A COMBINATORIAL THEORY OF FORMAL SERIES

ANDRÉ JOYAL

Translation and commentary by
BRENT A. YORGEY

André Joyal. Une théorie combinatoire des séries formelles. *Advances in mathematics*, 42 (1):1–82, 1981

Preface

In his classic 1981 paper *Une théorie combinatoire des séries formelles (A combinatorial theory of formal series)*, Joyal introduces the notion of *(combinatorial) species*, which has had a wide-ranging influence on combinatorics. During the course of researching my PhD thesis on the intersection of combinatorial species and programming languages, I read a lot of secondary literature on species, but not Joyal's original paper—it is written in French, which I do not read. I didn't think this would be a great loss, since I supposed the material in his paper would be well-covered elsewhere, for example in the textbook by Bergeron et al. [1998] (which thankfully *has* been translated into English). However, I eventually came across some questions to which I could not find answers, only to be told that the answers were already in Joyal's paper [Trimble, 2014]. Somewhat reluctantly, I found a copy and began trying to read it, whereupon I discovered two surprising things.

First, armed with a dictionary and Google Translate, reading mathematical French is not too difficult (even for someone who does not know any French!)—though it certainly helps if one already understands the mathematics. Second, it turns out that Joyal's paper makes for excellent reading, and is full of insights and examples which, as far as I know, do not appear in any of the secondary literature. The paper (and the theory of species more generally) has a lot to offer to computer science, and to functional programming in particular.

My initial joy at being able to read the paper was quickly tempered, however, by the knowledge that there are still formidable barriers preventing it from being more widely accessible within the functional programming community. The language barrier is the obvious one, for those who cannot read French; even armed with Google translate it is still very slow going. The second, more serious barrier is that the paper assumes background knowledge—in mathematics in general, and combinatorics in particular—which many in the programming languages community may not have (even those who can read French).

I have therefore decided to produce an English translation, accompanied by my own additional commentary, examples, and code which attempt to explain and illuminate the content to a wider audience. The commentary is typeset using inset boxes with a grey background, like this:

> This is some commentary.

Say something about language used for code examples, and how it fits/what background you need

Note `https://agda.readthedocs.io/en/v2.5.3/tools/generating-latex.html`

This is a long-term project; I do not know how long it will take to finish, but plan to work on it slowly and steadily. Collaboration and contributions are welcome—see the public git repository hosted at `https://github.com/byorgey/series-formelles`. Things I could particularly use help with include:

- Translation. If you know both French and English well, I would love to have your eyes on the translation. There are particular places where I am not sure about the translation, which I have indicated like this [*comme ça*]. However, just because I am sure about other parts does not mean I am right! All of the translation would benefit from checking for accuracy and style.
- Commentary. I would of course be happy to receive contributions of additional commentary, especially from those who are more deeply versed in the relevant mathematics or history than I am, or from those with a different perspective to offer.
- Feedback. Even (especially!) if you are not an expert, an incredibly helpful way you can contribute is simply to try reading the translation and commentary, and let me know of any parts you find confusing or unclear—a signal that the commentary needs to be clarified, or more commentary added.
- I am also interested to eventually publish this somehow—if you represent an interested publisher, or have a recommendation of a suitable one, please let me know!

Technical details. The original paper is available only in scanned form, so as a first step I ran it through the `tesseract` OCR engine.[1] `tesseract` does an excellent job with the text, although naturally it completely mangles the diagrams and equations. From there I have loaded the resulting text document into the Google Translator Toolkit, and use that to aid in the translating process, also typing up equations as I go. The result gets pasted into a LaTeX document, after which I proofread and add diagrams (all produced using the `diagrams` vector graphics framework[2]) and commentary.

A git repository containing the latest state of this project can be found at

`https://github.com/byorgey/series-formelles`.

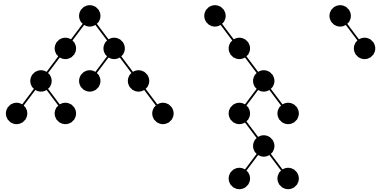
[1] `https://github.com/tesseract-ocr/tesseract`

[2] `http://projects.haskell.org/diagrams/`

A PDF compiled from the most recent source (automatically updated every time I record a commit) can be downloaded from `http://ozark.hendrix.edu/~yorgey/pub/series-formelles.pdf`.

Background

The informal notion of a *combinatorial class* or *combinatorial family* had already been around for quite a while before Joyal's paper. The basic idea is of a set of "shapes" or "structures" containing some sort of "atoms". (Functional programmers should think of an algebraic data type with a type parameter, although this does not capture the full sense.) As an example, the combinatorial class of *rooted binary trees* consists of things like this:



To start, we will suppose that the "atoms"—indicated by dots in the diagram above—are indistinguishable (there will be much more to say about this later!).

Combinatorial classes can be combined in various algebraic ways. Two of the most prominent and familiar are sum (disjoint union) and product (pairing):

- The *sum* $F + G$ of two classes F and G consists of the disjoint union of the classes. In other words, an $F+G$ structure consists of *either* an F structure or a G structure (along with a tag saying which). This corresponds to a sum type, for example, *Either F G* in Haskell.
- The *product* $F \cdot G$ of two classes F and G consists of the class of all *ordered pairs* of structures from the two classes, that is,

$$F \cdot G = \{(f, g) \mid f \in F, g \in G\}.$$

  This corresponds to a product or pair type, for example, (F, G) in Haskell.

Calling these operations *sum* and *product* can be initially justified by thinking about *finite* combinatorial classes. In the case that F and G are finite, one can verify that

$$\begin{aligned} |F + G| &= |F| + |G| \qquad \text{and} \\ |F \cdot G| &= |F| \cdot |G|, \end{aligned}$$

where we use $|F|$ to denote the size (that is, the number of structures) of a finite combinatorial class. That is, the size of a disjoint union $F+G$ is the sum of the sizes F and of G, and the size of a product $F \cdot G$ is the product of the sizes. Describing a combinatorial class algebraically (in terms of sums and products of more primitive classes) gives us a corresponding algebraic handle on its size; this is one of the key ideas of combinatorics.

However, considering only finite combinatorial classes is much too restrictive. For example, we have previously seen the class of binary trees, which is certainly not finite. How can we continue to do meaningful combinatorics with infinite classes? Combinatorics has *counting* at its heart, after all, but counting to infinity is not very informative (nor much fun).

The simple resolution comes from a slight change in perspective: although there are infinitely many binary trees, there are only a finite number *of any given size*, where the *size* of a structure is defined (informally, for now) as the number of

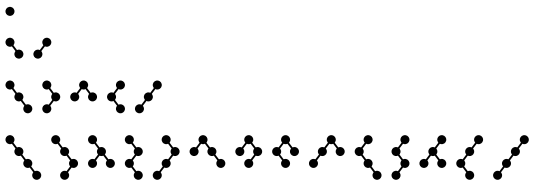


FIGURE 0. All nonempty binary trees with up to 4 atoms

atoms it contains. For example, Figure 0 shows all binary trees with up to 4 atoms, grouped by size. So it makes good sense to count binary trees if we focus on counting how many there are of each size. If T represents the set of all binary trees we will write T_n to denote the set of binary trees of size n, and $|T_n|$ its cardinality.

Combinatorial classes with this property—having only finitely many structures of each given size—are called *finitary*. The cardinality of a finite set is just a single natural number; generalizing to finitary combinatorial classes, we can say that the "cardinality" of a finitary class is a (countably) infinite sequence of natural numbers

$$|F_0|, |F_1|, |F_2|, \ldots,$$

describing the number of structures of each size. Famously, the generalized cardinality of the class of binary trees is the sequence of *Catalan numbers* [Stanley, 2015], which begins

$$1, 1, 2, 5, 14, 42, 132 \ldots$$

(the rows of Figure 0 show the 1, 2, 5, and 14 trees of sizes 1–4).

Let's now reconsider the combinatorial operations of sum and product, and see what operations they correspond to on generalized cardinalities of finitary classes.

- Since an $F + G$ structure is either an F structure or a G structure, the number of $F + G$ structures of size n is just the sum of the number of F structures of size n and the number of G structures of size n. That is,

$$|(F + G)_n| = |F_n| + |G_n|.$$

- What about the number of structures of $F \cdot G$ of size n? This turns out to be a bit more interesting. An $F \cdot G$ structure is a pair of an F structure and a G structure, whose total size is the sum of the sizes of the component structures. Thus, to get a structure of size n, we need to pair an F structure of size k and a G structure of size $n - k$. For each k, the number of ways to pick an appropriate pair is the product of the number of choices for an F structure of size k and the number of choices for a G structure of size $n - k$. That is,

$$|(F \cdot G)_n| = \sum_{0 \leqslant k \leqslant n} |F_k||G_{n-k}|.$$

These generalized cardinalities lead directly to the theory of *(ordinary) generating functions* [Wilf, 2005]. The observation is that we can encode sequences $|F_0|, |F_1|, |F_2|, \ldots$ as the coefficients of infinite power series,

$$|F_0| + |F_1|x + |F_2|x^2 + \cdots = \sum_{n \geqslant 0} |F_n|x^n.$$

This is much more than a gimmick, because sum and product of power series corresponds exactly to sum and product of combinatorial classes:

- To add two power series, one adds coefficients of like powers, that is,

$$\left(\sum_{n\geqslant 0} |F_n| x^n\right) + \left(\sum_{n\geqslant 0} |G_n| x^n\right) = \sum_{n\geqslant 0} \left(|F_n| + |G_n|\right) x^n.$$

  On the right-hand side we get $|F_n| + |G_n|$ as the coefficient of x^n, which as we have previously seen, is in fact the number of $F + G$ structures of size n. So adding the generating functions for F and G yields the generating function for $F + G$.
- Now consider multiplying two power series. Each term in the output will be the product of two terms, one from each input. Powers add when multiplying, so each x^n term in the output will arise from the product of some x^k term and some x^{n-k} term. In particular, multiplying ax^k and bx^{n-k} results in abx^n. Once we collect up like terms in the result, the coefficient of x^n will therefore be the sum of the products of coefficients of every possible pair of terms whose powers add up to n. Symbolically:

$$\left(\sum_{n\geqslant 0} |F_n| x^n\right)\left(\sum_{n\geqslant 0} |G_n| x^n\right) = \sum_{n\geqslant 0} \left(\sum_{0\leqslant k\leqslant n} |F_k||G_{n-k}|\right) x^n.$$

  Once again, we see that the coefficient of x^n in the result is exactly the expression which we previously argued counts the number of $F \cdot G$ structures of size n. Therefore, multiplying the generating functions for F and G yields the generating function for $F \cdot G$.

Cite Wilf, "Clothesline for hanging sequence of numbers."

To make this more concrete, consider the following Agda [Norell, 2007] code which implements these ideas. We encode the coefficients of a generating function not as a literally infinite sequence, but as a function $\mathbb{N} \to \mathbb{N}$, which takes a natural number n as input and ouputs the coefficient of x^n, that is, the number of structures of size n. Computationally, this is a nicer representation in many ways, and also turns out to be the proper perspective from which to later generalize to the notion of species.

In Agda, we can define the type of ordinary generating functions OGF literally as the function space $\mathbb{N} \to \mathbb{N}$:

```
OGF : Set
OGF = ℕ → ℕ
```

We now encode a few primitive generating functions which will come in handy: the constantly zero generating function $f(x) = 0$ has all coefficients zero; the constantly 1 generating function $f(x) = 1 = 1 + 0x + 0x^2 + \ldots$ has an x^0 coefficient of 1 and all the rest zero; and finally $f(x) = x$ has only a 1 coefficient for x^1.

```
ZERO : OGF
ZERO _ = 0

ONE : OGF
ONE 0 = 1
```

```
ONE _ = 0

X : OGF
X 1 = 1
X _ = 0
```

Next we can define the operations of sum and product for generating functions, according to the discussion above. Generating function sum just adds corresponding coefficients:

```
_⊕_ : OGF → OGF → OGF
(f ⊕ g) n = f n + g n
```

We can define generating function product using the standard library function applyUpTo, which applies the given function to each natural number in the list $[0, 1, \ldots, n]$.

```
_⊙_ : OGF → OGF → OGF
(f ⊙ g) n = sum (applyUpTo (λ k → f k * g (n ∸ k)) (suc n))
```

somewhere mention terminology "labelled" vs "unlabelled"

example application illustrating the power of the generating function approach?

