Exploring (De)categorification: Arithmetic, Structure, and Combinatorics

Warren Hatton

October 20, 2023

Introduction

- Who am I?
- How did I become interested in this?

Introduction

- Who am I?
- How did I become interested in this?

Goal and approach

The broad goal of this talk is twofold:

1. First, to convey the "philosophy" of categorification and the (higher) categorical point of view, and
2. second, more practically, to show that adopting this perspective can reveal deep connections between the "concrete" and the "abstract".

Goal and approach

The broad goal of this talk is twofold:

1. First, to convey the "philosophy" of categorification and the (higher) categorical point of view, and
2. second, more practically, to show that adopting this perspective can reveal deep connections between the "concrete" and the "abstract".

Natural numbers and their categorification in FinSet

Groupoids and groupoid cardinality

Stuff types

Combinatorial species and their decategorification to generating functions

Two creation myths of decategorification

> *Long ago, when shepherds wanted to see if two herds of sheep were isomorphic, they would look for a specific isomorphism. In other words, they would line up both herds and try to match each sheep in one herd with a sheep in the other. But one day, a shepherd invented decategorification. She realized one could take each herd and 'count' it, setting up an isomorphism between it and a set of 'numbers', which were nonsense words like 'one, two, three, . . . ' specially designed for this purpose. By comparing the resulting numbers, she could show that two herds were isomorphic without explicitly establishing an isomorphism! In short, the set* $\mathbb{N}$ *of natural numbers was created by decategorifying* FinSet*, the category whose objects are finite sets and whose morphisms are functions between these.*

(Baez and Dolan, [4])

Two creation myths of decategorification

> *The original sin of decategorification is the passage from the category of finite sets* Fin *to its set of isomorphism classes* $\mathbb{N}$.

(Yanovski, [16])

Arithmetic in $\mathbb{N}$

- We can add natural numbers: $n + m$.
- Adding 0 does nothing: $n + 0 = n$.
- We can multiply natural numbers: $n \times m$.
- Multiplying by 1 does nothing: $n \times 1 = n$.
- Multiplication distributes over sum: $n(m + p) = nm + np$.

This gives us what is called a *rig* structure on $\mathbb{N}$: a ring, possibly without negatives.

Arithmetic in $\mathbb{N}$

- We can add natural numbers: $n + m$.
- Adding 0 does nothing: $n + 0 = n$.
- We can multiply natural numbers: $n \times m$.
- Multiplying by 1 does nothing: $n \times 1 = n$.
- Multiplication distributes over sum: $n(m + p) = nm + np$.

This gives us what is called a *rig* structure on $\mathbb{N}$: a ring, possibly without negatives.

Arithmetic in $\mathbb{N}$

- We can add natural numbers: $n + m$.
- Adding 0 does nothing: $n + 0 = n$.
- We can multiply natural numbers: $n \times m$.
- Multiplying by 1 does nothing: $n \times 1 = n$.
- Multiplication distributes over sum: $n(m + p) = nm + np$.

This gives us what is called a *rig* structure on $\mathbb{N}$: a ring, possibly without negatives.

Arithmetic in $\mathbb{N}$

- We can add natural numbers: $n + m$.
- Adding 0 does nothing: $n + 0 = n$.
- We can multiply natural numbers: $n \times m$.
- Multiplying by 1 does nothing: $n \times 1 = n$.
- Multiplication distributes over sum: $n(m + p) = nm + np$.

This gives us what is called a *rig* structure on $\mathbb{N}$: a ring, possibly without negatives.

Arithmetic in $\mathbb{N}$

- We can add natural numbers: $n + m$.
- Adding 0 does nothing: $n + 0 = n$.
- We can multiply natural numbers: $n \times m$.
- Multiplying by 1 does nothing: $n \times 1 = n$.
- Multiplication distributes over sum: $n(m + p) = nm + np$.

This gives us what is called a *rig* structure on $\mathbb{N}$: a ring, possibly without negatives.

Arithmetic in $\mathbb{N}$

- We can add natural numbers: $n + m$.
- Adding 0 does nothing: $n + 0 = n$.
- We can multiply natural numbers: $n \times m$.
- Multiplying by 1 does nothing: $n \times 1 = n$.
- Multiplication distributes over sum: $n(m + p) = nm + np$.

This gives us what is called a *rig* structure on $\mathbb{N}$: a ring, possibly without negatives.

The category FinSet

We write $\top$ for any one-element set and $\perp$ for the zero-element set.

- We can take disjoint unions of finite sets: $X \sqcup Y$.
- Taking disjoint union with $\perp$ does nothing: $X \sqcup \perp \equiv X$.
- We can take products of finite sets: $X \sqcap Y$.
- Taking product with $\top$ does nothing $X \sqcap \top \equiv X$.
- Product distributes over disjoint union: $X \sqcap (Y \sqcup Z) = (X \sqcap Y) \sqcup (X \sqcap Z)$.
- We can take the cardinality of finite sets: $\#(X) = n$ for X a finite set of n elements. In particular, $\#(\top) = 1$, $\#(\perp) = 0$.
- Cardinality distributes over product and disjoint union: $\#(X \sqcup Y) = \#(X) + \#(Y)$, $\#(X \sqcap Y) = \#(X) \times \#(Y)$.

All this amounts to saying that FinSet *decategorifies* to $\mathbb{N}$: FinSet has a categorical analogue of a rig structure.[1]

[1] FinSet, in fact, is a topos.

The category FinSet

We write $\top$ for any one-element set and $\perp$ for the zero-element set.

- We can take disjoint unions of finite sets: $X \sqcup Y$.
- Taking disjoint union with $\perp$ does nothing: $X \sqcup \perp \equiv X$.
- We can take products of finite sets: $X \sqcap Y$.
- Taking product with $\top$ does nothing $X \sqcap \top \equiv X$.
- Product distributes over disjoint union: $X \sqcap (Y \sqcup Z) = (X \sqcap Y) \sqcup (X \sqcap Z)$.
- We can take the cardinality of finite sets: $\#(X) = n$ for X a finite set of n elements. In particular, $\#(\top) = 1$, $\#(\perp) = 0$.
- Cardinality distributes over product and disjoint union: $\#(X \sqcup Y) = \#(X) + \#(Y)$, $\#(X \sqcap Y) = \#(X) \times \#(Y)$.

All this amounts to saying that FinSet *decategorifies* to $\mathbb{N}$: FinSet has a categorical analogue of a rig structure.[1]

[1] FinSet, in fact, is a topos.

The category FinSet

We write $\top$ for any one-element set and $\bot$ for the zero-element set.

- We can take disjoint unions of finite sets: $X \sqcup Y$.
- Taking disjoint union with $\bot$ does nothing: $X \sqcup \bot \equiv X$.
- We can take products of finite sets: $X \sqcap Y$.
- Taking product with $\top$ does nothing $X \sqcap \top \equiv X$.
- Product distributes over disjoint union: $X \sqcap (Y \sqcup Z) = (X \sqcap Y) \sqcup (X \sqcap Z)$.
- We can take the cardinality of finite sets: $\#(X) = n$ for X a finite set of n elements. In particular, $\#(\top) = 1$, $\#(\bot) = 0$.
- Cardinality distributes over product and disjoint union: $\#(X \sqcup Y) = \#(X) + \#(Y)$, $\#(X \sqcap Y) = \#(X) \times \#(Y)$.

