Graphical abelian logic

David I. Spivak* and Brendan Fong

July 11, 2019

Outline

Abelian categories

Definition

A category $\mathcal{A}$ is *abelian* if

- it has a zero object 0;
- every pair of objects has a product and a coproduct;
- every morphism has a kernel and a cokernel; and
- every monomorphism is a kernel and every epimorphism is a cokernel.

This is a standard definition; we'll see a graphical presentation soon.

Abelian categories

Definition

A category $\mathcal{A}$ is *abelian* if

- it has a zero object 0;
- every pair of objects has a product and a coproduct;
- every morphism has a kernel and a cokernel; and
- every monomorphism is a kernel and every epimorphism is a cokernel.

This is a standard definition; we'll see a graphical presentation soon.

Examples: Ab, fgAb,

Abelian categories

Definition

A category $\mathcal{A}$ is *abelian* if

- it has a zero object 0;
- every pair of objects has a product and a coproduct;
- every morphism has a kernel and a cokernel; and
- every monomorphism is a kernel and every epimorphism is a cokernel.

This is a standard definition; we'll see a graphical presentation soon.

Examples: Ab, fgAb, $\mathrm{Vect}_{\mathbb{R}}$,

Abelian categories

Definition

A category $\mathcal{A}$ is *abelian* if

- it has a zero object 0;
- every pair of objects has a product and a coproduct;
- every morphism has a kernel and a cokernel; and
- every monomorphism is a kernel and every epimorphism is a cokernel.

This is a standard definition; we'll see a graphical presentation soon.

Examples: Ab, fgAb, $\mathrm{Vect}_{\mathbb{R}}$, sheaves of abelian groups on a space, ...

Why abelian categories are beloved

Abelian cats $\mathcal{A}$ are beloved because they are good for computation.

Why abelian categories are beloved

Abelian cats $\mathcal{A}$ are beloved because they are good for computation.

- $\mathcal{A}$ has biproducts! i.e. the canonical map $A \sqcup B \to A \times B$ is iso.
  - It follows that morphisms in $\mathcal{A}$ can be assembled into matrices.
  - Composition is matrix mult., biproduct is "block diagonal".

Why abelian categories are beloved

Abelian cats $\mathcal{A}$ are beloved because they are good for computation.

- $\mathcal{A}$ has biproducts! i.e. the canonical map $A \sqcup B \to A \times B$ is iso.
  - It follows that morphisms in $\mathcal{A}$ can be assembled into matrices.
  - Composition is matrix mult., biproduct is "block diagonal".
- For every object $A \in \mathcal{A}$, the subobjects form a lattice $\mathrm{Sub}(A)$.
  - $\mathrm{Sub}(A)$ has *meets* ($\wedge$) "intersection", *top* ($\top$) "all of A",
  - *joins* ($\vee$) "span", and *bottom* ($\perp$) "zero"

Why abelian categories are beloved

Abelian cats $\mathcal{A}$ are beloved because they are good for computation.

- $\mathcal{A}$ has biproducts! i.e. the canonical map $A \sqcup B \to A \times B$ is iso.
  - It follows that morphisms in $\mathcal{A}$ can be assembled into matrices.
  - Composition is matrix mult., biproduct is "block diagonal".
- For every object $A \in \mathcal{A}$, the subobjects form a lattice $\mathrm{Sub}(A)$.
  - $\mathrm{Sub}(A)$ has *meets* ($\wedge$) "intersection", *top* ($\top$) "all of A",
  - *joins* ($\vee$) "span", and *bottom* ($\bot$) "zero"
- Every morphism $f\colon A \to B$ in $\mathcal{A}$ has an image $A \twoheadrightarrow \mathrm{im}(f) \rightarrowtail B$.

Why abelian categories are beloved

Abelian cats $\mathcal{A}$ are beloved because they are good for computation.