So far, we have focused on combinatorial classes of structures with indistinguishable atoms. What about structures with *distinguishable* atoms? Taking the sum of two combinatorial classes with distinguishable atoms is straightforward: once again, the number of $F + G$ structures of size n is just the sum of the number of F structures of size n and G structures of size n.

Product, on the other hand, is more interesting. As a concrete example, consider the combinatorial class $\mathcal{C}$ of *cycles*, illustrated below.

illustration

There is only one cycle structure with n indistinguishable atoms, but with distinguishable atoms there are $(n-1)!$ distinct cycle structures (there are $n!$ distinct sequences of n atoms, but this counts each cycle n times, once for each of the n positions at which we could "cut open" a cycle to make it into a sequence).

The product $\mathcal{C} \cdot \mathcal{C}$ consists of *ordered pairs* of cycles. How many different pairs of cycles are there of a given size? The reason the atoms make a difference is that when forming a pair of cycles with some set of atoms, it matters how we distribute the atoms between the two cycles—this doesn't matter if the atoms are indistinguishable. For example,

show example

To count the number of $\mathcal{C} \cdot \mathcal{C}$ structures of size n, we can imagine the choices we could make to pick one particular such structure as follows:

- We must first choose some $k \in \{1, 2, \ldots, n-1\}$; to get a structure of size n overall we must pair a k-cycle with an $(n-k)$-cycle (note there are no size-0 cycles which explains why we do not choose $k = 0$ or n).
- Next, we decide how to partition the n atoms between the two cycles. There are $\binom{n}{k}$ ways to choose k of the n atoms to go in the k-cycle.
- Finally, we pick any of the $(k-1)!$ available k-cycles on the k chosen atoms, and likewise we pick one of the $(n-k-1)!$ cycles on the remaining atoms.

All told, then, the number of pairs of cycles on n distinguishable atoms is

$$\sum_{1 \leqslant k \leqslant n-1} \binom{n}{k} (k-1)!(n-k-1)!.$$

Generalizing, we can see that for arbitrary combinatorial classes F and G, the number of $F \cdot G$ structures on n distinguishable atoms is

$$|(F \cdot G)_n| = \sum_{0 \leqslant k \leqslant n} \binom{n}{k} |F_k||G_{n-k}|,$$

assuming that F_n is the set of F structures on n distinguishable atoms, and similarly for G_n. This is the same as the formula for indistinguishable atoms, except for the extra factor of $\binom{n}{k}$.

Somewhat magically, it turns out counting structures with distinguishable atoms is captured by a different kind of generating function. In particular, we define the *exponential generating function* (egf) by

$$\sum_{n \geqslant 0} |F_n| \frac{x^n}{n!}$$

The $n!$ may seem like a rabbit out of a hat at this point, but hopefully it is at least plausible: it corresponds to the $n!$ different ways a set of n distinguishable atoms can be permuted.

Let's check that multiplying two such egf's yields the egf for the product. Once again, an x^n term in the result will come from the product of an x^k term from the first egf (with a coefficient of $|F_k|/k!$) and an x^{n-k} term from the second (with a coefficient of $|G_{n-k}|/(n-k)!$). The trick is that to make the result into another egf, we must massage each term into the form of some coefficient times $(x^n/n!)$:

$$\begin{aligned}
\left(\sum_{n \geqslant 0} |F_n| \frac{x^n}{n!}\right) \left(\sum_{n \geqslant 0} |G_n| \frac{x^n}{n!}\right) &= \sum_{n \geqslant 0} \left(\sum_{0 \leqslant k \leqslant n} \frac{|F_k|}{k!} \frac{|G_{n-k}|}{(n-k)!} x^n \right) \\
&= \sum_{n \geqslant 0} \left(\sum_{0 \leqslant k \leqslant n} \frac{|F_k|}{k!} \frac{|G_{n-k}|}{(n-k)!} \frac{n!}{n!} x^n \right) \\
&= \sum_{n \geqslant 0} \left(\sum_{0 \leqslant k \leqslant n} \frac{n!}{k!(n-k)!} |F_k||G_{n-k}| \right) \frac{x^n}{n!} \\
&= \sum_{n \geqslant 0} \left(\sum_{0 \leqslant k \leqslant n} \binom{n}{k} |F_k||G_{n-k}| \right) \frac{x^n}{n!}
\end{aligned}$$

Once again, we see that the coefficient of $x^n/n!$ is exactly the same as the formula we already derived for the number of $(F \cdot G)$ structures with n distinguishable atoms; the binomial coefficient $\binom{n}{k}$ falls out of the extra $n!$ in the denominator of the egf terms.

Basic category theory. I assume that the reader is already familiar with the basic definitions of category theory: categories, functors, and ideally natural transformations (though you may be able to get away with learning a bit about natural transformations from this document).

The concept of a *groupoid* comes up quite a bit, and in particular the groupoid $\mathbb{B}$. A *groupoid* is a category where all the morphisms are "invertible", that is, each $m : A \to B$ has a corresponding $m^{-1} : B \to A$ such that $m^{-1} \circ m = id_A$ and $m \circ m^{-1} = id_B$. $\mathbb{B}$ is the category whose objects are *finite* sets and whose morphisms are bijections, that is, functions which are both injective and surjective. Since bijections are invertible, $\mathbb{B}$ is not just a category but a groupoid.

It is always possible to make a bijection between any two finite sets of the same size; conversely, there are no bijections between finite sets of different sizes. Thus, $\mathbb{B}$ can be thought of as the disjoint union of a number of connected components, one for each natural number size.

A good check of your understanding of the necessary basic category theory is to prove that functors preserve isomorphisms: that is, if a morphism $m : A \to B$ in the category $\mathbb{C}$ is invertible, then any functor $F : \mathbb{C} \to \mathbb{D}$ must necessarily send m to an invertible morphism in $\mathbb{D}$.

Categorification. The process of *categorification* cannot be defined rigorously. It attempts to take mathematical objects and find a way to see them as "shadows" of objects in some richer category, in such a way that operations and theorems we care about are also "shadows" of (richer/more complex/more informative) operations and theorems on the category. As we will see in the case of species, this approach often yields great insight into the original class of objects.

Look up/recall how Baez talks about categorification. I looked it up and realized that it is specifically about moving from sets to categories.

As a (particularly germane) example, consider the set of natural numbers $\mathbb{N} = \{0, 1, 2, \dots\}$, ordered by the usual $\leqslant$ relation, along with the usual binary operations of addition, multiplication, and exponentiation. Our goal is to find a category such that

- the natural numbers can be seen as "shadows" of the objects of the category;
- the $\leqslant$ relation can be seen as the shadow of morphisms in the category; and
- addition, multiplication, and exponentiation are shadows of suitable categorical constructions.

One category that fits the bill is the category whose objects are *finite sets* with morphisms being arbitrary (total) *functions* between sets.

XXX injection vs arbitrary function? Baez & Dolan mention FinSet with finite sets and functions.

- Each natural number n can be seen as the "shadow" of all the finite sets having cardinality n. The natural number n should be thought of as a sort of "degenerate finite set" where we have forgotten the identity of the set's elements, and remember only its size.
- There is an injection $S \hookrightarrow T$ if and only if $|S| \leqslant |T|$, so the "shadow" of a morphism is indeed a $\leqslant$ relation between the cardinalities of the sets. Put another way, if we have an injection $S \hookrightarrow T$ but then forget the identities of the elements of S and T, the only thing we can remember about the injection is the fact that the cardinality of S must be $\leqslant$ the cardinality of T.

- Addition of natural numbers is the shadow of coproducts (disjoint unions) of sets. That is,

Start with arithmetic: natural numbers with addition, multiplication, exponentiation. Turn natural numbers into finite sets with functions, get a category with coproducts, products, function spaces. Usual arithmetic laws e.g. $x * (y + z) = x * y + x * z$, $x^{(}y + z) = x^y * x^z$, etc. all turn into theorems expressing isomorphisms bewteen sets. If you take this process and lift it to act on generating functions, you get species.

Contributions.

Joyal's paper turns GFs into combinatorial objects in their own right via categorification.

- First to apply CT to combinatorics
- Unified and generalized known collection of GF techniques
- Applied theory to novel results, and very concise derivations of celebrated results (Cayley, Lagrange inversion)

A COMBINATORIAL THEORY OF FORMAL SERIES

ANDRÉ JOYAL

Abstract. This paper presents a combinatorial theory of formal power series. The combinatorial interpretation of formal power series is based on the concept of species of structures. A categorical approach is used to formulate it. A new proof of Cayley's formula for the number of labelled trees is given as well as a new combinatorial proof (due to G. Labelle) of Lagrange's inversion formula. Polya's enumeration theory of isomorphism classes of structures is entirely renewed. Recursive methods for computing cycle index polynomials are described. A combinatorial version of the implicit function theorem is stated and proved. The paper ends with general considerations on the use of coalgebras in combinatorics.

1. Introduction

The aim of this work is simultaneously to explain, clarify, and unify the subject. The usefulness of formal series in combinatorial calculations is well established. The combinatorial interpretation of the substitution operation has been the subject of relatively recent work (Bender and Goldman [1971], Doubilet et al. [1975], Foata and Schützenberger [1970], Garsia and Joni [1976], Gessel [1977]). The first interpretation (probabilistic) of the substitution of power series dates back to Watson [Kendall, 1966] (in the theory of branching processes).

The main feature of the theory presented here is its degree of generality and simplicity. In this theory, the combinatorial objects corresponding to formal series are the species of structures. The emphasis is placed on the transport of structures rather than on their properties. This point of view is reminiscent of that of Ehresmann [1965] and contrasts with that of Bourbaki [1968]. The combinatorial operations on formal series correspond to operations on species of structures. Algebraic identities between formal expressions often correspond to combinatorial identities. Intuition and calculation can then play on two fronts in a dialogue that resembles that between algebra and geometry. The result is a kind of combinatorial algebra analogous to the geometric algebra of Grassman (and Leibniz). The simplicity of the theory is largely due to the use it makes of the concepts of category theory [MacLane, 1971] (previous theories mainly use the theory of ordered sets and partitions). Furthermore, it highlights the fundamental fact that a very large number of constructed bijections are *natural*, that is to say, they do not depend on a system of coordinates introduced by means of an arbitrary enumeration.

The work contains a few combinatorial innovations, like the concept of the *vertebrate* and a new proof of Cayley's result on the number of trees. There is also a new combinatorial proof of Lagrange's inversion theorem. Polya theory is entirely redone and results in a method for calculating by *recurrence* the coefficients of cycle index polynomials.

The theory presented here is partial. Several fundamental aspects have not been addressed. For example, there is a very general categorical theory of the substitution operation [Kelly, 1974]. Further developments probably require such a level of generality. We limit ourselves to those aspects most suitable (in the opinion of the author) for capturing the attention of a reader unfamiliar with the concepts of category theory.

I am especially grateful to G. Labelle for expressing interest in the author's questions, and to whom I owe the proof of Lagrange's inversion theorem presented here. I am also indebted to J. Beck, F. W. Lawvere, P. Leroux, J. Labelle and S. Schanuel for stimulating conversations on the possible relationship between combinatorics and category theory. I thank G. C. Rota for encouraging me to write this work. I also thank Cathy Kicinski for her work of typing. Finally, this work, written in the sunny Australian spring, would never have been possible without the hospitality of Max Kelly of the University of Sydney.

2. Species of Structures

There already exists a precise concept of species of structures [Bourbaki, 1968, Chap. 4].

> Bourbaki's definition of species of structure is really quite a horrible mess. If one squints at it, one can make out the general idea; Joyal's definition really encapsulates all of its essence in a much more elegant and economical way. This seems to be due in large part to Bourbaki's insistence on using formal set theory as a foundation; as we will see, Joyal's definition benefits tremendously from using category theory instead. Interested and/or masochistic readers can read Bourbaki's definition in Appendix A.

Describing a particular species is often done by specifying the conditions which a structure must satisfy to belong to the species. This description may take the form of an axiomatic theory of the species being considered. A key part of the concept is the transport of structures. We will abstract the concept of species so that the transport of structures is the main aspect. Moreover, as we only deal with the problems of counting and finite enumeration, we will confine ourselves to finitary species, unless otherwise noted.