All this amounts to saying that FinSet *decategorifies* to $\mathbb{N}$: FinSet has a categorical analogue of a rig structure.[1]

[1] FinSet, in fact, is a topos.

The category FinSet

We write $\top$ for any one-element set and $\bot$ for the zero-element set.

- We can take disjoint unions of finite sets: $X \sqcup Y$.
- Taking disjoint union with $\bot$ does nothing: $X \sqcup \bot \equiv X$.
- We can take products of finite sets: $X \sqcap Y$.
- Taking product with $\top$ does nothing $X \sqcap \top \equiv X$.
- Product distributes over disjoint union: $X \sqcap (Y \sqcup Z) = (X \sqcap Y) \sqcup (X \sqcap Z)$.
- We can take the cardinality of finite sets: $\#(X) = n$ for X a finite set of n elements. In particular, $\#(\top) = 1$, $\#(\bot) = 0$.
- Cardinality distributes over product and disjoint union: $\#(X \sqcup Y) = \#(X) + \#(Y)$, $\#(X \sqcap Y) = \#(X) \times \#(Y)$.

All this amounts to saying that FinSet *decategorifies* to $\mathbb{N}$: FinSet has a categorical analogue of a rig structure.[1]

[1] FinSet, in fact, is a topos.

The category FinSet

We write $\top$ for any one-element set and $\bot$ for the zero-element set.

- We can take disjoint unions of finite sets: $X \sqcup Y$.
- Taking disjoint union with $\bot$ does nothing: $X \sqcup \bot \equiv X$.
- We can take products of finite sets: $X \sqcap Y$.
- Taking product with $\top$ does nothing $X \sqcap \top \equiv X$.
- Product distributes over disjoint union: $X \sqcap (Y \sqcup Z) = (X \sqcap Y) \sqcup (X \sqcap Z)$.
- We can take the cardinality of finite sets: $\#(X) = n$ for X a finite set of n elements. In particular, $\#(\top) = 1$, $\#(\bot) = 0$.
- Cardinality distributes over product and disjoint union: $\#(X \sqcup Y) = \#(X) + \#(Y)$, $\#(X \sqcap Y) = \#(X) \times \#(Y)$.

All this amounts to saying that FinSet *decategorifies* to $\mathbb{N}$: FinSet has a categorical analogue of a rig structure.[1]

[1] FinSet, in fact, is a topos.

The category FinSet

We write $\top$ for any one-element set and $\bot$ for the zero-element set.

- We can take disjoint unions of finite sets: $X \sqcup Y$.
- Taking disjoint union with $\bot$ does nothing: $X \sqcup \bot \equiv X$.
- We can take products of finite sets: $X \sqcap Y$.
- Taking product with $\top$ does nothing $X \sqcap \top \equiv X$.
- Product distributes over disjoint union: $X \sqcap (Y \sqcup Z) = (X \sqcap Y) \sqcup (X \sqcap Z)$.
- We can take the cardinality of finite sets: $\#(X) = n$ for X a finite set of n elements. In particular, $\#(\top) = 1$, $\#(\bot) = 0$.
- Cardinality distributes over product and disjoint union: $\#(X \sqcup Y) = \#(X) + \#(Y)$, $\#(X \sqcap Y) = \#(X) \times \#(Y)$.

All this amounts to saying that FinSet *decategorifies* to $\mathbb{N}$: FinSet has a categorical analogue of a rig structure.[1]

[1] FinSet, in fact, is a topos.

The category FinSet

We write $\top$ for any one-element set and $\bot$ for the zero-element set.

- We can take disjoint unions of finite sets: $X \sqcup Y$.
- Taking disjoint union with $\bot$ does nothing: $X \sqcup \bot \equiv X$.
- We can take products of finite sets: $X \sqcap Y$.
- Taking product with $\top$ does nothing $X \sqcap \top \equiv X$.
- Product distributes over disjoint union: $X \sqcap (Y \sqcup Z) = (X \sqcap Y) \sqcup (X \sqcap Z)$.
- We can take the cardinality of finite sets: $\#(X) = n$ for X a finite set of n elements. In particular, $\#(\top) = 1$, $\#(\bot) = 0$.
- Cardinality distributes over product and disjoint union: $\#(X \sqcup Y) = \#(X) + \#(Y)$, $\#(X \sqcap Y) = \#(X) \times \#(Y)$.

All this amounts to saying that FinSet *decategorifies* to $\mathbb{N}$: FinSet has a categorical analogue of a rig structure.[1]

[1] FinSet, in fact, is a topos.

The category FinSet

We write $\top$ for any one-element set and $\bot$ for the zero-element set.

- We can take disjoint unions of finite sets: $X \sqcup Y$.
- Taking disjoint union with $\bot$ does nothing: $X \sqcup \bot \equiv X$.
- We can take products of finite sets: $X \sqcap Y$.
- Taking product with $\top$ does nothing $X \sqcap \top \equiv X$.
- Product distributes over disjoint union: $X \sqcap (Y \sqcup Z) = (X \sqcap Y) \sqcup (X \sqcap Z)$.
- We can take the cardinality of finite sets: $\#(X) = n$ for X a finite set of n elements. In particular, $\#(\top) = 1$, $\#(\bot) = 0$.
- Cardinality distributes over product and disjoint union: $\#(X \sqcup Y) = \#(X) + \#(Y)$, $\#(X \sqcap Y) = \#(X) \times \#(Y)$.

All this amounts to saying that FinSet *decategorifies* to $\mathbb{N}$: FinSet has a categorical analogue of a rig structure.[1]

[1] FinSet, in fact, is a topos.

Playing around with FinSet

- The categorifier thinks of $\mathbb{N}$ and its structure as a sort of "residue" of constructions on FinSet. We have seen this with addition and multiplication.
- The categorifier would naturally ask: are there other things we do with $\mathbb{N}$ that might arise as the residue of some construction involving FinSet?
- We can divide natural numbers. Sometimes this yields another natural number, other times not. Let's categorify a very simple notion of fractions.

Playing around with FinSet

- The categorifier thinks of $\mathbb{N}$ and its structure as a sort of "residue" of constructions on FinSet. We have seen this with addition and multiplication.
- The categorifier would naturally ask: are there other things we do with $\mathbb{N}$ that might arise as the residue of some construction involving FinSet?
- We can divide natural numbers. Sometimes this yields another natural number, other times not. Let's categorify a very simple notion of fractions.

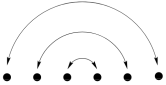
Playing around with FinSet

- The categorifier thinks of $\mathbb{N}$ and its structure as a sort of "residue" of constructions on FinSet. We have seen this with addition and multiplication.
- The categorifier would naturally ask: are there other things we do with $\mathbb{N}$ that might arise as the residue of some construction involving FinSet?
- We can divide natural numbers. Sometimes this yields another natural number, other times not. Let's categorify a very simple notion of fractions.

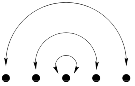
Dividing a finite set in half

Here are some illustrations of what we have in mind:

- Dividing a six-element set in half:

   (From [4].)