- $\mathcal{A}$ has biproducts! i.e. the canonical map $A \sqcup B \to A \times B$ is iso.
  - It follows that morphisms in $\mathcal{A}$ can be assembled into matrices.
  - Composition is matrix mult., biproduct is "block diagonal".
- For every object $A \in \mathcal{A}$, the subobjects form a lattice $\mathrm{Sub}(A)$.
  - $\mathrm{Sub}(A)$ has *meets* ($\wedge$) "intersection", *top* ($\top$) "all of A",
  - *joins* ($\vee$) "span", and *bottom* ($\perp$) "zero"
- Every morphism $f\colon A \to B$ in $\mathcal{A}$ has an image $A \twoheadrightarrow \mathrm{im}(f) \rightarrowtail B$.

Biggest math application: homological algebra can be done in $\mathcal{A}$.

- A *chain complex* in $\mathcal{A}$ is a sequence of maps, s.t. wherever you look

$$\cdots \to A \xrightarrow{f} B \xrightarrow{g} C \to \cdots$$

you have $\mathrm{im}(f) \subseteq \ker(g)$. Then the *homology* there is $\ker(g)/\mathrm{im}(f)$.

Plan for the talk

In this talk, I'll discuss abelian categories from a totally different angle.

- Usual perspective: a category with four axioms.
- Graphical perspective: the syntactic category of a graphical language.

Plan for the talk

In this talk, I'll discuss abelian categories from a totally different angle.

- Usual perspective: a category with four axioms.
- Graphical perspective: the syntactic category of a graphical language.

Plan:

- Show pictures of the graphical language in action for f.g. ab. groups

Plan for the talk

In this talk, I'll discuss abelian categories from a totally different angle.

- Usual perspective: a category with four axioms.
- Graphical perspective: the syntactic category of a graphical language.

Plan:

- Show pictures of the graphical language in action for f.g. ab. groups
- Explain what these pictures mean.

Plan for the talk

In this talk, I'll discuss abelian categories from a totally different angle.

- Usual perspective: a category with four axioms.
- Graphical perspective: the syntactic category of a graphical language.

Plan:

- Show pictures of the graphical language in action for f.g. ab. groups
- Explain what these pictures mean.
- Explain the main theorem—stated below—and finally conclude.

Plan for the talk

In this talk, I'll discuss abelian categories from a totally different angle.

- Usual perspective: a category with four axioms.
- Graphical perspective: the syntactic category of a graphical language.

Plan:

- Show pictures of the graphical language in action for f.g. ab. groups
- Explain what these pictures mean.
- Explain the main theorem—stated below—and finally conclude.

Theorem (Fong-S.)

Abelian categories are reflective in the 2-category of abelian calculi,

$$\mathbb{A}\mathrm{bCalc} \underset{\mathrm{Prd}}{\overset{\mathrm{Syn}}{\rightleftarrows}} \mathbb{A}\mathrm{bCat}. \quad (\Rightarrow)$$

In particular for $\mathcal{A} \in \mathbb{A}\mathrm{bCat}$*, the unit* $\mathcal{A} \xrightarrow{\cong} \mathrm{SynPrd}(\mathcal{A})$ *is an equivalence.*

Outline

Graphical languages in category theory

String diagrams for (traced) monoidal categories were invented by Joyal and Street (and Verity).

Graphical languages in category theory

String diagrams for (traced) monoidal categories were invented by Joyal and Street (and Verity).

- Defined in terms of topological spaces and homotopies.
- See Selinger's "A survey of graphical languages for monoidal cats"

Graphical languages in category theory

String diagrams for (traced) monoidal categories were invented by Joyal and Street (and Verity).

- Defined in terms of topological spaces and homotopies.
- See Selinger's "A survey of graphical languages for monoidal cats"

Can also be defined combinatorially using lax monoidal functors.

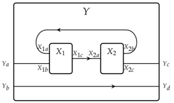
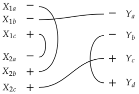
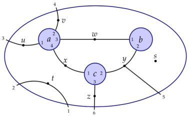
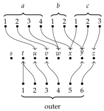
Graphical languages in category theory

String diagrams for (traced) monoidal categories were invented by Joyal and Street (and Verity).

- Defined in terms of topological spaces and homotopies.
- See Selinger's "A survey of graphical languages for monoidal cats"

Can also be defined combinatorially using lax monoidal functors.