> See Bergeron et al. [1998, pp. 6–7] for some examples of "axiomatic theories" for particular species. The main point here is *transport* of species, that is, the ability to "swap out" the labels in a structure for some other labels. Joyal's insight is to make transport itself the central defining feature of species; there will be much more to say on this later.

2.1. **Species and cardinality.**

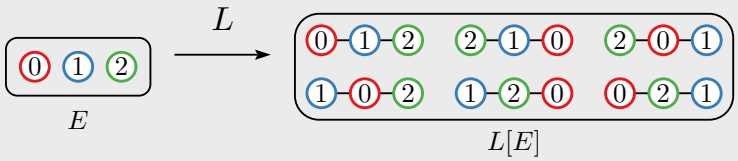
Definition 2.1. A (finitary) species is an endofunctor $M : \mathbb{B} \to \mathbb{B}$ on the groupoid $\mathbb{B}$ of finite sets and bijections.

It's worth pointing out that most subsequent publications (even by Joyal himself) actually define species as functors $\mathbb{B} \to \mathbb{E}$, where $\mathbb{E}$ is the category of (finite) sets and total functions, rather than bijections [Joyal, 1986, Bergeron et al., 1998]. In fact, Joyal introduces functors $\mathbb{B} \to \mathbb{E}$ later in this very paper, in §2.2. It may not seem to make much difference when considering an individual species, but the *category* of species ends up being much richer if we define species as functors $\mathbb{B} \to \mathbb{E}$. For more details, see the commentary on §2.2.

If E is a finite set, $M[E]$ is the set of all *structures* of the species M on E. We say that E is the *underlying* set of $s \in M[E]$, or also that it is *supported* by E. We say also, in an abuse of language, that s is an *element* of M ($s \in M$) and that it is an *M-structure*.

Recall that a functor has two components, a mapping from objects to objects and a mapping from morphisms to morphisms. This first paragraph after the definition sets up some terminology related to the object mapping, which sends finite sets (that is, objects of $\mathbb{B}$) to finite sets. In particular, one should think of M as sending finite sets of *labels* to finite sets of *labelled structures*.

For example, consider the functor $L : \mathbb{B} \to \mathbb{B}$ which sends each finite set of labels to the set of all linear orderings on the labels, as illustrated below for a particular set of labels:



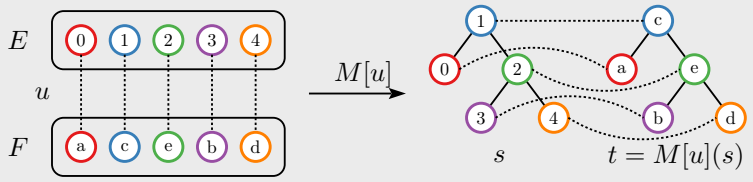
If the set on the left is $E = \{0, 1, 2\}$, then the set on the right is $L[E]$, the set of all L-structures supported by E. Each linear order $s \in L[E]$ is called an L-structure, or element of L, and has $E = \{0, 1, 2\}$ as its underlying set.

If $u : E \to F$ is a bijection, the element $t = M[u](s)$ is the structure on F obtained by *transport along* u. The bijection u is an *isomorphism* between s and t:

$$u : s \to t.$$

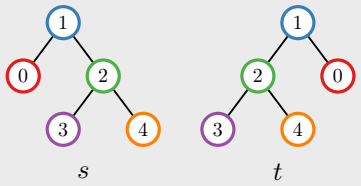
These next sentences have to do with the second component of the functor, the mapping from morphisms to morphisms. In this case, morphisms in the source category are bijections between label sets (which one can think of as *relabellings*), which are sent to bijections between sets of labelled structures (with each structure corresponding to its relabelled version).

The diagram below shows an example, where M is some species of labelled binary trees. u is a bijection between the label sets $E = \{0, \ldots, 4\}$ and $F = \{a, \ldots, e\}$; so $M[u]$ is a bijection between the set $M[E]$ of trees labelled by E and $M[F]$ of trees labelled by F. The diagram shows a particular $s \in M[E]$ and its image $t = M[u](s)$ under the relabelling $M[u]$; notice how each label in the structure has simply been replaced by its corresponding label under u.



Joyal writes $u : s \to t$, which is a mild abuse of notation; u is in fact a bijection between label sets E and F, and s and t are elements of $M[E]$ and $M[F]$ respectively, not sets. The point is that $u : E \to F$ induces a bijection $M[u]$ between the sets of structures $M[E]$ and $M[F]$; if this bijection relates $s \in M[E]$ and $t \in M[F]$, then we say that s and t are isomorphic, as witnessed by u, and write $u : s \to t$. That is, whereas $u : E \to F$ is pronounced "u is a function from E to F", $u : s \to t$ is pronounced "u is a relabelling which, via transport, sends the structure s to the structure t".

A concrete example should help to clarify the idea. The trees s and t shown in the example above are isomorphic, as witnessed by the relabelling u. Any other tree t' of the "same shape" would also be isomorphic to s, as long as there exists some bijection u' which swaps out the labels of s for the corresponding labels in t'. As an example of non-isomorphic tree structures, however, consider the diagram below: there is no way to turn s into t merely by permuting labels. These two tree structures are fundamentally different.



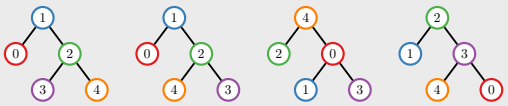
The fact that a species is a *functor* (as opposed to just a mapping from sets of labels to sets of structures) corresponds to the idea of "baking in" the notion of *transport*, mentioned earlier. Here we see that the functorial mapping from a bijection on label sets to a bijection on sets of structures is exactly what allows us to "transport" structures along a bijection of labels to get new, relabelled structures. The fact that we can always do this means that the precise nature of the labels does not matter: there will always be *some* set of labels to allow us to

talk about the atoms in a structure, but we get the same structures no matter what labels we use.

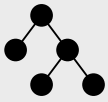
This is, it seems to me, the crux of Joyal's insight: replacing the complicated *intensional* definition of Bourbaki (defined as a very explicit construction of sets and relations) with a simple, *extensional* definision. Species are defined by their behavior, that is, what we can *do* with them, not by what they *are*: species are precisely those things which generate a set of structures from a set of labels and which can be functorially relabelled. This is what Joyal is referring to in the introduction when he writes "The emphasis is placed on the transport of structures rather than on their properties."

We denote by $\mathrm{el}(M)$ the category whose objects are the M-structures and whose morphisms are isomorphisms of M-structures; it is the groupoid of *elements* of M. There is a *forgetful* functor $U : \mathrm{el}(M) \to \mathbb{B}$ whose value on $s \in M$ is the *underlying* set of the structure s. The concept of isomorphism of structures defines an equivalence relation whose classes are the *types* of structures of the species M; these classes are the *connected components* of the groupoid $\mathrm{el}(M)$. We use the notation $\pi_0(M)$ to denote the set of types (of structures) of the species M. If $s \in M$, we denote the *type* of s by the notation $|s| \in \pi_0(M)$.

This definition of structure *types* is an important one, and worth unpacking a bit. A *type* of structures is formally defined as an equivalence class of structures under isomorphism, that is, under relabelling. For example, all the labelled trees shown below are isomorphic, since any two of them are related by some relabelling.



The collection of all such trees (there are infinitely many, since we could use *any* set of labels, not just $\{0 \ldots 4\}$) forms a *type*. I also tend to use the word "shape" to mean the same thing. We can visually represent this type/shape like so:



That is, intuitively, we can often think of types in terms of these pictures of "unlabelled shapes" with indistinguishable dots in place of labels. However, thinking of shapes as structures with indistinguishable labels isn't always good enough—we'll see some examples later where it breaks down—so it's good to be able to return to the formal definition in terms of equivalence classes of labelled structures.

$\mathrm{el}(M)$ is a *groupoid* because all its morphisms are invertible: if $u : s \to t$, that is, if u is a bijection between label sets such that relabelling the structure s by u yields t, then we necessarily have $u^{-1} : t \to s$.

Example 1. Recall that a *simplicial scheme* structure on E is a set $\mathcal{S}$ of non-empty subsets of E such that (i) every non-empty subset contained in an element of $\mathcal{S}$ belongs to $\mathcal{S}$, (ii) the singletons $\{x\}$ for $x \in E$ belong to $\mathcal{S}$. The elements of $\mathcal{S}$ are *simplices*. The dimension of a simplex is one less than its cardinality. A *graph* is a simplicial scheme whose simplices have dimension $\leqslant 1$. If $u : E \to F$ is a bijection, it is clear that $u(\mathcal{S}) = \{u(S) \mid S \in \mathcal{S}\}$ is also a simplicial scheme structure on F. We can therefore consider the *species* of simplicial schemes. It is also clear that if $\mathcal{S}$ is a graph then $u(\mathcal{S})$ is one too; we obtain the species of graphs, which is a *subspecies* of the species of simplicial schemes. More generally, any property P which applies to simplicial schemes, and which is invariant under isomorphism, determines a subspecies of the species of simplicial schemes. For example, connectedness is such an invariant property. The species of *forests* is that of graphs *without cycles*, the species of *trees* is that of connected forests, etc.

This whole business with *simplicial schemes* is unnecessarily general in some sense, but it allows Joyal to deal in one fell swoop with a large collection of common species (graphs, forests, trees…) which can all be seen as special cases of the species of simplicial schemes. It also incidentally provides a relevant motivation for introducing the concept of subspecies. It would be tedious to go through each slight variant on the concept of a graph and show that it defines a valid species, but at the same time, it is unsatisfying to just wave our hands and say things like "this case is similar…". By moving up a notch in abstraction, we get a rigorous yet economical definition that handles many similar cases at once.

Geometrically, *simplices* are what we get when we start with a single point (a "0-simplex"), and then repeatedly add points, each time connecting the new point to all the previous points (and also "filling in" the interior). For example, a 1-simplex is a line segment between two points; a 2-simplex is a (filled in) triangle; a 3-simplex is a solid tetrahedron; and after that they get harder to visualize, but conceptually are just the natural generalization of triangles and tetrahedra to n dimensions.

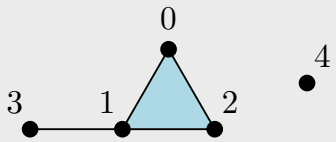
Notice that every n-simplex has a "boundary" which consists of simplices one dimension smaller: a line segment has two endpoints; a triangle has three edges; a tetrahedron has four triangular faces, and so on (yes, a 4-simplex has five tetrahedral "faces".) In fact, if we label the vertices of an n-simplex with elements from some finite set of size $n+1$, *every* size n subset corresponds to the vertices of one of the $(n-1)$-dimensional faces. Of course the same is true of the faces in turn, that is, every size

$n-1$ subset of the vertices is a simplex which is the face of a face, and so on.

This explains what the geometric idea of simplices has to do with the abstract set-based definition given above. In the present context we do not actually care about the geometry of simplices, but only their combinatorial structure. In this case, we just identify a simplex with its set of vertices. A *simplicial complex* $\mathcal{S}$ is a collection of sets which we think of as a collection of simplices; the requirement that every non-empty subset of an element of $\mathcal{S}$ also belongs to $\mathcal{S}$ just means that whenever we have a simplex in our collection, all its faces, faces of faces, and so on, are also simplices in our collection.

The fact that the simplices in $\mathcal{S}$ are all built from some underlying set of labels E means that they can share vertices, so $\mathcal{S}$ is not just a collection of independent, disjoint simplices, but in general they can be glued together in various ways along their faces. For example, below is a graphical representation of the simplicial scheme

$$\{\{0\},\{1\},\{2\},\{3\},\{4\},\{1,3\},\{0,1\},\{0,2\},\{1,2\},\{0,1,2\}\}.$$



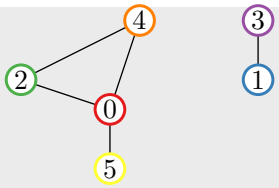
Simplicial schemes, seeing as they consist simply of collections of subsets of the underlying label type E, can be relabelled just by applying the relabelling to every subset. We can check this is functorial: the identity relabelling acts as the identity on a simplicial scheme, and if we do one relabelling followed by another, it is the same as applying the composition of the two relabellings. Therefore simplicial schemes form a valid species.

