mu+\mu^2)$$

as well as with "falling exponentiation":

$$\mathrm{Syn}(h) = (1+\varepsilon+\varepsilon^2)(\varepsilon+\sigma+\sigma^2)$$

Here's where things get tough. If one knows what the right side of the equation looks like, no problem, but if such knowledge is withheld in advance, one has to guess and check. This leads back to the "layercake" approach used with recursion, for example:

$$\begin{aligned} (a+b)^n &= (a+b)(a+b)^{n-1} \\ &= (a+b)(a+b)(a+b)^{n-2} \\ &= (a^2+2ab+b^2)(a+b)^{n-2} \\ &= (a^2+2ab+b^2)(a+b)(a+b)^{n-3} \\ &= (a^3+3a^2b+3ab^2+b^3)(a+b)^{n-3} \end{aligned}$$

In working out these few examples one first looks at the first parenthesis and notices the operator structure:

$$\begin{aligned} a+b &:\equiv 1+\sigma+\sigma^2 \\ a^2+2ab+b^2 &\equiv 1+\sigma+\sigma^2+\sigma^3 \\ a^3+3a^2b+3ab^2+b^3 &\equiv 1+\sigma+\sigma^2+\sigma^3+\sigma^4 \end{aligned}$$

Given that the manipulation process is the same each time, it's easy to prove with mathematical induction that after n recursions, one ends up with a left side operator as a sum of degree $n+1$. This is the first layer of this cake that we are baking. It's just as easy to prove the right side of the outer "product" reduces to $(a+b)^0$ after n recursions, leaving us with a nice and simple sum.

If I were proving only this one binomial theorem, I would now determine the character of each addend σ^k. This is not the general proof though, the thing is each of these three binomial theorems have their own little quirks, so one must look at the bigger picture and abstract this process.

γ is lowercase gamma:

Lemma 5.1 (Binomial Lemma) *Let F be a structor space with*

$$\mathit{Syn}(F) = 1+(\beta+\beta^2)(1+\alpha+\alpha^2)$$

Let G be a globally associative and commutative operator with

$$\mathit{Syn}(G) = 1+\gamma+\gamma^2$$

If there exists a rule r of F such that for each $f \in F$ the representative of f under r is some $g \in G$ with

$$\begin{aligned} g/\gamma &= f\backslash\beta\alpha^2+1| \\ g/\gamma^2 &= f\backslash\beta^2\alpha^2+1| \end{aligned}$$

then

$$r^n(f) = 1 + \gamma + \gamma^2 + \ldots + \gamma^{n+1}$$

where

$$r^n(f)/\gamma^{k+1} = \binom{n}{k} f\backslash \beta\alpha^2 + n - k | \beta^2\alpha^2 + k| \quad \textit{for } 0 \leq k < n$$

which is to say that $r^n(f)$ possesses a binomial representation.

Proof Let $f \in F$, the representative of f under r is such that $\mathrm{Syn}(r(f)) = 1 + \gamma + \gamma^2$. Now take

$$r(f)/\gamma^k \quad \text{where } k = 1, 2$$

this is equivalent to $f\backslash \beta^k\alpha^2 + 1|$, but then $r(f)/\gamma^k \in F$, thus $r(r(f)) = r^2(f)$ is such that we end up with a $1 + \gamma + \gamma^2 + \gamma^3 + \gamma^4$ allomorphic formula, since (γ) is globally associative. This process may be repeated ending up with a $1 + \gamma + \ldots + \gamma^{2^n}$ formula. Now it's a matter of grouping the terms.

Everytime one applies an r representative, one increases either $\beta\alpha^2$ by one, or increases $\beta^2\alpha^2$ by one. After n such increments, we end up with a structor of the form

$$f\backslash\ \beta\alpha^2 + n - k \mid \beta^2\alpha^2 + k|$$

There are $\binom{n}{k}$ paths to take to end up with such a term, thus, one may collect the terms together, hence, the binomial lemma. ∎

The three binomial theorems displayed at the begining of this subsection follow as special cases once their appropriate rules are determined.

6 Conclusion

I have to admit, there is so much more I wanted to add, especially example-wise, but doing so would at least double the length of this article. I have a whole other article I've written (though incomplete) regarding multivariable sums. I wrote it long ago, before this one, which could easily be revised and rewritten in the scope of this article.

Furthermore, there is the so called "Operator Calculus", which either isn't considered rigourous (or the level of rigour isn't worth it for most), but allows for a fascinating treatment of manipulations including:

$$\Delta + I = e^D$$

where

$$\begin{aligned} \Delta f(k) &= f(k+1) - f(k) \\ If(k) &= f(k) \\ e^D f(k) &= (I + D + \frac{D^2}{2!} + \frac{D^3}{3!} + \ldots) f(k) \end{aligned}$$

It's related to Bernoulli numbers.

There are other areas I haven't yet explored, for example proving the Harmonic series $H_n = 1 + \frac{1}{2} + \ldots + \frac{1}{n}$ has no closed form in terms of addition, multiplication which is of a simpler nature.

I've also considered this as a standard way to represent algorithms, but I will admit computing science isn't my strength.

In anycase, I'm only mentioning these examples as a way of saying there's much more that can be done. To further this point, I have displayed some variation in the use of my notation across the different examples, there's a little less consistency there than even I would prefer, but I haven't worked with it enough myself to know for sure which variations and techniques are the best, or appropriate, maybe they all are.

If you've read this far, then I hope, to quote Mr. Burns from the Simpsons, "that you have enjoyed reading this dear reader as much as I have enjoyed writing it!"

In anycase, I think the point is had, I will leave it at that.

The End

References

[1] William O'Grady, John Archibald. Contemporary Linguistic Analysis: An Introduction, Fifth Edition. Pearson Education Canada Inc. (2004).

[2] Karel Hrbacek, Thomas Jech. Introduction to Set Theory: Third Edition, Revised and Expanded. Marcel Dekker, Inc. (1999).

[3] R.L. Graham, D.E. Knuth, O. Patashnik. Concrete Mathematics. Addison-Wesley Publishing (1994).

[4] John M. Harris, Jeffry L. Hirst, Michael J. Mossinghoff. Combinatorics and Graph Theory. Springer-Verlag New York, Inc. (2000).

[5] Adam Drozdek. Data Structures and Algorithms in C++: Third Edition. Course Technology, a division of Thomson Learning, Inc. (2005).