- Lax functors Cob $\to$ Set give traced monoidal categories.
- Lax monoidal functors Cospan $\to$ Set give hypergraph categories.

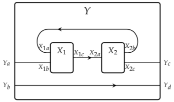
Graphical languages in category theory

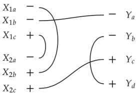
String diagrams for (traced) monoidal categories were invented by Joyal and Street (and Verity).

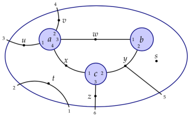
- Defined in terms of topological spaces and homotopies.
- See Selinger's "A survey of graphical languages for monoidal cats"

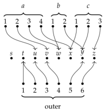
Can also be defined combinatorially using lax monoidal functors.

- Lax functors Cob → Set give traced monoidal categories.
- Lax monoidal functors Cospan → Set give hypergraph categories.









- Brendan talked about how to get regular categories this way.
- Today: abelian categories this way.

Introducing the po-prop of abelian relations

We will be discussing a graphical syntax for abelian categories.

- The graphical syntax is governed by a certain kind of *monoidal theory*.

Introducing the po-prop of abelian relations

We will be discussing a graphical syntax for abelian categories.

- The graphical syntax is governed by a certain kind of *monoidal theory*.
- A *prop* is a strict monoidal category whose object monoid is $(\mathbb{N}, 0, +)$.
- A *po-prop* $\mathbb{P}$ is a locally-posetal version: $\mathbb{P}(m, n) \in \mathsf{Poset}$.

Introducing the po-prop of abelian relations

We will be discussing a graphical syntax for abelian categories.

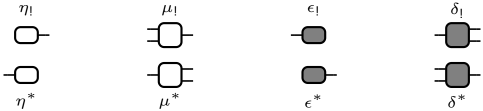
- The graphical syntax is governed by a certain kind of *monoidal theory*.
- A *prop* is a strict monoidal category whose object monoid is $(\mathbb{N}, 0, +)$.
- A *po-prop* $\mathbb{P}$ is a locally-posetal version: $\mathbb{P}(m, n) \in \text{Poset}$.
- Think of the maps in $\mathbb{P}$ as icons we can use in our graphical language.

Introducing the po-prop of abelian relations

We will be discussing a graphical syntax for abelian categories.

- The graphical syntax is governed by a certain kind of *monoidal theory*.
- A *prop* is a strict monoidal category whose object monoid is $(\mathbb{N}, 0, +)$.
- A *po-prop* $\mathbb{P}$ is a locally-posetal version: $\mathbb{P}(m, n) \in \text{Poset}$.
- Think of the maps in $\mathbb{P}$ as icons we can use in our graphical language.

The po-prop $\mathbb{A}$ of *abelian relations* has eight generating 1-morphisms:

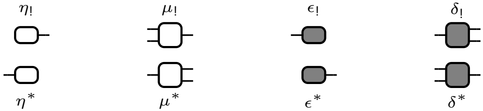


Introducing the po-prop of abelian relations

We will be discussing a graphical syntax for abelian categories.

- The graphical syntax is governed by a certain kind of *monoidal theory*.
- A *prop* is a strict monoidal category whose object monoid is $(\mathbb{N}, 0, +)$.
- A *po-prop* $\mathbb{P}$ is a locally-posetal version: $\mathbb{P}(m, n) \in$ Poset.
- Think of the maps in $\mathbb{P}$ as icons we can use in our graphical language.

The po-prop $\mathbb{A}$ of *abelian relations* has eight generating 1-morphisms:



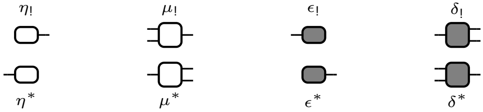
Intuition: icons $m \to n$ are maps between subsp's of $\mathbb{R}^m$ and subsp's of $\mathbb{R}^n$

Introducing the po-prop of abelian relations

We will be discussing a graphical syntax for abelian categories.