Joyal then defines *graphs* as simplicial schemes with simplices of dimension $\leqslant 1$. Hence graphs consist only of vertices (dimension 0 simplices, that is, singleton sets) and edges (dimension 1 simplices, that is, sets of size 2), so this is an abstract way of defining what should be a familiar concept. Note that this definition corresponds to graphs in which

- the edges are undirected, since edges are represented by *sets* of vertices, which have no inherent order;
- there are no self-loops, since there cannot be a set containing two copies of the same vertex; and
- there is at most one edge between any two given vertices.

Other types of graphs relaxing some or all of these restrictions can also be defined as species.

An example graph structure is shown in the figure below.



A *subspecies* of a species S is then defined as a subset of S (thought of as the collection of its structures) which is also a species. Joyal points out that a valid subspecies results from restricting to any subset defined by a property which is not affected by relabelling (an *equivariant* property), since in that case relabelling a structure in the subset will always result in another structure in the subset. The property of *dimension* is not affected by relabelling (since relabellings are bijections, they cannot change the size of a set), and hence graphs are a subspecies of simplicial schemes.

As a non-example of a subspecies, consider the subset "simplicial schemes containing the simplex $\{0, 1, 2\}$". This property is not preserved by relabelling (for example, consider the relabelling that sends $0 \mapsto a$, $1 \mapsto b$, and $2 \mapsto c$), and hence this subset is not a valid subspecies of simplicial schemes.

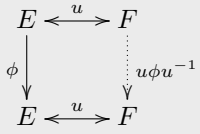
Using this same technique of carving out a subspecies via an equivariant property, Joyal defines the species of forests (graphs with no cycles) and trees (connected forests). For example, removing the 2—4 edge from the example shown above would make it a forest; then adding another edge, say, 4—3, would make it a tree.

It is important to point out that the term *tree* is used here in the common mathematical sense of a connected graph without cycles, rather than in the common computer science sense of an inductive data structure with a root and children. The difference is that a mathematical "tree" has undirected edges, and no privileged root node; adjacent nodes in the tree are just "neighbors", rather than parent and child. A computational "tree" is what mathematicians would call a *rooted* tree, since one can obtain an inductive parent-child tree from a mathematical tree just by picking a distinguished root node; this will be explored in more detail later.

Example 2. The transport of an endofunction $\phi : E \to E$ along a bijection $u : E \xrightarrow{\sim} F$ is by *conjugation*: $\phi \mapsto u\phi u^{-1}$. The species of *permutations* is a subspecies of the species of endofunctions. If we require that the graph of an endofunction is connected, we obtain the subspecies of *connected endofunctions* and, likewise, that of *circular permutations*. An important concept is that of *contraction*: an endofunction $\phi : E \to E$ is a contraction if there exists $x_0 \in E$ such that for every $x \in E$ we have $\phi^n(x) = x_0$ when n is large enough. That is, a contraction is an endofunction which is ultimately constant.

The first sentence of Example 2 both introduces the class of *endofunctions* $\phi : E \to E$ over some set E, and explains why it is a valid species.

To show that it is a valid species, we are required to say how to functorially relabel an endofunction $\phi : E \to E$ given a bijection (relabelling) $u : E \xrightarrow{\sim} F$. The situation can be pictured as below:



"Transporting ϕ along u" means using the bijection u between E and F to turn ϕ from an endofunction on E into an endofunction on F. The way to do this can be read off from the picture: the ϕ on the left side of the square turns into the dotted arrow on the right side of the square, formed by following three sides of the square around. That is, first run u^{-1}, to turn F back into E; then perform ϕ; then run u to get back to F again. Note that $u \cdot id \cdot u^{-1} = u \cdot u^{-1} = id$, and $(u \cdot \phi \cdot u^{-1}) \cdot (u \cdot \psi \cdot u^{-1}) = u \cdot (\phi \cdot \psi) \cdot u^{-1}$, so this defines a valid functor.

Functional programmers are not used to thinking of a definition like $\mathbf{End}(E) = E \to E$ as a functor, since E occurs both positively and negatively. And indeed, given a *function* $u : E \to F$ it is not possible to turn $\phi : E \to E$ into an endofunction $F \to F$. But since u is a *bijection*, we can deal just as easily with negative occurrences as positive, by making use of the inverse u^{-1}. Intuitively, almost *any* type built out of products, coproducts, and functions will define a valid species. (The word "almost" is necessary since the type must be *finitary*, that is, have only finitely many structures of any given size.)

For those who are familiar with homotopy type theory, note that this situation matches up nicely with the notion of transport along a path. Indeed, one can define species in HoTT in such a way that transport of species is literally given by transport along paths [Yorgey, 2014].

Permutations are endofunctions which are invertible; invertibility is invariant under relabelling, so permutations are a subspecies of endofunctions; likewise connected endofunctions, connected permutations (which must consist of one single cycle), and contractions. Typical structures of these species are illustrated in

illustration

.

Joyal claims that *contractions* are important, but note that this is not because they were already important (as far as I know), but simply because they will play an important role later in the paper.

Example 3. Let S be the species of permutations. Consider the groupoid $\mathrm{el}(S)$ of elements of S. The objects of $\mathrm{el}(S)$ are the sets $E \in \mathbb{B}$ equipped with a permutation $\sigma_E \in S[E]$. The morphisms $(E, \sigma_E) \to (F, \sigma_F)$ are the bijections $u : E \to F$ such that $u\sigma_E = \sigma_F u$. Let $x_1, x_2, x_3, \ldots$ be an infinite sequence of variables. Let $I(\sigma_E) = x_1^{d_1} \ldots x_n^{d_n}$, where $n = \operatorname{Card} E$ and where d_i is the number of cycles of length i in σ_E. Two objects (E, σ_E) and (F, σ_E) in $\mathrm{el}(S)$ are isomorphic if and only if $I(\sigma_E) = I(\sigma_F)$. The set $\pi_0(S)$ of connected components of the groupoid $\mathrm{el}(S)$ is

therefore naturally identified with the set $\mathrm{Mon}(x)$ of all monomials in the variables $x_1, x_2, x_3, \ldots$

2.1.1. The group $E!$ of permutations of E acts on $M[E]$ by transport of structures. The set $\pi_0(M[E])$ of *orbits* is identified with the set of types of M-structures supported by sets equipotent with E. We identify the orbit of $s \in M[E]$ with its type $|s|$. The *stabilizer* subgroup of an element $s \in M[E]$ is the group $\mathrm{Aut}(s)$ of *automorphisms* of s. We have the well-known formula

$$\mathrm{Card}\,|s| = \frac{n!}{\mathrm{Card}\,\mathrm{Aut}(s)}.$$

One of the fundamental problems of enumerative combinatorics is to evaluate the two infinite sequences of numbers

$$\mathrm{Card}\,M[n], \quad n \geqslant 0 \ ([n] = \{1, 2, \ldots, n\}), \tag{1}$$

$$\mathrm{Card}\,\pi_0(M[n]), \quad n \geqslant 0. \tag{2}$$

We define two generating functions. The first is a series of Hurwitz [Comtet, 1974]:

$$M(x) = \sum_{n \geqslant 0} \mathrm{Card}\,M[n] \frac{x^n}{n!}. \tag{3}$$

The second is a power series with integer coefficients (without factorial):

$$\tilde{M}(x) = \sum_{n \geqslant 0} \mathrm{Card}\,\pi_0(M[n]) x^n. \tag{4}$$

We say that $M(x)$ is the *cardinality* of M. Let us see immediately that the calculation of $\tilde{M}(x)$ boils down to computing the cardinality of the *associated* species $\tilde{M}$.

Definition 2.2. A structure of the species $\tilde{M}$ is a pair (σ, s) where σ is an automorphism of $s \in M$.

Proposition 2.3. *We have*

$$\tilde{M}(x) = \mathrm{Card}\,\tilde{M}.$$

Proof. We use Burnside's lemma: if a finite group G acts on a finite set X, then the cardinality of the set $\pi_0(X)$ of orbits of X is equal to that of the set $\{(\sigma, x) \mid \sigma \in G, x \in X, \sigma \cdot x = x\}$ *divided by* $\mathrm{Card}\,G$. (See Burnside [1955, p. 191, Theorem VII].) □

2.2. **The category of species.** Species form a category: they are functors, and one can take natural transformations as morphisms. As it is desirable to have a larger class of morphisms than that of isomorphisms, it is best to consider a species as a functor $M : \mathbb{B} \to \mathbb{E}$ to the category $\mathbb{E}$ of finite sets and *functions* (by composing with the inclusion $\mathbb{B} \hookrightarrow \mathbb{E}$).

When considering an individual species, the distinction between a functor $\mathbb{B} \to \mathbb{B}$ and a functor $\mathbb{B} \to \mathbb{E}$ does not seem important: of course any functor $\mathbb{B} \to \mathbb{B}$ can easily be turned into a functor $\mathbb{B} \to \mathbb{E}$ (by composing

with the canonical inclusion $\mathbb{B} \hookrightarrow \mathbb{E}$), but we can actually convert the other way too. Since functors necessarily preserve isomorphisms, every bijection in $\mathbb{B}$ must actually map to a *bijection* in $\mathbb{E}$ under the action of any functor, so the image of a functor $\mathbb{B} \to \mathbb{E}$ actually lies in $\mathbb{B}$.

However, considering the whole collection of species as a category, the distinction is an important one, since the functor categories $\mathbb{B}^{\mathbb{B}}$ and $\mathbb{E}^{\mathbb{B}}$ are not equivalent. Although we have seen above that these two categories have essentially the same collection of *objects* (namely, functors $\mathbb{B} \to \mathbb{B}$ or $\mathbb{B} \to \mathbb{E}$), they do not have the same *morphisms* (namely, natural transformations). A natural transformation between two functors $\mathcal{C} \to \mathcal{D}$ consists of a collection of morphisms in $\mathcal{D}$; if all the morphisms of $\mathcal{D}$ are isomorphisms, then every natural transformation between functors $\mathcal{C} \to \mathcal{D}$ is necessarily an isomorphism as well. But this won't do: we certainly want to be able to talk about morphisms between species which are *not* isomorphisms, for example, the morphism that sends the species of nonempty lists to the species of cycles by "forgetting" which element comes first. Categorically, having only isomorphisms also means the category can't have things like products and coproducts. This, then, is what Joyal means by saying it is "desirable to have a larger class of morphisms than that of isomorphisms" (which turns out to be rather an understatement). Hence $\mathbb{B} \to \mathbb{E}$ really seems to be the "right" notion of species to use. (I am grateful to Ian Price for pointing out this fact to me, which I had somehow missed even after many years of thinking about such things! And of course, it was in Joyal's original paper all along.)

Definition 2.4. A *morphism* $\alpha : M \to N$ is a natural transformation from M to N, considered as functors from $\mathbb{B}$ to $\mathbb{E}$.

One can interpret α as follows: one has a *construction* α allowing one to produce a structure of the species N (output) from a structure of the species M (input), and for every bijection $u : E \to F$ the rectangle

$$\begin{array}{ccc} M[E] & \xrightarrow{\alpha_E} & N[E] \\ {\scriptstyle M[u]}\downarrow & & \downarrow{\scriptstyle N[u]} \\ M[F] & \xrightarrow{\alpha_F} & N[F] \end{array}$$

commutes; this means that the construction is *equivariant* (or invariant): it does not change if one *simultaneously* transports the input and the output along the same bijection; the vast majority of mathematical constructions have this property.

If σ_E is invertible regardless of E, the morphism α is an *isomorphism* between M and N. In this case, we write $M \overset{\alpha}{=} N$, or more simply (in an abuse of notation) $M = N$. If M and N satisfy the weaker condition $\operatorname{Card} M = \operatorname{Card} N$, we say that M and N are *equipotent* species, and we write $M \equiv N$.

Example 4. The construction of the transitive closure of a graph determines a morphism from the species of graphs to the species of partitions.

Example 5. A *rooted tree* is a tree equipped with a *root* (which is an arbitrary vertex of the underlying set). We usually orient the edges of a rooted tree in the

direction of the root. If we adjoin a loop to the root, we obtain the graph of a *contraction* (Example 2). There is an *isomorphism* between the species of rooted trees and the species of contractions.

Example 6. The species of linear orders, of permutations, of permutations equipped with a fixed point, and of circular permutations equipped with an automorphism are all *equipotent*, without being isomorphic.