- Dividing a five-element set in half:

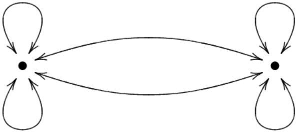
   (From [4].)

To understand what is going on in these diagrams, we will need to consider *group actions*, *groupoids*, and the *weak quotient* of a groupoid by a group.

Groupoids

A *groupoid* is a category where every morphism is invertible. That is to say, every morphism is an isomorphism. A groupoid is *discrete* when there are no morphisms except the identity morphisms.

This is analogous to when we say a group is a monoid with inverses.



(From Bergner and Walker, [7])

Groupoids

- For an object x in a groupoid, an *automorphism* of x is a morphism from x to itself. The set of automorphisms, denoted $\mathrm{Aut}(x)$, form a group.
- If x, y are in the same component of a groupoid, then $\mathrm{Aut}(x) = \mathrm{Aut}(y)$

Group actions

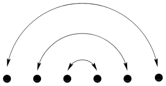
- An action of a group G on a set X is a map $\alpha : G \times X \to X$ such that $\alpha(e, x) = x$ and $\alpha(gh, x) = \alpha(g, \alpha(h, x))$.
- An action is called *free* when each nontrivial $g \in G$ moves every $x \in X$.

Group actions

- An action of a group G on a set X is a map $\alpha : G \times X \to X$ such that $\alpha(e, x) = x$ and $\alpha(gh, x) = \alpha(g, \alpha(h, x))$.
- An action is called *free* when each nontrivial $g \in G$ moves every $x \in X$.

The quotient of a groupoid by a group (orbit)

- The *orbit* Gx of $x \in X$ are all the $x' \in X$ to which x can be moved: $\{gx | g \in G\}$
- We write X/G for the set of all orbits of X.
- Recall our image of "folding" [6]:

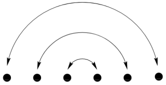


(Baez and Dolan, [4])

- We have $\#([6]/(\mathbb{Z}/2\mathbb{Z})) = 6/2$.
- We can now view this image as depicting the orbits of the action of $\mathbb{Z}/2\mathbb{Z}$ on the discrete groupoid of six objects.
- But it is also a groupoid: $\mathcal{G}/G$ yields a groupoid.
- This action is free: everything gets moved.

The quotient of a groupoid by a group (orbit)

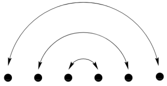
- The *orbit* Gx of $x \in X$ are all the $x' \in X$ to which x can be moved: $\{gx | g \in G\}$
- We write X/G for the set of all orbits of X.
- Recall our image of "folding" [6]:


(Baez and Dolan, [4])

- We have $\#([6]/(\mathbb{Z}/2\mathbb{Z})) = 6/2$.
- We can now view this image as depicting the orbits of the action of $\mathbb{Z}/2\mathbb{Z}$ on the discrete groupoid of six objects.
- But it is also a groupoid: $\mathcal{G}/G$ yields a groupoid.
- This action is free: everything gets moved.

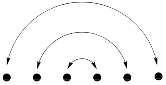
The quotient of a groupoid by a group (orbit)

- The *orbit* Gx of $x \in X$ are all the $x' \in X$ to which x can be moved: $\{gx | g \in G\}$
- We write X/G for the set of all orbits of X.
- Recall our image of "folding" [6]:

   (Baez and Dolan, [4])
- We have $\#([6]/(\mathbb{Z}/2\mathbb{Z})) = 6/2$.
- We can now view this image as depicting the orbits of the action of $\mathbb{Z}/2\mathbb{Z}$ on the discrete groupoid of six objects.
- But it is also a groupoid: $\mathcal{G}/G$ yields a groupoid.
- This action is free: everything gets moved.

The quotient of a groupoid by a group (orbit)

- The *orbit* Gx of $x \in X$ are all the $x' \in X$ to which x can be moved: $\{gx | g \in G\}$
- We write X/G for the set of all orbits of X.
- Recall our image of "folding" [6]:

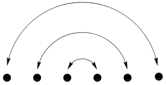


(Baez and Dolan, [4])

- We have $\#([6]/(\mathbb{Z}/2\mathbb{Z})) = 6/2$.
- We can now view this image as depicting the orbits of the action of $\mathbb{Z}/2\mathbb{Z}$ on the discrete groupoid of six objects.
- But it is also a groupoid: $\mathcal{G}/G$ yields a groupoid.
- This action is free: everything gets moved.

The quotient of a groupoid by a group (orbit)

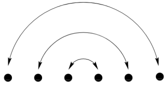
- The *orbit* Gx of $x \in X$ are all the $x' \in X$ to which x can be moved: $\{gx | g \in G\}$
- We write X/G for the set of all orbits of X.
- Recall our image of "folding" [6]:

  (Baez and Dolan, [4])
- We have $\#([6]/(\mathbb{Z}/2\mathbb{Z})) = 6/2$.
- We can now view this image as depicting the orbits of the action of $\mathbb{Z}/2\mathbb{Z}$ on the discrete groupoid of six objects.
- But it is also a groupoid: $\mathcal{G}/G$ yields a groupoid.
- This action is free: everything gets moved.

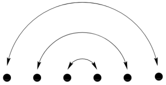
The quotient of a groupoid by a group (orbit)

- The *orbit* Gx of $x \in X$ are all the $x' \in X$ to which x can be moved: $\{gx|g \in G\}$
- We write X/G for the set of all orbits of X.
- Recall our image of "folding" [6]:

   (Baez and Dolan, [4])
- We have $\#([6]/(\mathbb{Z}/2\mathbb{Z})) = 6/2$.
- We can now view this image as depicting the orbits of the action of $\mathbb{Z}/2\mathbb{Z}$ on the discrete groupoid of six objects.
- But it is also a groupoid: $\mathcal{G}/G$ yields a groupoid.
- This action is free: everything gets moved.

The quotient of a groupoid by a group (orbit)

- The *orbit* Gx of $x \in X$ are all the $x' \in X$ to which x can be moved: $\{gx | g \in G\}$
- We write X/G for the set of all orbits of X.
- Recall our image of "folding" [6]:

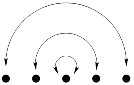


(Baez and Dolan, [4])

- We have $\#([6]/(\mathbb{Z}/2\mathbb{Z})) = 6/2$.
- We can now view this image as depicting the orbits of the action of $\mathbb{Z}/2\mathbb{Z}$ on the discrete groupoid of six objects.
- But it is also a groupoid: $\mathcal{G}/G$ yields a groupoid.
- This action is free: everything gets moved.

Nonfree action

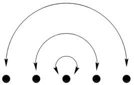
- Now recall our image of folding [5] in half:



- This action is not free: one element is folded upon itself.
- We have $\#([5]/(\mathbb{Z}/2\mathbb{Z})) = 3 \neq 2\frac{1}{2}$

Nonfree action

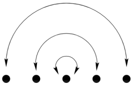
- Now recall our image of folding [5] in half:



- This action is not free: one element is folded upon itself.
- We have $\#([5]/(\mathbb{Z}/2\mathbb{Z})) = 3 \neq 2\frac{1}{2}$

Nonfree action

- Now recall our image of folding [5] in half:



- This action is not free: one element is folded upon itself.
- We have $\#([5]/(\mathbb{Z}/2\mathbb{Z})) = 3 \neq 2\frac{1}{2}$

Weak quotient

- The weak quotient $\mathcal{G} /\!\!/ G$ of a groupoid $\mathcal{G}$ by a group G is the groupoid with objects the objects of $\mathcal{G}$ and morphisms $g : s \to s'$ when $gs = s'$.
- Now, we want to define some notion of cardinality $\kappa(\mathcal{G})$ of groupoids so that $\kappa(\mathcal{G} /\!\!/ G) = \frac{\kappa(\mathcal{G})}{\kappa(G)}$.