- The graphical syntax is governed by a certain kind of *monoidal theory*.
- A *prop* is a strict monoidal category whose object monoid is $(\mathbb{N}, 0, +)$.
- A *po-prop* $\mathbb{P}$ is a locally-posetal version: $\mathbb{P}(m, n) \in \text{Poset}$.
- Think of the maps in $\mathbb{P}$ as icons we can use in our graphical language.

The po-prop $\mathbb{A}$ of *abelian relations* has eight generating 1-morphisms:



Intuition: icons $m \to n$ are maps between subsp's of $\mathbb{R}^m$ and subsp's of $\mathbb{R}^n$

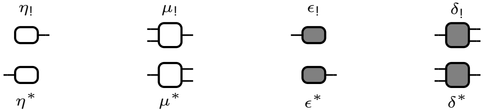
- All come in adjoint pairs;
- η is "0", μ is "+", ϵ is "everything", δ is "equality"; as for Pawel.

Introducing the po-prop of abelian relations

We will be discussing a graphical syntax for abelian categories.

- The graphical syntax is governed by a certain kind of *monoidal theory*.
- A *prop* is a strict monoidal category whose object monoid is $(\mathbb{N}, 0, +)$.
- A *po-prop* $\mathbb{P}$ is a locally-posetal version: $\mathbb{P}(m, n) \in \text{Poset}$.
- Think of the maps in $\mathbb{P}$ as icons we can use in our graphical language.

The po-prop $\mathbb{A}$ of *abelian relations* has eight generating 1-morphisms:



Intuition: icons $m \to n$ are maps between subsp's of $\mathbb{R}^m$ and subsp's of $\mathbb{R}^n$

- All come in adjoint pairs;
- η is "0", μ is "+", ϵ is "everything", δ is "equality"; as for Pawel.
- Example:  projects onto a coordinate plane,  intersects with it.

Some terms in the graphical language

Given a map $f\colon X \to Y$, its cokernel and kernel are canonical maps:

- Cokernel: $Y \twoheadrightarrow Y/\operatorname{im}(f)$. "Add $\operatorname{im}(f)$-valued noise to data in Y."

Some terms in the graphical language

Given a map $f\colon X \to Y$, its cokernel and kernel are canonical maps:

- Cokernel: $Y \twoheadrightarrow Y/\operatorname{im}(f)$. "Add $\operatorname{im}(f)$-valued noise to data in Y."
- Kernel: $\{x : X \mid f(x) = 0\} \rightarrowtail X$. "Select data in X with null f."

Some terms in the graphical language

Given a map $f\colon X \to Y$, its cokernel and kernel are canonical maps:

- Cokernel: $Y \twoheadrightarrow Y/\operatorname{im}(f)$. "Add $\operatorname{im}(f)$-valued noise to data in Y."
- Kernel: $\{x : X \mid f(x) = 0\} \rightarrowtail X$. "Select data in X with null f."

In pictures...

$f\colon X \to Y$,
cokernel
and kernel:

Some terms in the graphical language

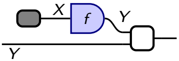
Given a map $f\colon X \to Y$, its cokernel and kernel are canonical maps:

- Cokernel: $Y \twoheadrightarrow Y/\operatorname{im}(f)$. "Add $\operatorname{im}(f)$-valued noise to data in Y."
- Kernel: $\{x : X \mid f(x) = 0\} \rightarrowtail X$. "Select data in X with null f."

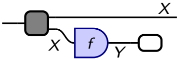
In pictures...

$f\colon X \to Y$, cokernel and kernel:

$Y \to \operatorname{coker}(f)$



$\ker(f) \to X$



Some terms in the graphical language

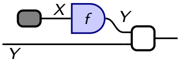
Given a map $f\colon X \to Y$, its cokernel and kernel are canonical maps:

- Cokernel: $Y \twoheadrightarrow Y/\operatorname{im}(f)$. "Add $\operatorname{im}(f)$-valued noise to data in Y."
- Kernel: $\{x : X \mid f(x) = 0\} \rightarrowtail X$. "Select data in X with null f."

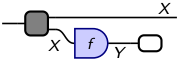
In pictures...