2.2.1. A morphism of species $M \to N$ determines a functor $\mathrm{el}(M) \to \mathrm{el}(N)$ between the corresponding groupoids. Note that this functor commutes with the forgetful functors

$$\begin{array}{ccc} \mathrm{el}(M) & \longrightarrow & \mathrm{el}(N) \\ & {\scriptstyle U}\searrow \quad \swarrow{\scriptstyle U} & \\ & B. & \end{array}$$

It is not true that a functor $\mathrm{el}(M) \to \mathrm{el}(N)$ is always induced by a morphism of species $M \to N$. For example, if M is the species of preorders and N the species of the orders, the usual construction of an order relation to from a pre-order (on a quotient of the pre-order's underlying set) determines a functor $\mathrm{el}(M) \to \mathrm{el}(N)$ that *does not come* from a morphism of species $M \to N$. However, it is easily checked that every functor $\mathrm{el}(M) \to \mathrm{el}(N)$ which commutes with the forgetful functors U is induced by one and only one species morphism $M \to N$.

2.3. **Relative species.** We want to examine the concept of a *relative* species. We begin with an example. Let G be the species of *graphs*. The concept of *orientation* gives us a functor $O : \mathrm{el}(G) \to \mathbb{E}$, because one can transport a graph orientation along a graph isomorphism. The species of *orientations* (of a graph) is *relative* to that of graphs. On the other hand, the species GO of *oriented graphs* is equipped with a projection $GO \to G$.

Definition 2.5. Let M be a species. A species *relative* to M is a functor $T_M : \mathrm{el}(M) \to \mathbb{E}$.

This paragraph will need a lot of commentary and/or some nice pictures.

Given T_M, one can construct a species T equipped with a morphism $T \to^p M$: set $T[E] = \{(s, \alpha) \mid s \in M[E], \alpha \in T_M[s]\}$. To transport $(s, \alpha) \in T[E]$ along a bijection $u : E \to F$ we begin by transporting s to obtain $t = M[u](s)$, which gives first an isomorphism $s \to^u t \in \mathrm{el}(M)$ and then $\beta = T_M[u](\alpha)$; we set $T[u](s, \alpha) = (t, \beta)$. The morphism $p : T \to M$ is the projection $p(s, \alpha) = s$. Conversely, given a morphism $T \to^p M$, we can construct T_M: if $s \in M[E]$, we have $p_E : T[E] \to M[E]$ and set $T_M[s] = p_E^{-1}\{s\} \subseteq T[E]$. Naturality of p allows us to verify that if $u : E \to F$ is an isomorphism between $s \in M[E]$ and $T \in M[F]$ then $T[u]$ turns $p_E^{-1}\{s\}$ into $p_E^{-1}\{t\}$, which gives $T_M[u] : T_M[s] \to T_M[t]$. We have, in fact, a precise proposition: a *species over* M is a species T equipped with a morphism $T \to^p M$. A morphism $(T, p) \to (T', p')$ between species above M is an arrow $T \to^u T'$ such that $p'\ u = p$.

Proposition 2.6. *The constructions described above define a equivalence between the category of species relative to M and the category $E\|X\|/_M$ of species over M.* (See §3 for the notation $E\|X\|$.)

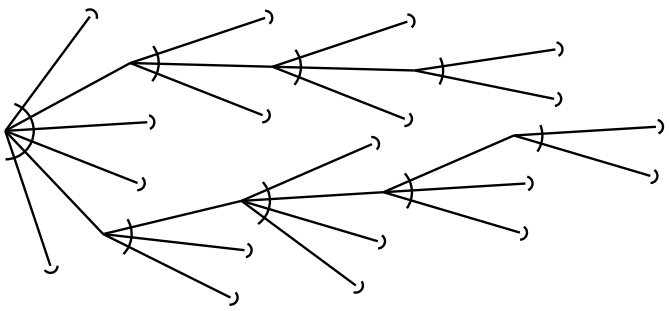


FIGURE 1. An R-enriched tree

Suppose $T_M : \mathrm{el}(M) \to E$ is given. We often say that $(s, \alpha) \in T[E]$ is an M-structure s *equipped* with an element $\alpha \in T_M[s]$. For example, a directed graph is a graph *equipped* with an orientation. A structure of the species $\tilde{M}$ (Definition 2.2) is an M-structure *equipped* with an automorphism. *etc.*

We sometimes use the term "enriched" rather than "equipped". Thus, if R is any species, we will say that endofunction $\phi : E \to E$ is *R-enriched* if each of its *fibers* $\phi^{-1}\{x\}$, $x \in E$ is equipped with an R-structure. Similarly, let a be an rooted tree on E. The *fiber* $a^{-1}\{x\}$ of a vertex $x \in E$ is the set of vertices of a connected to x by an edge adjacent to x (for the orientation of a tree as described in Example 5). We say that a is *R-enriched* if each of its fibers is equipped with an R-structure. (Keeping in mind the empty fibers.)

In graphical representations of endofunctions or R-enriched trees it is often convenient to assume that R-structures on the fibers are placed on the set of *edges* of the fibers. For example, an R-enriched tree can be represented as in Figure 1 where an arc cutting through the edges of a fiber denotes an R-structure. Don't forget the leaves.

3. The Combinational Operations

The category of species is rich in various operations. In this section, we describe several operations of which three are binary. The first two are the sum (disjoint) and the product. With these two operations, the category of species becomes a kind of semi-ring. More precisely, let R be a commutative ring, and denote by $R[\![x]\!]$ the ring of Hurwitz series with coefficients in R: these are the formal series

$$\sum_{n \geqslant 0} a_n \frac{x^n}{n!}, \quad \text{where } n \geqslant 0, a_n \in R.$$

The continued analogy here is that the category of species would be the semiring $\mathbb{E}[\![X]\!]$ of Hurwitz series, but *with coefficients from the category* $\mathbb{E}$ *of finite sets*. The concept of cardinality induces a *homomorphism*

$$\mathrm{Card} : \mathbb{E}[\![X]\!] \to \mathbb{Z}[\![x]\!].$$

In addition, the evaluation $M \mapsto M[0]$ is a functor preserving sum and product

$$\mathbb{E}[\![X]\!] \to \mathbb{E}$$

whose kernel J is an ideal on which we will describe the operations of *divided powers* [Cartan, 1967]

$$\gamma_n : J \to \mathbb{E}[\![X]\!] \quad (n \geqslant 0)$$

so that we have

$$\operatorname{Card} \gamma_n(M) = \frac{M(x)^n}{n!} \quad (n \geqslant 0).$$

Using these operations of divided powers we then describe the operation of *substitution* of one species into another. We end this chapter with an introduction to the *differential calculus* of species and a *combinatorial* proof of the Lagrange inversion formula.

3.1. **Sum and product.** The disjoint sum of two species M and N is the *coproduct* in the category of species:

$$(M + N)[E] = M[E] + N[E].$$

More generally, an arbitrary family of species $(M_i)_{i\in I}$ is summable if for any finite set E, the set of indices $i \in I$ for which $M_i[E] \neq \varnothing$ is finite. We set

$$\left(\sum_{i\in I} M_i\right)[E] = \sum_{i\in I} M_i[E].$$

It is clear that Card preserves sum. We now turn to the definition of the *product* $M \cdot N$ of two species M and N. Define first a *partition* of a set E into two *parts* is a pair (E_1, E_2) such that $E_1 \cup E_2 = E$ and $E_1 \cap E_2 = \varnothing$. One defines in the same way the concept of a partition of E into n pieces ($n \in \mathbb{N}$): we write $E = E_+ \cdots + E_n$ to indicate that $(E_1, \ldots, E_n)$ is a partition of D into n pieces.

Definition 3.1. A structure of the species $M \cdot N$ on $E \in \mathbb{B}$ is a quadruplet (E_1, E_2, s, t), where $E = E_1 + E_2$ and $(s, t) \in M[E_1] \times N[E_2]$.

Proposition 3.2. *We have*

$$\operatorname{Card}(M \cdot N) = \operatorname{Card}(M) \cdot \operatorname{Card}(N).$$

Proof. By definition,

$$(M \cdot N)[E] = \sum_{E_1+E_2=E} M[E_1] \times N[E_2].$$

If $\operatorname{Card} E = n$ and $0 \leqslant k \leqslant n$ there are $\binom{n}{k}$ partitions $E = E_1 + E_2$ with $\operatorname{Card} E_1 = k$. As a result,

$$\operatorname{Card}(M \cdot N)[n] = \sum_{k=0}^{n} \binom{n}{k} \operatorname{Card} M[k] \operatorname{Card} N[n-k].$$

□

The *uniform* species is a species with only one structure on each set; one can give it various representations: the structures of complete graphs, chaotic topologies, and identity functions [*applications identiques*] determine the uniform species.

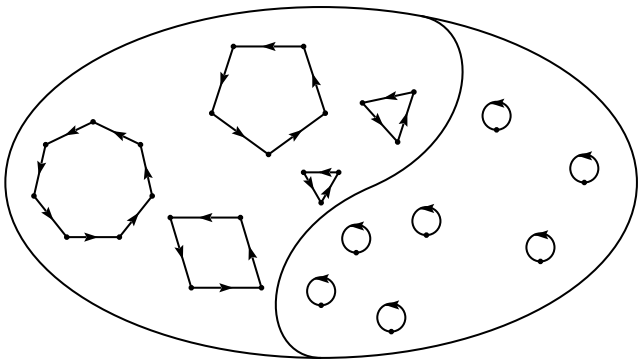


FIGURE 2. $S = S_0 \cdot U$

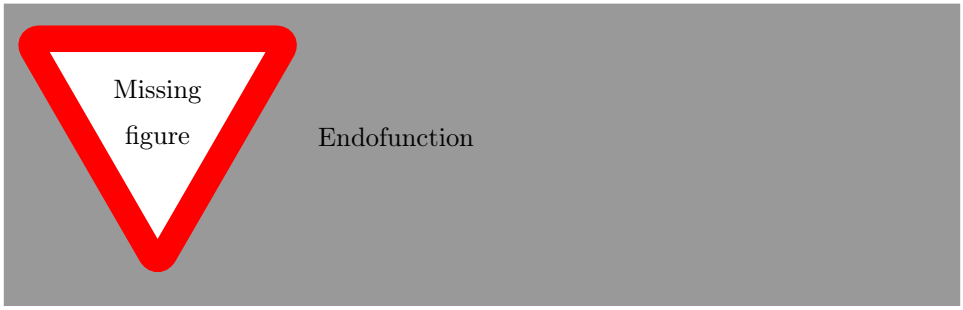


FIGURE 3. XXX

Example 7. Let S be the species of *permutations*, and S_0 that of permutations without *fixed points*, and U the uniform species. We have $S = S_0 \cdot U$ (Figure 2); taking cardinalities, we get:

$$\frac{1}{1-x} = S_0(x)e^x$$

and therefore

$$S_0(x) = \frac{e^{-x}}{1-x}.$$

add 2-cycle to Figure 2. Function for placing arrowhead @ midpoint of arbitrary path?

Example 8. Let D be the species of endofunctions, D_0 the species of endofunctions equipped with a fixed point, and A and the species of rooted trees. Then

$$D_0 = A \cdot D.$$

Indeed, one can partition the domain E of an endofunction ϕ equipped with a fixed point x_0 into two parts $E = E_1 + E_2$. The first, E_1, consists of all the points ultimately transformed into x_0 by ϕ. On E_1, ϕ induces a contraction (Example 2), which is equivalent to a rooted tree. On the second part E_2, ϕ induces an arbitrary endofunction (Figure 3).

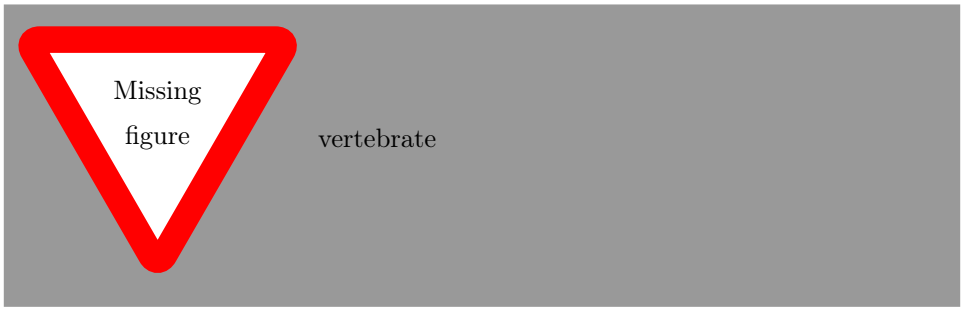


FIGURE 4. XXX

The *product* $M = \prod_{i=1}^{n} M_i$ of a finite sequence of species can be explicitly defined as follows: a structure of the species M on E is a *partition* $E = E_1 + \cdots + E_n$ where each part E_i (possibly empty) is *equipped* with a structure of the species M_i.