Weak quotient

- The weak quotient $\mathcal{G} /\!\!/ G$ of a groupoid $\mathcal{G}$ by a group G is the groupoid with objects the objects of $\mathcal{G}$ and morphisms $g : s \to s'$ when $gs = s'$.
- Now, we want to define some notion of cardinality $\kappa(\mathcal{G})$ of groupoids so that $\kappa(\mathcal{G} /\!\!/ G) = \frac{\kappa(\mathcal{G})}{\kappa(G)}$.

Groupoid cardinality

- The *groupoid cardinality* of a groupoid $\mathcal{G}$ is

$$\kappa(\mathcal{G}) = \sum_{[x]\in\underline{\mathcal{G}}} \frac{1}{\#(\mathrm{Aut}(x))},$$

  where $\underline{\mathcal{G}}$ is the set of isomorphism classes of objects of $\mathcal{G}$
- A groupoid is called *tame* if its groupoid cardinality converges.
- Groupoid cardinality agrees with cardinality on finite sets (discrete n-object groupoids)
- A finite group of order n (viewed as a one-object groupoid) has groupoid cardinality $\frac{1}{n}$. So, for any real number x we can construct, by taking the appropriate disjoint unions, a groupoid whose groupoid cardinality converges to x.

Groupoid cardinality

- The *groupoid cardinality* of a groupoid $\mathcal{G}$ is

$$\kappa(\mathcal{G}) = \sum_{[x] \in \underline{\mathcal{G}}} \frac{1}{\#(\mathrm{Aut}(x))},$$

where $\underline{\mathcal{G}}$ is the set of isomorphism classes of objects of $\mathcal{G}$
- A groupoid is called *tame* if its groupoid cardinality converges.
- Groupoid cardinality agrees with cardinality on finite sets (discrete n-object groupoids)
- A finite group of order n (viewed as a one-object groupoid) has groupoid cardinality $\frac{1}{n}$. So, for any real number x we can construct, by taking the appropriate disjoint unions, a groupoid whose groupoid cardinality converges to x.

Groupoid cardinality

- The *groupoid cardinality* of a groupoid $\mathcal{G}$ is

$$\kappa(\mathcal{G}) = \sum_{[x]\in\underline{\mathcal{G}}} \frac{1}{\#(\mathrm{Aut}(x))},$$

  where $\underline{\mathcal{G}}$ is the set of isomorphism classes of objects of $\mathcal{G}$
- A groupoid is called *tame* if its groupoid cardinality converges.
- Groupoid cardinality agrees with cardinality on finite sets (discrete n-object groupoids)
- A finite group of order n (viewed as a one-object groupoid) has groupoid cardinality $\frac{1}{n}$. So, for any real number x we can construct, by taking the appropriate disjoint unions, a groupoid whose groupoid cardinality converges to x.

Groupoid cardinality

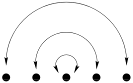
- The *groupoid cardinality* of a groupoid $\mathcal{G}$ is

$$\kappa(\mathcal{G}) = \sum_{[x] \in \underline{\mathcal{G}}} \frac{1}{\#(\mathrm{Aut}(x))},$$

  where $\underline{\mathcal{G}}$ is the set of isomorphism classes of objects of $\mathcal{G}$
- A groupoid is called *tame* if its groupoid cardinality converges.
- Groupoid cardinality agrees with cardinality on finite sets (discrete n-object groupoids)
- A finite group of order n (viewed as a one-object groupoid) has groupoid cardinality $\frac{1}{n}$. So, for any real number x we can construct, by taking the appropriate disjoint unions, a groupoid whose groupoid cardinality converges to x.

$2\frac{1}{2}$

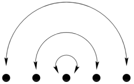
- Let's look once again at our folding in half of [5]:



- Computing the groupoid cardinality of this diagram viewed as a groupoid with identity morphisms "hidden", we obtain:

$$\frac{1}{1}+\frac{1}{1}+\frac{1}{2}$$

- Let's look once again at our folding in half of [5]:



- Computing the groupoid cardinality of this diagram viewed as a groupoid with identity morphisms "hidden", we obtain:

$$\frac{1}{1} + \frac{1}{1} + \frac{1}{2}$$

An "explanation" of why $\frac{5}{2} = 2\frac{1}{2}$

When your students ask you about how fractions work, now you can answer with a clarifying example:

- $\frac{5}{2} = 2\frac{1}{2}$ because $\kappa([5] /\!\!/ (\mathbb{Z}/2\mathbb{Z})) = 2\frac{1}{2}$, where [5] is the discrete groupoid with five objects.

Groupoid cardinality of FinSet_0

Let FinSet_0 be the category whose objects are finite sets and whose morphisms are bijections.

- For any $X \in \mathrm{FinSet}_0$, the automorphisms of X are just the permutations of X. So, $\mathrm{Aut}(X) = S_n$, and hence

$$\kappa(\mathrm{FinSet}_0) = \sum_{[n] \in \underline{\mathrm{FinSet}_0}} \frac{1}{n!} = \sum_{n \in \mathbb{N}} \frac{1}{n!} = e.$$

∞-groupoid cardinality

There is an interesting connection between groupoid cardinality and Euler characteristic:

∞-*Groupoid cardinality*

This is the special case of a more general definition:

The groupoid cardinality of an ∞-groupoid X – equivalently the ***Euler characteristic*** of a topological space X (that's the same, due to the homotopy hypothesis) – is, if it converges, the alternating product of cardinalities of the (simplicial) homotopy groups

$$|X| := \sum_{[x] \in \pi_0(X)} \prod_{k=1}^{\infty} |\pi_k(X,x)|^{(-1)^k} = \sum_{[x]} \frac{1}{|\pi_1(X,x)|} |\pi_2(X,x)| \frac{1}{|\pi_3(X,x)|} |\pi_4(X,x)| \cdots .$$

This corresponds to what is referred to as the *total homotopy order of a space* by Quinn (1995), although similar ideas were explored by several researchers at that time.

(From [15])

"Forgetting stuff"

- Moving to groupoids and working with groupoid cardinality has allowed us to introduce "new structure" which is forgotten when we take $\#(X)$ to count finite sets.
- Let's look further into this notion of "forgetting stuff about a groupoid".

“Forgetting stuff”

- Moving to groupoids and working with groupoid cardinality has allowed us to introduce “new structure” which is forgotten when we take $\#(X)$ to count finite sets.
- Let’s look further into this notion of “forgetting stuff about a groupoid”.