$f\colon X \to Y$, cokernel and kernel:

$Y \to \operatorname{coker}(f)$



$\ker(f) \to X$



Snake lemma connecting homomorphism:

Some terms in the graphical language

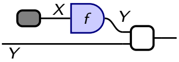
Given a map $f\colon X \to Y$, its cokernel and kernel are canonical maps:

- Cokernel: $Y \twoheadrightarrow Y/\operatorname{im}(f)$. "Add $\operatorname{im}(f)$-valued noise to data in Y."
- Kernel: $\{x : X \mid f(x) = 0\} \rightarrowtail X$. "Select data in X with null f."

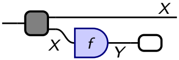
In pictures...

$f\colon X \to Y$, cokernel and kernel:

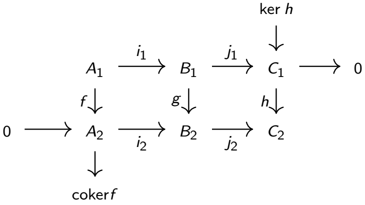
$Y \to \operatorname{coker}(f)$



$\ker(f) \to X$



Snake lemma connecting homomorphism:



Some terms in the graphical language

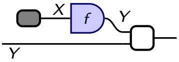
Given a map $f\colon X \to Y$, its cokernel and kernel are canonical maps:

- Cokernel: $Y \twoheadrightarrow Y/\operatorname{im}(f)$. "Add $\operatorname{im}(f)$-valued noise to data in Y."
- Kernel: $\{x : X \mid f(x) = 0\} \rightarrowtail X$. "Select data in X with null f."

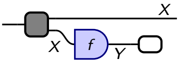
In pictures...

$f\colon X \to Y$, cokernel and kernel:

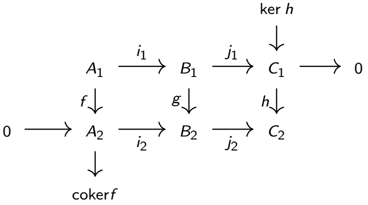
$Y \to \operatorname{coker}(f)$

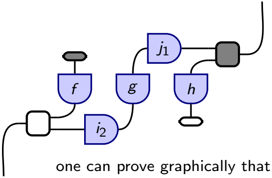


$\ker(f) \to X$



Snake lemma connecting homomorphism:





one can prove graphically that this relation is a left adjoint

The po-prop of abelian relations

The po-prop $\mathbb{A}$ of abelian relations has eight generating 1-morphisms:

such that: $(\eta_!, \mu_!, \eta^*, \mu^*)$ is an adjoint Frobenius monoid;

The po-prop of abelian relations

The po-prop $\mathbb{A}$ of abelian relations has eight generating 1-morphisms:

such that: $(\eta_!, \mu_!, \eta^*, \mu^*)$ is an adjoint Frobenius monoid; $(\epsilon_!, \delta_!, \epsilon^*, \mu^*)$ is an adjoint Frobenius comonoid,

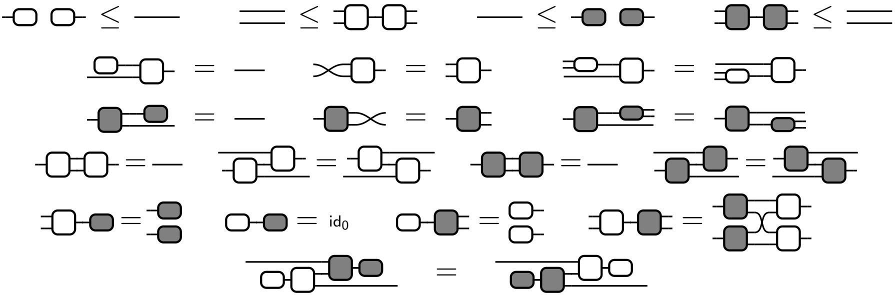
the left (iff right) adjoints form a bimonoid, mediating isomorphism is an involution:

The po-prop of abelian relations

The po-prop $\mathbb{A}$ of abelian relations has eight generating 1-morphisms:

such that: $(\eta_!, \mu_!, \eta^*, \mu^*)$ is an adjoint Frobenius monoid; $(\epsilon_!, \delta_!, \epsilon^*, \mu^*)$ is an adjoint Frobenius comonoid,

the left (iff right) adjoints form a bimonoid, mediating isomorphism is an involution:

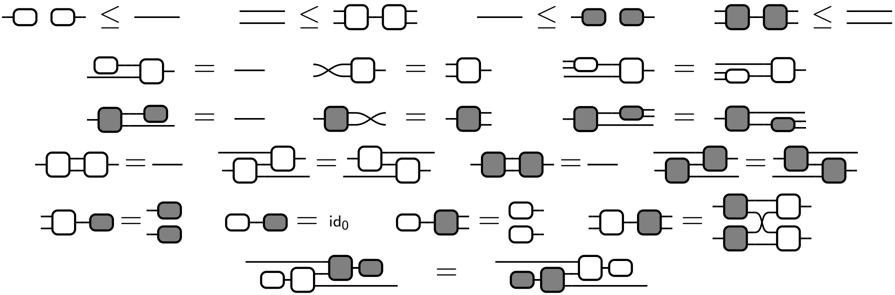


The po-prop of abelian relations

The po-prop $\mathbb{A}$ of abelian relations has eight generating 1-morphisms:

such that: $(\eta_!, \mu_!, \eta^*, \mu^*)$ is an adjoint Frobenius monoid; $(\epsilon_!, \delta_!, \epsilon^*, \mu^*)$ is an adjoint Frobenius comonoid,

the left (iff right) adjoints form a bimonoid, mediating isomorphism is an involution:



Better characterization?

Drawing functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Let $\mathbb{A}$ be as above; it has objects $\mathbb{N}$ and morphisms generated by the icons

Drawing functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Let $\mathbb{A}$ be as above; it has objects $\mathbb{N}$ and morphisms generated by the icons

Let $\mathbb{P}\mathrm{oset}$ be the symmetric monoidal po-category where:

- objects are partially ordered sets $(S, \leq)$,
- 1-morphisms $f\colon S \to T$ are monotone functions, a.k.a. functors,
- 2-morphisms $\alpha\colon f \to g$ are natural transformations.
- Cartesian monoidal structure: unit is 1, product is $\times$.

Drawing functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Let $\mathbb{A}$ be as above; it has objects $\mathbb{N}$ and morphisms generated by the icons

Let $\mathbb{P}\mathrm{oset}$ be the symmetric monoidal po-category where:

- objects are partially ordered sets $(S, \leq)$,
- 1-morphisms $f\colon S \to T$ are monotone functions, a.k.a. functors,
- 2-morphisms $\alpha\colon f \to g$ are natural transformations.
- Cartesian monoidal structure: unit is 1, product is $\times$.

Main interest: lax monoidal po-functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$. What's one do?

Drawing functors $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$

Let $\mathbb{A}$ be as above; it has objects $\mathbb{N}$ and morphisms generated by the icons

Let $\mathbb{P}\text{oset}$ be the symmetric monoidal po-category where:

- objects are partially ordered sets $(S, \leq)$,
- 1-morphisms $f\colon S \to T$ are monotone functions, a.k.a. functors,
- 2-morphisms $\alpha\colon f \to g$ are natural transformations.
- Cartesian monoidal structure: unit is 1, product is $\times$.

Main interest: lax monoidal po-functors $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$. What's one do?

- It assigns a poset $P(n)$ to each object $n \in \mathbb{N} = \text{Ob}(\mathbb{A})$,

Drawing functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Let $\mathbb{A}$ be as above; it has objects $\mathbb{N}$ and morphisms generated by the icons

Let $\mathbb{P}\mathrm{oset}$ be the symmetric monoidal po-category where:

- objects are partially ordered sets $(S, \leq)$,
- 1-morphisms $f\colon S \to T$ are monotone functions, a.k.a. functors,
- 2-morphisms $\alpha\colon f \to g$ are natural transformations.
- Cartesian monoidal structure: unit is 1, product is $\times$.

Main interest: lax monoidal po-functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$. What's one do?