One can also describe the *power* N^S of a species N by a finite set S: a structure of the species N^S on E is a function $\chi : E \to S$ where each fiber is *equipped* with a structure of the species N; in other words, it is an N-*enriched* function.

Example 9 (Joyal)**.** A *vertebrate* is a tree *bipointed* by a pair (p_0, p_1) of vertices. We say that p_0 is the *tail vertex* and p_1 the *head vertex*. The shortest path from the tail vertex to the head vertex is the *spine*. The vertices along the spine are *vertebrae*. For each vertex p, let $v(p)$ be the vertebra closest to p. The function v is idempotent, and there is a rooted tree structure on each fiber of v. The roots of these trees are the vertebrae. Thus, a vertebrate on E determines a variable length partition $E = E_1 + \cdots + E_n$, in which each part E_i is equipped with a rooted tree structure, and vice versa. So we have the identity:

$$V = A + A^2 + A^3 + \ldots,$$

where V is the species of vertebrates and A that of rooted trees (Figure 4).

Example 10. Let L be the species of linear orders and S a finite set. A structure of the species L^S on E is a function $E \to S$ where each fiber is equipped with a linear order. The number $l(n, s)$ of such objects (if $n = \operatorname{Card} E$ and $s = \operatorname{Card} S$) is the coefficient of $x^n/n!$ in the series $(1 - x)^{-s}$. This shows that

$$l(n, s) = s(s + 1) \ldots (s + n - 1).$$

Hereafter, we will often consider that a finite set A determines a species by setting

$$A[E] = \begin{cases} A & \text{if } E = \varnothing \\ \varnothing & \text{otherwise.} \end{cases}$$

With this convention, the category $\mathbb{E}$ acts as a ring of coefficients for $\mathbb{E}[\![X]\!]$, because the disjoint sum and Cartesian product of sets can be conflated with the sum and product of species as described earlier.

3.2. **Divided powers and substitution.**

Definition 3.3. Let N be a species such that $N[0] = \varnothing$ and let E be a finite set. An *assembly* of structures of the species N on E is a partition of E where each

class is equipped with a structure of the species N. A *member* of the assembly is a class equipped with the corresponding N-structure. The *divided power* $\gamma_n(N)$ is the species of assemblies of N-structures with exactly n members. The *exponential* $\exp(N)$ is the species of all assemblies of N-structures.

Proposition 3.4. *We have*

$$\operatorname{Card}\gamma_n(N) = \frac{N(x)^n}{n!},$$
$$\operatorname{Card}\exp(N) = \exp(N(x)).$$

Proof. It obviously suffices to prove the first identity. Note first that a structure of the species N^n on E determines a partition $E = E_1 + \cdots + E_n$ into *non-empty* (and disjoint) parts: indeed, E_i is equipped with an N-structure and by hypothesis $N[0] = \varnothing$. If we forget about the linear order on the parts, we have a partition $\{E_1, \ldots, E_n\}$ where each class is equipped with an N-structure. We have shown that an N^n-structure is none other than an N-assembly whose members have been placed in a *linear order*:

$$\operatorname{Card} N^n = n!\operatorname{Card}\gamma_n(N).$$

□

For many species, one may indicate a concept of *connectedness* and demonstrate that any structure consists of a partition where each class is equipped with a connected structure. Under these conditions, a species M is isomorphic to the species of assemblies of connected structures: $M = \exp(M_C)$.

Example 11. *Forests* are assemblies of trees. Forests of rooted trees are assemblies of rooted trees. Permutations are assemblies of circular permutations. Partitions are assemblies of partitions with a single class, *etc.*

The operation of *substitution* of one species into another is the richest in possibilities.

Definition 3.5. Let R and N be species and assume that $N[0] = \varnothing$. The species $R(N)$ is that of pairs (a, ρ), where a is an assembly of N- structures and ρ is an R-structure on the set of members of a.

We say that $R(N)$ is the result of *substituting* N into R. We say that an element of $R(N)$ is an R-assembly (of N-structures). Note immediately that $\exp(N)$ is the result of substituting N in the *uniform* species (Ex. 7).

Theorem 3.6. *Assuming $N[0] = \varnothing$, we have*

$$\operatorname{Card} R(N) = R(N(x)).$$

Proof. For each integer $n \geqslant 0$, let R_n be the species of R-structures where the underlying set has cardinality n. It has a decomposition as a disjoint sum

$$R = \sum_{n\geqslant 0} R_n,$$

inducing a decomposition of R-assemblies according to the number of members:

$$R(N) = \sum_{n\geqslant 0} R_n(N).$$

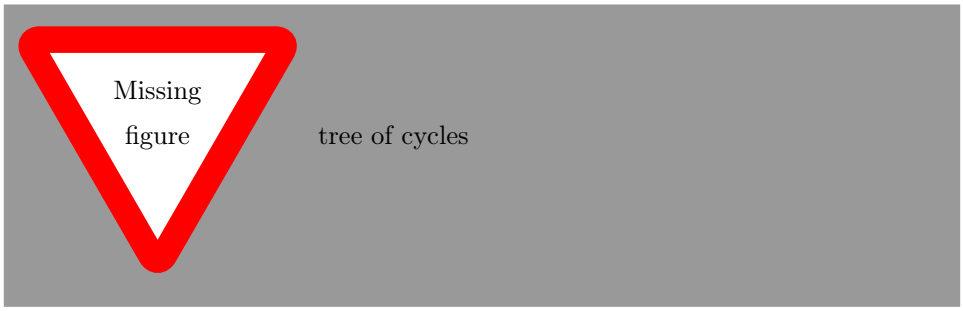


FIGURE 5. XXX

An element of $R_n(N)$ is an assembly of n members *equipped* with an R-structure. We have therefore

$$\operatorname{Card} R_n(N) = \operatorname{Card} \gamma_n(N) \times \operatorname{Card} R[n],$$

resulting in

$$\begin{aligned} \operatorname{Card} R(N) &= \sum_n \geqslant 0 \operatorname{Card} R[n] \frac{N(x)^n}{n!} \\ &= R(N(x)). \end{aligned}$$

□

Remark. To think combinatorially it is necessary to give visual representations. To grasp the nature of a species is to be capable of representing the general shape of its structures. The *shape* of a structure is invariant under isomorphism. What is needed to represent first is the general type of structures of a given species [*Ce qu'il faut arriver à se représenter d'abord c'est le type général des structures d'une espèce donnée*]. This type is independent of a labelling or an enumeration of the vertices of the underlying set. For example, suppose that we want to represent the general type of structures of the species $R(N)$ knowing that we already have a representation for R and N. What we can do is to literally substitute arbitrarily selected N-structures in place of each vertex of an R-structure, For this, one can imagine that the vertices of the R-structure blow up into cells containing the N-structures. The underlying set of a cellular configuration is the sum of the underlying sets of the structures contained in the cells. Thus, if we substitute the species of circular permutations into the species of trees, we obtain a species whose general type can be represented as in Figure 5.

This representation is not the only one, and it is convenient to adapt to the particularities of a species. For example, suppose that the species N is *pointed*, that is, equipped with a morphism $N \to^p B$ where B is the species of "vertices" ($B[E] = E$ for $E \in B$). Each structure $s \in N[E]$ then has a *base point* $p(s) \in E$. One can used the base points to give another representation of $R(N)$-structures: for each vertex of an R-structure one chooses an N-structure and makes the vertex *coincide* with the base point of the selected N-structure. For example, the base point a rooted tree is the root; if we substitute the species A of rooted trees into that of (nonempty) linear orders, we obtain the species of vertebrates (Ex. 9).

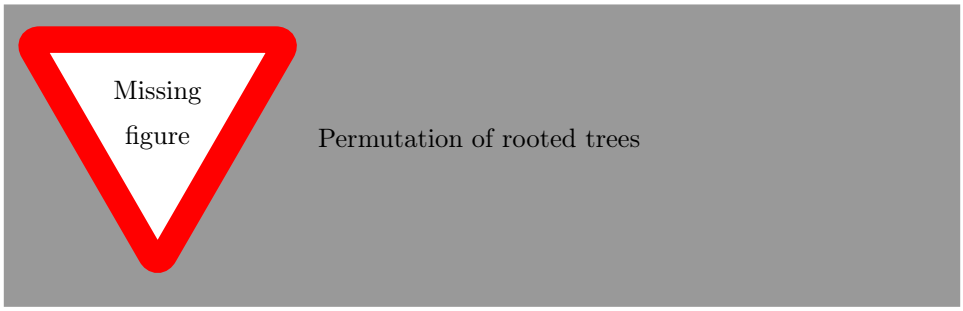


FIGURE 6. XXX

When N is pointed we can define substitution as follows: an $R(N)$-structure on E is a triplet (v, α, β), where

(1) v is an idempotent function $E \to E$,
(2) α is a function that selects for each $x \in \Im(v)$ a structure $\alpha(x) \in N[v^{-1}\{x\}]$ such that the base point of $\alpha(x)$ *coincides* with $x \in v^{-1}\{x\}$,
(3) β is an R-structure on $\Im(v)$.

Example 12. Let D be the species of endofunctions, S that of permutations, and A that of rooted trees. We have the decomposition

$$D = S(A).$$

Indeed, let $\phi \in D[E]$. A point $x \in E$ is *periodic* if there exists an integer $n \geqslant 1$ such that $\phi^n(x) = x$. The function ϕ *permutes* the periodic points. For each $x \in E$ let $v(x)$ be the first periodic point in the sequence $x, \phi(x), \phi^2(x) \ldots$. The function v is idempotent and its image is the set of periodic points. For each $x \in \Im(v)$ the fiber $v^{-1}\{x\}$ is equipped with a rooted tree structure whose root is x. *Conversely*, if one has an assembly of rooted trees and a permutation of the set of roots, it is clear that one can construct a corresponding endofunction ϕ (Figure 6).

Examples 9 and 12 give a simple proof of Cayley's theorem: the number a_n of trees on a set of cardinality $n \geqslant 1$ is n^{n-2}. Indeed, the number of vertebrates (bipointed trees) is equal to $n^2 a_n$. Example 9 shows that the vertebrates are *linear* assembies of rooted trees. Example 10 shows that endofunctions are *permuted* assemblies of rooted trees. As the number of linear orders coincides with the number of permutations, one obtains $n^2 a_n = n^n$ $(n \geqslant 1)$.