Reminder of forgetful functors

- A *functor* $F : \mathcal{C} \to \mathcal{B}$ is a mapping between categories which maps objects of $\mathcal{C}$ to objects of $\mathcal{B}$, and maps morphisms f, g of $\mathcal{C}$ to morphisms Ff, Fg of $\mathcal{B}$ such that $F(1_c) = 1_{Fc}$ for any object c of $\mathcal{C}$, and $F(g \circ f) = F(g) \circ F(f)$.
- We can think of certain functors as being *forgetful*. For instance, a functor $F : \mathrm{Top} \to \mathrm{Set}$ which sends a topological space to its underlying set "forgets" the topology.

Reminder of forgetful functors

- A *functor* $F : \mathcal{C} \to \mathcal{B}$ is a mapping between categories which maps objects of $\mathcal{C}$ to objects of $\mathcal{B}$, and maps morphisms f, g of $\mathcal{C}$ to morphisms Ff, Fg of $\mathcal{B}$ such that $F(1_c) = 1_{Fc}$ for any object c of $\mathcal{C}$, and $F(g \circ f) = F(g) \circ F(f)$.
- We can think of certain functors as being *forgetful*. For instance, a functor $F : \mathrm{Top} \to \mathrm{Set}$ which sends a topological space to its underlying set "forgets" the topology.

Stuff types

- A *stuff type* is a functor $F : X \to \mathrm{FinSet}_0$, where X is a groupoid.
- We think of the functor F as forgetting extra stuff we can place on finite sets.
- A stuff type F is called a *structure type* when F is faithful, a *property type* when F is full and faithful, a *vacuous property type* when F is full, faithful, and essentially surjective.
- We won't look too closely at stuff types in general, but we will look closely at *structure types*.

Stuff types

- A *stuff type* is a functor $F : X \to \mathrm{FinSet}_0$, where X is a groupoid.
- We think of the functor F as forgetting extra stuff we can place on finite sets.
- A stuff type F is called a *structure type* when F is faithful, a *property type* when F is full and faithful, a *vacuous property type* when F is full, faithful, and essentially surjective.
- We won't look too closely at stuff types in general, but we will look closely at *structure types*.

Stuff types

- A *stuff type* is a functor $F : X \to \mathrm{FinSet}_0$, where X is a groupoid.
- We think of the functor F as forgetting extra stuff we can place on finite sets.
- A stuff type F is called a *structure type* when F is faithful, a *property type* when F is full and faithful, a *vacuous property type* when F is full, faithful, and essentially surjective.
- We won't look too closely at stuff types in general, but we will look closely at *structure types*.

Stuff types

- A *stuff type* is a functor $F : X \to \mathrm{FinSet}_0$, where X is a groupoid.
- We think of the functor F as forgetting extra stuff we can place on finite sets.
- A stuff type F is called a *structure type* when F is faithful, a *property type* when F is full and faithful, a *vacuous property type* when F is full, faithful, and essentially surjective.
- We won't look too closely at stuff types in general, but we will look closely at *structure types*.

"Forgetting stuff"

We say F:

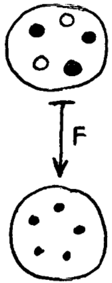
- forgets nothing if it's faithful, full, & essentially surjective
- forgets properties if it's faithful & full
- forgets structure if it's faithful
- forgets stuff always

(From Baez [3])

Example

$$
\begin{tikzcd}
E^2 = [\text{2-colored finite sets}]_0 \arrow[d, "F"'] \\
E = \mathrm{FinSet}_0
\end{tikzcd}
$$

— forget the coloring



(From Baez [3])

A construction on stuff types involving groupoid cardinality

The *power series of a stuff type* is

$$\sigma(F) = \sum_{n \in \mathbb{N}} \kappa(F_n) x^n$$

where F_n is the preimage of $[n]$ under F.

Baez's dissection table of stuff

If F is a...	then $\lvert F\rvert(z) = \sum \frac{a_n}{n!} z^n$ where:	since these numbers are cardinalities of:
stuff type	$a_n \in \mathbb{R}^+ = [0,\infty)$	(tame) groupoids = 1-groupoids
structure type	$a_n \in \mathbb{N}$	(finite) sets = 0-groupoids
property type	$a_n \in \{0,1\} \cong \{F,T\}$	truth values = -1-groupoids
vacuous property type	$a_n \in \{1\} \cong \{T\}$	true = the only -2-groupoid

The reason this all works in this way is that in equipping a finite set with extra $\left\{\begin{array}{l}\text{stuff}\\ \text{structure}\\ \text{properties}\\ \text{vacuous properties}\end{array}\right.$, there's

a $\left\{\begin{array}{l}\text{groupoid}\\ \text{set}\\ \text{pair}\\ \text{1-elt set}\end{array}\right.$, i.e a $\left\{\begin{array}{l}\text{1-groupoid}\\ \text{0-groupoid}\\ \text{-1-groupoid}\\ \text{-2-groupoid}\end{array}\right.$ of choices.

(From Baez [3]) (homotopy distraction: this table is evocative of the Postnikov tower. What could it mean to make an ∞-groupoid of choices? This is pursued in [9])

Structure types are species

- FACT: When F is faithful, *i.e.* F is a structure type, we may view F equivalently as $F^* : \mathrm{FinSet}_0 \to \mathrm{Set}$.[2]

A functor $F: X \to \mathrm{FinSet}_0$ that forgets ____ ...	is the same as
stuff	$F^* : \mathrm{FinSet}_0 \longrightarrow \mathrm{Gpd} = 1\text{-}\mathrm{Gpd}$
structure	$F^* : \mathrm{FinSet}_0 \longrightarrow \mathrm{Set} = 0\text{-}\mathrm{Gpd}$
properties	$F^* : \mathrm{FinSet}_0 \longrightarrow \{\emptyset, 1\} = -1\text{-}\mathrm{Gpd}$
vacuous properties	$F^* : \mathrm{FinSet}_0 \longrightarrow \{1\} = -2\text{-}\mathrm{Gpd}$

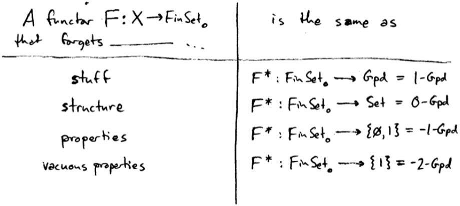
(From Baez [3])

- A functor $F : \mathrm{FinSet}_0 \to \mathrm{Set}$ is precisely the definition, due to Joyal [11], of *combinatorial species*.
- So we were secretly dealing with (a generalization of) combinatorial species all along! For the remainder of the talk, we will look at combinatorial species and their decategorification as generating functions.

[2]To see this worked out, consult John Baez, Quantum Gravity Seminar 2004, Week 4:
`https://math.ucr.edu/home/baez/qg-spring2004/s04week04.pdf`

Structure types are species

- FACT: When F is faithful, *i.e.* F is a structure type, we may view F equivalently as $F^* : \mathrm{FinSet}_0 \to \mathrm{Set}$.[2]



(From Baez [3])

- A functor $F : \mathrm{FinSet}_0 \to \mathrm{Set}$ is precisely the definition, due to Joyal [11], of *combinatorial species*.
- So we were secretly dealing with (a generalization of) combinatorial species all along! For the remainder of the talk, we will look at combinatorial species and their decategorification as generating functions.