Theorem 3.6 suggests adopting the notation $M(X)$ to designate a species M. The variable X is interpreted as the *singleton* species: there is only a single structure of the species X (up to isomorphism) and it is supported by the singletons [*portée par les singletons*]. The result $M(X)$ of the substitution of X in M is isomorphic to M. For some species we adopt a frankly [*franchement*] algebraic notation if it does not create ambiguity. Thus, $\exp(X)$ or e^X designate the uniform species, Xe^X the species of pointed sets, $e^X - 1$ the uniform nonempty species, $\cosh(X)$ and $\sinh(X)$ the uniform even and odd species, $1/(1-X)$ the species of linear orders, *etc.* However, we retain the notation $S(X)$ to designate the species of permutations in order to avoid confusion with $1/(1-X)$. Similarly, we will use the notation $C(X)$ rather than $\log 1/(1-X)$ to designate the species of circular permutations, *etc.*

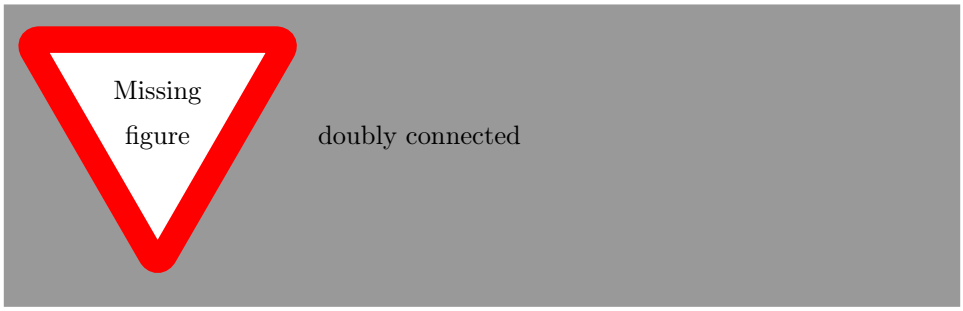


FIGURE 7. XXX

Example 13. A preorder relation $\leqslant$ on E determines an equivalence relation: $x \equiv y$ if and only if one has $x \leqslant y$ and $y \leqslant x$. The preorder relation induced on the quotient $E/\equiv$ is an order relation [*relation d'ordre*], and conversely. This shows that the species of preorders is obtained by substituting the species $e^X - 1$ in the species of order relations. In particular, the species of total preorders is

$$\frac{1}{1-(e^X-1)} = \frac{1}{2-e^X}.$$

Example 14. In a graph, two vertices are *doubly connected* if we can connect them with a path avoiding any edge selected beforehand. The vertices of a graph can be partitioned into doubly connected components. Let $G_c^\bullet$ be the species of connected pointed graphs and $G_{cc}^\bullet$ that of doubly connected pointed graphs. We have the relation

$$G_c^\bullet = G_{cc}^\bullet(Xe^{G_c^\bullet(X)}).$$

Indeed, in a connected pointed graph, consider the doubly connected component H of the base point; for each vertex x let $v(x)$ be the closest vertex located in the component H. We can easily verify that v is well defined. It is an idempotent function whose fibers are equipped with a structure of the species $X \cdot e^{G_c^\bullet(X)}$; the image of v coincides with the doubly connected pointed graph H (Figure 7).

Example 15 (Pólya [1937])**.** Consider the species A of rooted trees. We have the identity (Figure 8)

$$A = X \cdot \exp(A).$$

More generally, the species A_R of R-enriched trees satisfies the equation (see Figure 1)

$$A_R = X \cdot R(A_R).$$

3.3. **Differential calculus.** For any finite set E, let E^+ be the set obtained by adjoining to E an additional item $*$:

$$E^+ = E + \{*\}.$$

Definition 3.7. The *derivative* species M' of a species M is defined as follows:

$$M'[E] = M[E^+].$$

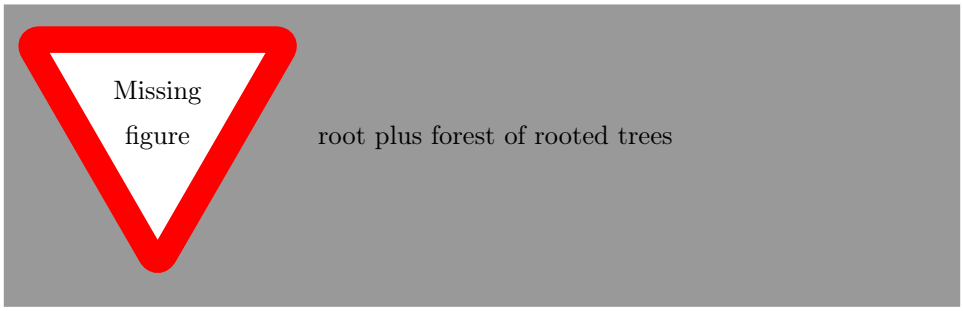


FIGURE 8. XXX

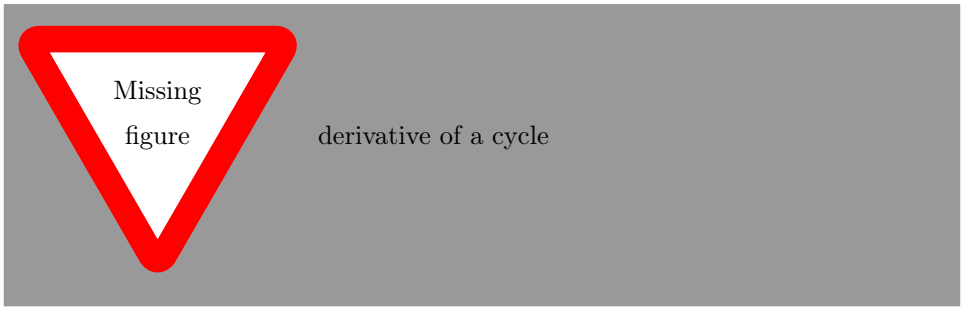


FIGURE 9. XXX

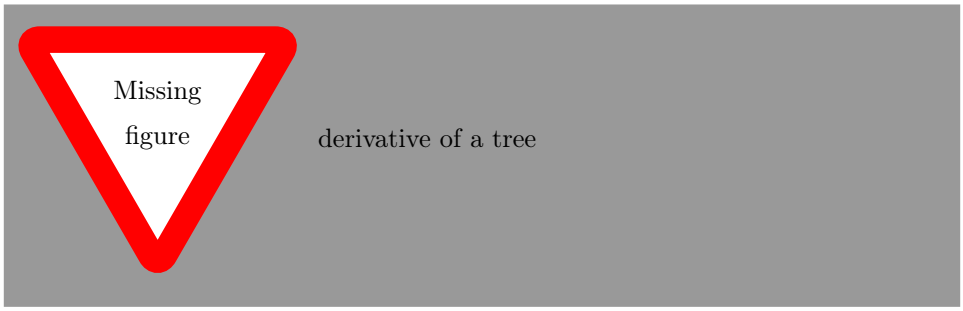


FIGURE 10. XXX

Example 16. The derivative of the species $C(X)$ of circular permutations is the species of linear orders (Figure 9):

$$C'(X) = \frac{1}{1-X}.$$

Example 17. The derivative of the species of trees is that of forests of pointed trees (Figure 10).

Example 18. Recall that a graph is *even* if the number of edges adjacent to each vertex is even. The derivative of the species of even graphs is the species of graphs (Figure 11). [Harary and Palmer, 1973].

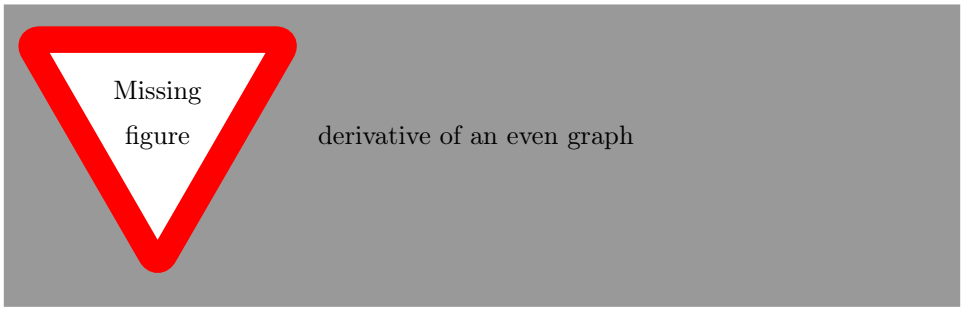


FIGURE 11. XXX

Example 19. The derivative of the species of linear orders $1/(1-X)$ is equal to $(1/(1-X)) \cdot (1/(1-X))$.

Recall that a *pointed* structure of the species M on E is an element of $E \times M[E]$. We denote by $M^\bullet$ the species of pointed M-structures.

Proposition 3.8. *We have the relations*

$$\begin{aligned} M^\bullet &= X \cdot M', \\ (M+N)' &= M' + N', \\ (M \cdot N)' &= M' \cdot N + M \cdot N', \\ M(N)' &= M'(N) \cdot N'. \end{aligned}$$

These identities are not only relationships between numeric quantities but real *combinatorial* identities. We leave to the reader the pleasure of demonstrating this.

Example 20. To illustrate the combinatorial differential calculus, consider the equation satisfied by the species A of rooted trees:

$$A = X \cdot e^A.$$

The operation of "pointing" is a derivation [*est une dérivation*]:

$$A^\bullet = X^\bullet \cdot e^A + X \cdot e^A A^\bullet.$$

Pointed, rooted trees are vertebrates:

$$\begin{aligned} V = A^\bullet &= X \cdot e^A + X \cdot e^A V \\ &= A + A \cdot V \\ &= A + A^2 + A^3 + \dots. \end{aligned}$$

It has been shown that vertebrates are (non-empty) linear assemblies of rooted trees. (See Example 9.)

3.4. **The Lagrange inversion formula.**

3.4.1. The methods of analysis sometimes leave traces in the restricted world of formal series. An example is the theory of the *residue* (at the point 0) of a formal meromorphic series. The invariance property of the residue with respect to an invertible change of parameter $x = w(t)$ is a fundamental *algebraic identity*:

$$\operatorname{Res} f(x)dx = \operatorname{Res} f(u(t))u'(t)dt.$$

This identity is equivalent to the *Lagrange inversion formula*. This formula gives the coefficient c_n of x_n in the series $g(v(x))$ when $v(x)$ satisfies the equation $v(x) = xR(v(x))$:

$$c_n = \frac{1}{n} \times \text{coefficient of } t^{n-1} \text{ in } g'(t)R(t)^n.$$

To demonstrate this, it is sufficient to note that this coefficient is equal to the residue at the origin of $g(v(x))/x^{n+1}$. Note further that $u(t) = t/R(t)$ is the inverse series of v. Performing the change of the parameter $x = u(t)$ and using the invariance property yields

$$\begin{aligned} c_n &= \operatorname{Res} g(t)\frac{u'(t)}{u(t)^{n+1}}dt \\ &= \operatorname{Res}\left[\frac{g'(t)}{nu(t)^n} - \left(\frac{g(t)}{nu(t)^n}\right)'\right]dt \\ &= \operatorname{Res}\frac{g'(t)}{nu(t)^n}dt \\ &= \frac{1}{n}\operatorname{Res}\frac{g'(t)R(t)^n}{t^n}dt \\ &= \frac{1}{n} \times \text{coefficient of } t^{n-1} \text{ in } g'(t)R(t)^n. \end{aligned}$$

Since the calculations are reversible, the equivalence between the inversion formula and the invariance property is clear.

3.4.2. Since Pólya [1937], the inversion formula is often used in combinatorics to calculate the coefficients of certain generating series (the canonical example is the species A of rooted trees that satisfy the equation $A = X \cdot \exp(A)$; see Moon [1970]).

There is already a purely combinatorial proof of the inversion formula [Raney, 1960]. That proof is based on different concepts than those used in the following proof. In Chapter 4 XXX, we will give another proof of the inversion formula.

Theorem 3.9 (Lagrange Inversion)**.** *Let R and F be species and let A_R be the species of R-enriched rooted trees. For $n \geqslant 1$, we have:*

$$F(A_R)[n] \equiv F'R^n[n-1]$$

(the $\equiv$ denotes equipotence).

Proof. Recall (Example 15) that A_R satisfies the equation

$$A_R = X \cdot R(A_R).$$

This leads to the derivation

$$\begin{aligned} A'_R &= R(A_R) + X \cdot R'(A_R)A'_R \\ &= R(A_R) + C_R A'_R \end{aligned}$$

and by iteration ($C_R = X \cdot R'(A_R)$)

$$\begin{aligned} &= R(A_R)(1 + C_R + C_R^2 + \dots) \\ &= R(A_R)\frac{1}{1 - C_R}. \end{aligned}$$

At this point, we want to replace $1/(1 - C_R)$ by $S(C_R)$, where S is the species of permutations. We need to interpret the result of such replacement combinatorially.