[2] To see this worked out, consult John Baez, Quantum Gravity Seminar 2004, Week 4:
`https://math.ucr.edu/home/baez/qg-spring2004/s04week04.pdf`

Species

- A species is a functor $F : \mathrm{FinSet}_0 \to \mathrm{Set}$.
- We speak of "putting an F-structure on a finite set" when we apply F to a finite set X.

Examples of species

The species of sets.

Examples of species

The species of permutations:

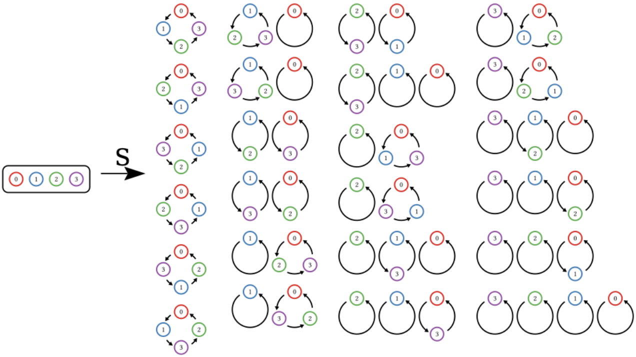


Figure 3.7: The species S of permutations

(From Yorgey [8])

Examples of species

The species of cycles

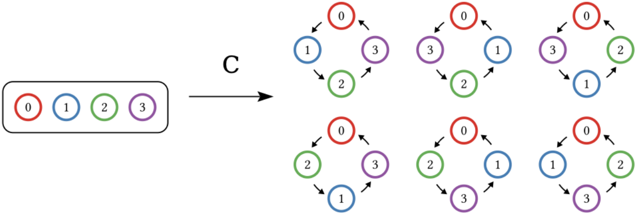


Figure 3.6: The species C of cycles

(From Yorgey [8])

The generating series of a species

Let F_n denote the set of F-structures on the "canonical" n-element set $[n]$. The *generating series* $\sigma(F)$ of a species F is the formal power series

$$\sigma(F) = \sum_{n=0}^{\infty} \frac{\#(F_n) x^n}{n!}.$$

The generating series of a species

- This definition agrees with the definition of the power series of a stuff type (see [14]).
- This suggests that the definition of the generating series of a species is not merely a convenient gadget set up to make the generating functions we want fall out: this definition flows naturally from the categorical constructions we looked at in context with groupoid cardinality and stuff types.

Examples of generating series of species

- Let P be the species of permutations. Then $\#(P_n) = n!$ since there are $n!$ permutations of a set of n elements, and so

$$\sigma(P) = \sum_{n=0}^{\infty} \frac{n! x^n}{n!} = \sum_{n=0}^{\infty} x^n = \frac{1}{1-x}.$$

- Let C be the species of cycles. Then $\#(C_n) = \frac{n!}{n} = (n-1)!$ since there are $n!$ ways to represent a cycle in cycle notation but , and so

$$\sigma(C) = \sum_{n=0}^{\infty} \frac{(n-1)! x^n}{n!} = \sum_{n=0}^{\infty} \frac{x^n}{n} = \log(\frac{1}{1-x}).$$

- Let E be the species of finite sets. Then $\#(E_n) = 1$, and so

$$\sigma(E) = \sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x.$$

Recall that $\kappa(\text{FinSet}) = e$.

Examples of generating series of species

- Let P be the species of permutations. Then $\#(P_n) = n!$ since there are $n!$ permutations of a set of n elements, and so

$$\sigma(P) = \sum_{n=0}^{\infty} \frac{n! x^n}{n!} = \sum_{n=0}^{\infty} x^n = \frac{1}{1-x}.$$

- Let C be the species of cycles. Then $\#(C_n) = \frac{n!}{n} = (n-1)!$ since there are $n!$ ways to represent a cycle in cycle notation but , and so

$$\sigma(C) = \sum_{n=0}^{\infty} \frac{(n-1)! x^n}{n!} = \sum_{n=0}^{\infty} \frac{x^n}{n} = \log(\frac{1}{1-x}).$$

- Let E be the species of finite sets. Then $\#(E_n) = 1$, and so

$$\sigma(E) = \sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x.$$

Recall that $\kappa(\mathrm{FinSet}) = e$.

Examples of generating series of species

- Let P be the species of permutations. Then $\#(P_n) = n!$ since there are $n!$ permutations of a set of n elements, and so

$$\sigma(P) = \sum_{n=0}^{\infty} \frac{n! x^n}{n!} = \sum_{n=0}^{\infty} x^n = \frac{1}{1-x}.$$

- Let C be the species of cycles. Then $\#(C_n) = \frac{n!}{n} = (n-1)!$ since there are $n!$ ways to represent a cycle in cycle notation but , and so

$$\sigma(C) = \sum_{n=0}^{\infty} \frac{(n-1)! x^n}{n!} = \sum_{n=0}^{\infty} \frac{x^n}{n} = \log(\frac{1}{1-x}).$$

- Let E be the species of finite sets. Then $\#(E_n) = 1$, and so

$$\sigma(E) = \sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x.$$

Recall that $\kappa(\mathrm{FinSet}) = e$.

Examples of generating series of species

- Let P be the species of permutations. Then $\#(P_n) = n!$ since there are $n!$ permutations of a set of n elements, and so

$$\sigma(P) = \sum_{n=0}^{\infty} \frac{n! x^n}{n!} = \sum_{n=0}^{\infty} x^n = \frac{1}{1-x}.$$

- Let C be the species of cycles. Then $\#(C_n) = \frac{n!}{n} = (n-1)!$ since there are $n!$ ways to represent a cycle in cycle notation but , and so

$$\sigma(C) = \sum_{n=0}^{\infty} \frac{(n-1)! x^n}{n!} = \sum_{n=0}^{\infty} \frac{x^n}{n} = \log(\frac{1}{1-x}).$$

- Let E be the species of finite sets. Then $\#(E_n) = 1$, and so

$$\sigma(E) = \sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x.$$

Recall that $\kappa(\mathrm{FinSet}) = e$.

Operations on species

Note that here we use $\sqcap, \sqcup$ to denote, respectively, product and disjoint union now *in the category of sets*, rather than of finite sets. Where X denotes an object of FinSet_0:

- The *sum* $F + G$ of species F, G is defined by $(F + G)(X) = F(X) \sqcup G(X)$.
- The *product* $F \cdot G$ of species F, G is defined by $(F \cdot G)(X) = \bigsqcup_{X_1 \sqcup X_2 = X} F(X_1) \sqcap F(X_1)$.
- The *derivative* F' of a species F is defined by $F'(X) = F(X \sqcup \{*\})$, where $\{*\}$ is some one-element set.

Note: There is also a notion of "composition" of species, but its definition is less straightforward than these operations.

Operations on species

Note that here we use $\sqcap, \sqcup$ to denote, respectively, product and disjoint union now *in the category of sets*, rather than of finite sets. Where X denotes an object of FinSet_0:

- The *sum* $F + G$ of species F, G is defined by $(F + G)(X) = F(X) \sqcup G(X)$.
- The *product* $F \cdot G$ of species F, G is defined by $(F \cdot G)(X) = \bigsqcup_{X_1 \sqcup X_2 = X} F(X_1) \sqcap F(X_1)$.
- The *derivative* F' of a species F is defined by $F'(X) = F(X \sqcup \{*\})$, where $\{*\}$ is some one-element set.