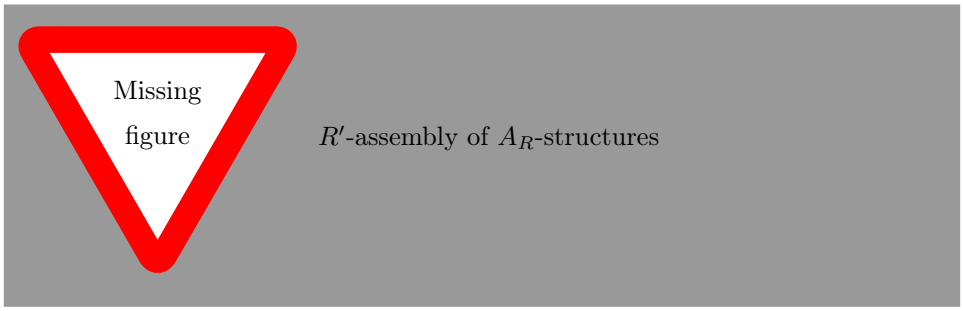


FIGURE 12. XXX

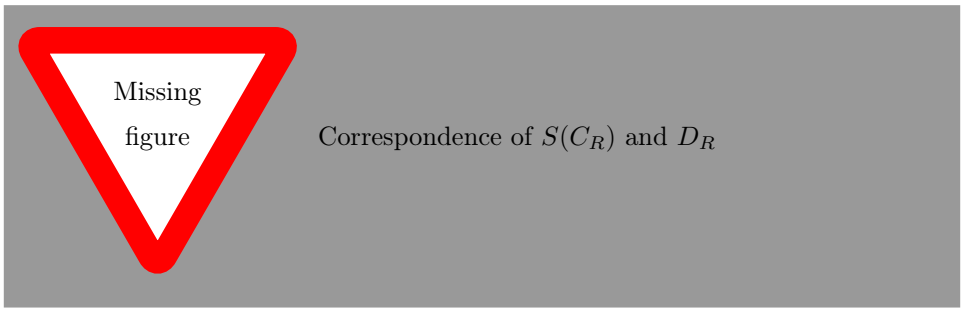


FIGURE 13. XXX

Lemma 3.10 (Labelle [1981])**.** *The species $C_R = X \cdot R'(A_R)$ coincides with that of R-enriched contractions. (See §2.3 and Example 5.)*

Proof. Indeed, consider an R-enriched contraction $\phi : E \to E$. Let x_0 be the point of convergence of ϕ. We can partition E into two parts: $\{x_0\} + E - \{x_0\}$. On the second part, there is the structure of an R'-assembly of A_R-structures as seen in Figure 12 (the loop explains the presence of the derivative in the formula $C_R = X \cdot R'(A_R)$). *And conversely.* □

Lemma 3.11 (Labelle)**.** *The result of the substitution of C_R in the species S of permutations coincides with the species D_R of R-enriched endofunctions.*

Proof. The lemma states that $S(C_R) = D_R$. We use the decomposition of an endofunction into permutations and rooted trees as described in Example 12. Replace the rooted trees in this decomposition by contractions. If we compare, at each point, the fiber of the endofunction with the fiber of the contraction "containing" that point, we realize that they are in bijection (they coincide if the point is not periodic). One can then carry the R-structures along these bijections. This shows that the R-enriched endofunctions are in canonical bijection with the permutation assemblies [*assemblées permutées*] of R-enriched contractions. □

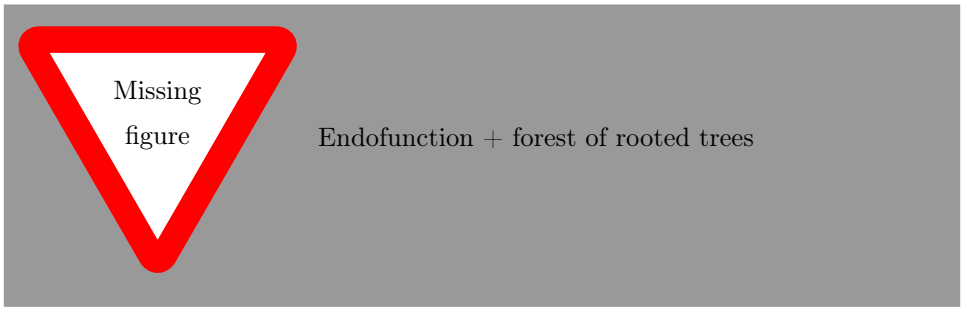


FIGURE 14. XXX

The proof of the theorem continues by taking the *derivative* of $F(A_R)$:

$$\begin{aligned} F(A_R)' &= F'(A_R)A'_R \\ &= F'(A_R)R(A_R)\frac{1}{1-C_R} \\ &\equiv F'(A_R)R(A_R)D_R. \end{aligned}$$

Lemma 3.12 (Labelle, Repartitioning [*repartage*] lemma)**.** *Let G be any species. A structure of the species $G(A_R)D_R$ on a set E can be interpreted as giving a partition $E = E_1 + E_2$ whose first part E_1 is equipped with a G-structure γ and the second part E_2 is equipped with an R-enriched function $\lambda : E_2 \to E$.*

Proof. Let (F_1, F_2, g, f) be a structure of the species $G(A_R) \cdot D_R$ on E. We have $E = F_1 + F_2$. We can describe g as a forest of R-enriched rooted trees where the set of roots is equipped with a G-structure γ. Let E_1 be the set of these roots, and let E_2 be its complement. This forest of rooted trees, together with the endofunction f, defines an R-enriched function $\lambda : E_2 \to E$. (See Figure 14.) Conversely, starting from $(E_1, E_2, \gamma, \lambda)$, we first recover F_1 as the set of all points in E that are ultimately transformed into E_1 by λ. One may complete the proof by meditating on Figure 14. □

We now use this lemma to compute the cardinality of $(G(A_R)D_R)[n]$.

note typo R_R

Indeed, the R-enriched functions $\lambda : E_2 \to E$ *coincide* with the R-enriched functions $\lambda : E_2 \to [n]$ in the case where $E = [n]$ (see the note preceding Example 9). We have therefore

$$(G(A_R)D_R)[n] = (G \cdot R^n)[n].$$

Finally, note that we want to calculate $F(A_R)[n]$, and if $n \geqslant 1$, we have

should there be a prime on the first line?

$$\begin{aligned} F(A_R)[n] &= F(A_R)[n-1] \\ &\equiv F'(A_R)R(A_R)D_R[n-1] \\ &\equiv (F'R)R^{n-1}[n-1] \\ &= F'R^n[n-1]. \end{aligned}$$

□

Translate Section 3 and onwards.

References

Edward A. Bender and Jay R. Goldman. Enumerative uses of generating functions. *Indiana Univ. Math. J.*, 20:753–765, 1971.

François Bergeron, Gilbert Labelle, and Pierre Leroux. *Combinatorial species and tree-like structures*. Number 67 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 1998.

Nicolas Bourbaki. *Élements de mathématiques. Théorie des ensembles*. Hermann, Paris, 1968.

William Burnside. *Theory of Groups of Finite Order*. Dover, New York, 2nd edition, 1955.

Henri Cartan. *Séminaire Henri Cartan (7th year, 1954–1955)*. Benjamin, 1967.

Louis Comtet. *Advanced Combinatorics*. Reidel, Dordrecht-Holland/Boston, 1974.

Peter Doubilet, Gian-Carlo Rota, and Richard P. Stanley. The idea of generating functions. In Gian-Carlo Rota, editor, *Finite Operator Calculus*, pages 83–134. Academic Press, New York, 1975.

Charles Ehresmann. *Catégories et structures*. Dunod, Paris, 1965.

Dominique Foata and Marcel-Paul Schützenberger. Théorie Géométrique des Polynômes Eulérins. *Lecture Notes in Mathematics*, (138), 1970.

Adriano M. Garsia and Saj-nicole A. Joni. Lecture Notes in Combinatorics. UCSD Department of Mathematics, La Jolla, 1976.

Ira Martin Gessel. *Generating Functions and Enumeration of Sequences*. PhD thesis, Massachusetts Institute of Technology, 1977.

Frank Harary and Edgar M. Palmer. *Graphical Enumeration*. Academic Press, New York/London, 1973.

André Joyal. Une théorie combinatoire des séries formelles. *Advances in mathematics*, 42(1):1–82, 1981.

André Joyal. Foncteurs analytiques et espèces de structures. In Gilbert Labelle and Pierre Leroux, editors, *Combinatoire Énumérative*, volume 1234 of *Lecture Notes in Mathematics*, pages 126–159. Springer Berlin Heidelberg, 1986. ISBN 978-3-540-17207-9. doi: 10.1007/BFb0072514.

Gregory M. Kelly. On clubs and doctrines. In Gregory M. Kelly, editor, *Category Seminar, Lecture Notes in Mathematics*, number 420. Springer-Verlag, New York/Berlin, 1974.

David G. Kendall. Branching processes since 1873. *J. London Math.Soc.*, 41:385–406, 1966.

Gilbert Labelle. Une nouvelle démonstration combinatoire des formules d'inversion de lagrange. *Advances in Mathematics*, 42(3):217–247, 1981.

Saunders MacLane. *Categories for the working mathematician*. Springer-Verlag, 1971.

John W. Moon. Counting labelled trees, canadian mathematical monographs, no. 1. In *Canadian Mathematical Congress*, Montreal, 1970.

Ulf Norell. *Towards a practical programming language based on dependent type theory*. PhD thesis, Chalmers University of Technology, 2007.

George Pólya. Kombinatorische Anzahlbestimmungen fur Gruppen, Graphen und chemische Verbindungen. *Acta Math.*, 68:145–254, 1937.

George N. Raney. Functional composition patterns and power series reversion. *Trans. Amer. Math. Soc.*, 94:441–451, 1960.

Richard P. Stanley. *Catalan Numbers*. Cambridge University Press, 2015. doi: 10.1017/CBO9781139871495.

Todd Trimble. Answer to "Examples of functors **Set** → **Set** which are not analytic". MathOverflow, 2014. URL: `http://mathoverflow.net/a/171456` (visited on 2014-06-10).

Herbert S Wilf. *generatingfunctionology*. AK Peters/CRC Press, 2005.

Brent Abraham Yorgey. *Combinatorial species and labelled structures*. University of Pennsylvania, 2014.

Appendix A. Bourbaki's definition of species

This appears on page 6 of *Éléments de mathématique*, Book 1 (Theory of Sets), Chapter 4, available from `http://sites.mathdoc.fr/archives-bourbaki/PDF/180_nbr_083.pdf`.

cite

We say that we have defined, in a theory $\mathcal{C}$ at least as strong as set theory, a species of structure *when we have:*

(1) *A certain number of distinct variables $x_1, \dots, x_n, s_1, \dots, s_p$ besides the constants of $\mathcal{C}$.*
(2) *The* echelons $T_1, T_2, \dots, T_p$ *of an* echelon construction *on n letters $x_1, \dots, x_n$, equal in number to the letters s_j, distinct or not.*
(3) *A relation $R\{x_1, \dots, x_n, s_1, \dots, s_p\}$ of the theory $\mathcal{C}$, of the form*

$$s_1 \subset T_1(x_1, \dots, x_n) \text{ and } s_2 \subset T_2(x_1, \dots, x_n) \text{ and } \dots \text{ and } s_p \subset T_p(x_1, \dots, x_n) \text{ and } R'\{x_1, \dots, x_n, s_1, \dots, s_p\}$$

so that the following relationship is a theorem of $\mathcal{C}$:

(IS) ($R\{x_1, \dots, x_n, s_1, \dots, s_p\}$ and f_1 is a bijection of x_1 onto y_1 and $\dots$ and f_n is a bijection of x_n onto y_n) $\Longrightarrow$ $R\{y_1, \dots, y_n, s'_1, \dots s'_p\}$, where $y_1, \dots, y_n$ are variables distinct from the constants of $\mathcal{C}$ and from all the variables appearing in $R\{x_1, \dots, x_n, s_1, \dots, s_p\}$, and where we set

$$s'_j = T_j\langle f_1, \dots, f_n\rangle\langle s_j\rangle$$

for $1 \leqslant j \leqslant p$.

The basic idea seems to be that the variables $x_1, \dots, x_n$ represent sets containing labels, and the variables $s_1, \dots, s_p$ represent structures. The relation $R\{x_1, \dots, x_n, s_1, \dots, s_p\}$ holds precisely when the s_j are all the (multi-sorted) structures that can be built out of the labels in the x_i. Condition (IS) ensures that we can bijectively swap out the label sets x_i for different ones.

Functoriality is embedded in the definition of an *echelon construction* on page 2:

Translate echelon construction definition with commentary.

Translated by BRENT A. YORGEY

Department of Mathematics, University of Quebec Montreal, Montreal, Quebec H30 3P8, Canada

Department of Mathematics and Computer Science, Hendrix College, 1600 Washington Ave, Conway, Arkansas, 72032, United States of America