Note: There is also a notion of "composition" of species, but its definition is less straightforward than these operations.

Operations on species

Note that here we use $\sqcap, \sqcup$ to denote, respectively, product and disjoint union now *in the category of sets*, rather than of finite sets. Where X denotes an object of FinSet_0:

- The *sum* $F + G$ of species F, G is defined by $(F+G)(X) = F(X) \sqcup G(X)$.
- The *product* $F \cdot G$ of species F, G is defined by $(F \cdot G)(X) = \bigsqcup_{X_1 \sqcup X_2 = X} F(X_1) \sqcap F(X_1)$.
- The *derivative* F' of a species F is defined by $F'(X) = F(X \sqcup \{*\})$, where $\{*\}$ is some one-element set.

Note: There is also a notion of "composition" of species, but its definition is less straightforward than these operations.

Operations on species

Note that here we use $\sqcap, \sqcup$ to denote, respectively, product and disjoint union now *in the category of sets*, rather than of finite sets. Where X denotes an object of FinSet_0:

- The *sum* $F + G$ of species F, G is defined by $(F + G)(X) = F(X) \sqcup G(X)$.
- The *product* $F \cdot G$ of species F, G is defined by $(F \cdot G)(X) = \bigsqcup_{X_1 \sqcup X_2 = X} F(X_1) \sqcap F(X_1)$.
- The *derivative* F' of a species F is defined by $F'(X) = F(X \sqcup \{*\})$, where $\{*\}$ is some one-element set.

Note: There is also a notion of "composition" of species, but its definition is less straightforward than these operations.

Sum of species

We think of putting an $F + G$ structure on A as putting either an F structure on A **or** a G structure on A.

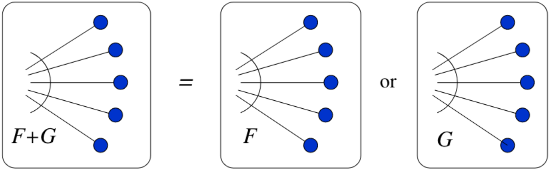


Figure 2.1: A typical structure of species $F + G$.

(From Bergeron, Labelle, Leroux [6])

Example of the sum of species

- Where E is the species of finite sets, E_{even} is the species of finite sets of evenly many elements, and E_{odd} is the species of finite sets of oddly many elements, $E = E_{\text{even}} + E_{\text{odd}}$
- It turns out that $e^x = \sigma(E) = \sigma(E_{\text{even}}) + \sigma(E_{\text{odd}}) = \cosh(x) + \sinh(x)$, see [6].

Example of the sum of species

- Where E is the species of finite sets, E_{even} is the species of finite sets of evenly many elements, and E_{odd} is the species of finite sets of oddly many elements, $E = E_{\text{even}} + E_{\text{odd}}$
- It turns out that $e^x = \sigma(E) = \sigma(E_{\text{even}}) + \sigma(E_{\text{odd}}) = \cosh(x) + \sinh(x)$, see [6].

Product of species

We think of putting an $F \cdot G$ structure on X as first partitioning X into two disjoint sets $X_1 \sqcup X_2$ then putting an F structure on X_1 **and** a G structure on X_2.

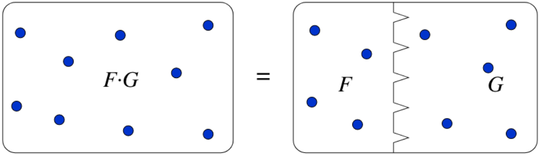


Figure 2.3: A typical product structure.

(From Bergeron, Labelle, Leroux [6])

Examples of the product of species

- We have $S = E \cdot Der$, where E is the species of finite sets, S is the species of permutations, and Der is the species of derangements.

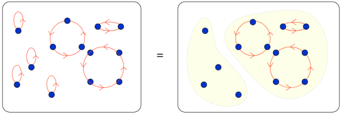


Figure 2.2: A permutation as a set of fixed points together with a derangement. (From Bergeron, Labelle, Leroux [6])

Derivative of species

We think of putting an F' structure on X as putting an F structure on X *with a designated extra point*.

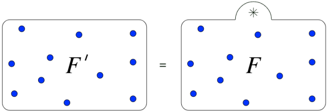


Figure 2.14: A typical structure of species F'.

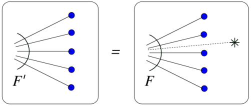


Figure 2.15: Alternate representation of a typical structure of species F'

(From Bergeron, Labelle, Leroux [6])

Examples of the derivative of a species

- The derivative of the species of cycles is the species of lists:

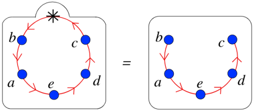


Figure 2.16: Breaking off a cycle at the special point gives a list.

- The derivative of the species of finite sets is itself: $E' = E$. Adding a new point to a finite set yiels another finite set.

The generating series respects species operations

- $\sigma(F + G) = \sigma(F) + \sigma(G) = F(x) + G(x)$
- $\sigma(F \cdot G) = \sigma(F) \times \sigma(G) = F(x) \times G(x)$
- $\sigma(F') = \sigma(F)' = \frac{d}{dx}F(x)$

Details can be found in Bergeron, Labelle, Leroux [5]

Decategorifying species

Thus, combinatorial species decategorify to generating functions.

Further directions

- The relationship between category theory, logic, and computation is called the *Curry-Howard-Lambek correspondence*. The connection between homotopy and computation is a crucial insight in the development of *homotopy type theory*.
- Leinster ([13], [12]) has defined notions of generalized Möbius inversion and Euler characteristic of categories which are closely related to groupoid cardinality.
- Carillo, Kock, and Tonks ([9]) generalize stuff types to ∞-groupoids in the guise of "homotopy linear algebra".
- Aguiar and Mahajan apply *vector species*, a cousin of combinatorial species, to hyperplane arrangements and Hopf algebras: [2] and elsewhere.
- In programming language theory, a notion of *container type* similar to species has been developed, which also has a notion of derivative and an object which "behaves like e": [1]
- *Goodwillie calculus* is another sort of categorification of power series and differentiation: see for instance [10]

Further directions

- The relationship between category theory, logic, and computation is called the *Curry-Howard-Lambek correspondence*. The connection between homotopy and computation is a crucial insight in the development of *homotopy type theory*.
- Leinster ([13], [12]) has defined notions of generalized Möbius inversion and Euler characteristic of categories which are closely related to groupoid cardinality.
- Carillo, Kock, and Tonks ([9]) generalize stuff types to ∞-groupoids in the guise of "homotopy linear algebra".
- Aguiar and Mahajan apply *vector species*, a cousin of combinatorial species, to hyperplane arrangements and Hopf algebras: [2] and elsewhere.
- In programming language theory, a notion of *container type* similar to species has been developed, which also has a notion of derivative and an object which "behaves like e": [1]
- *Goodwillie calculus* is another sort of categorification of power series and differentiation: see for instance [10]

Further directions

- The relationship between category theory, logic, and computation is called the *Curry-Howard-Lambek correspondence*. The connection between homotopy and computation is a crucial insight in the development of *homotopy type theory*.
- Leinster ([13], [12]) has defined notions of generalized Möbius inversion and Euler characteristic of categories which are closely related to groupoid cardinality.
- Carillo, Kock, and Tonks ([9]) generalize stuff types to ∞-groupoids in the guise of "homotopy linear algebra".
- Aguiar and Mahajan apply *vector species*, a cousin of combinatorial species, to hyperplane arrangements and Hopf algebras: [2] and elsewhere.
- In programming language theory, a notion of *container type* similar to species has been developed, which also has a notion of derivative and an object which "behaves like e": [1]
- *Goodwillie calculus* is another sort of categorification of power series and differentiation: see for instance [10]

Further directions

- The relationship between category theory, logic, and computation is called the *Curry-Howard-Lambek correspondence*. The connection between homotopy and computation is a crucial insight in the development of *homotopy type theory*.
- Leinster ([13], [12]) has defined notions of generalized Möbius inversion and Euler characteristic of categories which are closely related to groupoid cardinality.
- Carillo, Kock, and Tonks ([9]) generalize stuff types to ∞-groupoids in the guise of "homotopy linear algebra".
- Aguiar and Mahajan apply *vector species*, a cousin of combinatorial species, to hyperplane arrangements and Hopf algebras: [2] and elsewhere.
- In programming language theory, a notion of *container type* similar to species has been developed, which also has a notion of derivative and an object which "behaves like e": [1]
- *Goodwillie calculus* is another sort of categorification of power series and differentiation: see for instance [10]

Further directions

- The relationship between category theory, logic, and computation is called the *Curry-Howard-Lambek correspondence*. The connection between homotopy and computation is a crucial insight in the development of *homotopy type theory*.
- Leinster ([13], [12]) has defined notions of generalized Möbius inversion and Euler characteristic of categories which are closely related to groupoid cardinality.
- Carillo, Kock, and Tonks ([9]) generalize stuff types to ∞-groupoids in the guise of "homotopy linear algebra".
- Aguiar and Mahajan apply *vector species*, a cousin of combinatorial species, to hyperplane arrangements and Hopf algebras: [2] and elsewhere.
- In programming language theory, a notion of *container type* similar to species has been developed, which also has a notion of derivative and an object which "behaves like e": [1]
- *Goodwillie calculus* is another sort of categorification of power series and differentiation: see for instance [10]

Further directions

- The relationship between category theory, logic, and computation is called the *Curry-Howard-Lambek correspondence*. The connection between homotopy and computation is a crucial insight in the development of *homotopy type theory*.
- Leinster ([13], [12]) has defined notions of generalized Möbius inversion and Euler characteristic of categories which are closely related to groupoid cardinality.
- Carillo, Kock, and Tonks ([9]) generalize stuff types to ∞-groupoids in the guise of "homotopy linear algebra".
- Aguiar and Mahajan apply *vector species*, a cousin of combinatorial species, to hyperplane arrangements and Hopf algebras: [2] and elsewhere.
- In programming language theory, a notion of *container type* similar to species has been developed, which also has a notion of derivative and an object which "behaves like e": [1]
- *Goodwillie calculus* is another sort of categorification of power series and differentiation: see for instance [10]

Bibliography

[1] Michael Abbott et al. "Derivatives of Containers". In: *Typed Lambda Calculi and Applications*. Ed. by Gerhard Goos et al. Vol. 2701. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 16–30. ISBN: 978-3-540-40332-6 978-3-540-44904-1. DOI: 10.1007/3-540-44904-3_2. (Visited on 04/23/2022).

[2] Marcelo Aguiar and Swapneel Mahajan. *Bimonoids for Hyperplane Arrangements*. Encyclopedia of Mathematics and Its Applications 173. Cambridge New York, NY Port Melbourne New Delhi Singapore: Cambridge University Press, 2020. ISBN: 978-1-108-49580-6.

[3] John C. Baez. *Quantum Gravity Seminar - Spring 2004, Quantization and Categorification*. URL: https://math.ucr.edu/home/baez/qg-spring2004/.

[4] John C. Baez and James Dolan. *From Finite Sets to Feynman Diagrams*. Apr. 2000. arXiv: math/0004133. (Visited on 02/21/2023).

[5] F. Bergeron, Gilbert Labelle, and P. Leroux. *Combinatorial Species and Tree-like Structures*. Encyclopedia of Mathematics and Its Applications v. 67. Cambridge ; New York, NY, USA: Cambridge University Press, 1998. ISBN: 978-0-521-57323-8.

[6] F. Bergeron, Gilbert Labelle, and P. Leroux. *Introduction to the Theory of Species of Structures*. URL: https://bergeron.math.uqam.ca/wp-content/uploads/2013/11/book.pdf.

[7] Julia E. Bergner and Christopher D. Walker. "Groupoid Cardinality and Egyptian Fractions". In: *The College Mathematics Journal* 46.2 (Mar. 2015), pp. 122–129. ISSN: 0746-8342, 1931-1346. DOI: 10.4169/college.math.j.46.2.122. (Visited on 10/04/2023).

[8] Brent Yorgey. "COMBINATORIAL SPECIES AND LABELLED STRUCTURES". PhD thesis.

[9] Imma Gálvez-Carrillo, Joachim Kock, and Andrew Tonks. "Homotopy Linear Algebra". In: *Proceedings of the Royal Society of Edinburgh: Section A Mathematics* 148.2 (Apr. 2018), pp. 293–325. ISSN: 0308-2105, 1473-7124. DOI: 10.1017/S0308210517000208. arXiv: 1602.05082 [math]. (Visited on 02/23/2023).

[10] Thomas G Goodwillie. "Calculus III: Taylor Series". In: *Geometry & Topology* 7.2 (Oct. 2003), pp. 645–711. ISSN: 1364-0380, 1465-3060. DOI: 10.2140/gt.2003.7.645. (Visited on 04/05/2023).

[11] André Joyal. "Une théorie combinatoire des séries formelles". In: *Advances in Mathematics* 42.1 (Oct. 1981), pp. 1–82. ISSN: 00018708. DOI: 10.1016/0001-8708(81)90052-9. (Visited on 12/25/2020).

[12] Tom Leinster. "Notions of Möbius Inversion". In: *Bulletin of the Belgian Mathematical Society - Simon Stevin* 19.5 (Dec. 2012). ISSN: 1370-1444. DOI: 10.36045/bbms/1354031556. (Visited on 01/14/2023).

[13] Tom Leinster. *The Euler Characteristic of a Category*. Oct. 2006. arXiv: math/0610260. (Visited on 02/21/2023).

[14] Jeffrey Morton. *Categorified Algebra and Quantum Mechanics*. Jan. 2006. arXiv: math/0601458. (Visited on 10/11/2023).

[15] nLab authors. *groupoid cardinality*. https://ncatlab.org/nlab/show/groupoid+cardinality. Revision 30. Oct. 2023.

[16] Lior Yanovski. *Homotopy Cardinality via Extrapolation of Morava-Euler Characteristics*. Mar. 2023. DOI: 10.48550/arXiv.2303.02603. arXiv: 2303.02603 [math]. (Visited on 07/07/2023).