- It assigns a poset $P(n)$ to each object $n \in \mathbb{N} = \mathrm{Ob}(\mathbb{A})$,
- It assigns a monotone map $P(\iota)\colon P(m) \to P(n)$ for each icon $\iota \in \mathbb{A}$,

Drawing functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Let $\mathbb{A}$ be as above; it has objects $\mathbb{N}$ and morphisms generated by the icons

Let $\mathbb{P}\mathrm{oset}$ be the symmetric monoidal po-category where:

- objects are partially ordered sets $(S, \leq)$,
- 1-morphisms $f\colon S \to T$ are monotone functions, a.k.a. functors,
- 2-morphisms $\alpha\colon f \to g$ are natural transformations.
- Cartesian monoidal structure: unit is 1, product is $\times$.

Main interest: lax monoidal po-functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$. What's one do?

- It assigns a poset $P(n)$ to each object $n \in \mathbb{N} = \mathrm{Ob}(\mathbb{A})$,
- It assigns a monotone map $P(\iota)\colon P(m) \to P(n)$ for each icon $\iota \in \mathbb{A}$,
- It assigns maps $1 \to P(0)$ and $P(m_1) \times P(m_2) \to P(m_1 + m_2)$,

Drawing functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Let $\mathbb{A}$ be as above; it has objects $\mathbb{N}$ and morphisms generated by the icons

Let $\mathbb{P}\mathrm{oset}$ be the symmetric monoidal po-category where:

- objects are partially ordered sets $(S, \leq)$,
- 1-morphisms $f\colon S \to T$ are monotone functions, a.k.a. functors,
- 2-morphisms $\alpha\colon f \to g$ are natural transformations.
- Cartesian monoidal structure: unit is 1, product is $\times$.

Main interest: lax monoidal po-functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$. What's one do?

- It assigns a poset $P(n)$ to each object $n \in \mathbb{N} = \mathrm{Ob}(\mathbb{A})$,
- It assigns a monotone map $P(\iota)\colon P(m) \to P(n)$ for each icon $\iota \in \mathbb{A}$,
- It assigns maps $1 \to P(0)$ and $P(m_1) \times P(m_2) \to P(m_1 + m_2)$,
- It obeys all equations; ineq's $\iota \leq \iota'$ in $\mathbb{A}$ sent to nat.trans. in $\mathbb{P}\mathrm{oset}$.

Drawing functors $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$

Let $\mathbb{A}$ be as above; it has objects $\mathbb{N}$ and morphisms generated by the icons

Let $\mathbb{P}$oset be the symmetric monoidal po-category where:

- objects are partially ordered sets $(S, \leq)$,
- 1-morphisms $f\colon S \to T$ are monotone functions, a.k.a. functors,
- 2-morphisms $\alpha\colon f \to g$ are natural transformations.
- Cartesian monoidal structure: unit is 1, product is $\times$.

Main interest: lax monoidal po-functors $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$. What's one do?

- It assigns a poset $P(n)$ to each object $n \in \mathbb{N} = \text{Ob}(\mathbb{A})$,
- It assigns a monotone map $P(\iota)\colon P(m) \to P(n)$ for each icon $\iota \in \mathbb{A}$,
- It assigns maps $1 \to P(0)$ and $P(m_1) \times P(m_2) \to P(m_1 + m_2)$,
- It obeys all equations; ineq's $\iota \leq \iota'$ in $\mathbb{A}$ sent to nat.trans. in $\mathbb{P}$oset.

Let's see one in action.

Finitely-generated abelian groups

Recall the abelian category fgAb of finitely-generated abelian groups.

- Its most important object is $\mathbb{Z}$.
- Every other object is isomorphic to a quotient of a finite sum of $\mathbb{Z}$'s.

Finitely-generated abelian groups

Recall the abelian category fgAb of finitely-generated abelian groups.

- Its most important object is $\mathbb{Z}$.
- Every other object is isomorphic to a quotient of a finite sum of $\mathbb{Z}$'s.

Let's use fgAb to build a lax monoidal po-functor $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$.

- For each $n \in \text{Ob}(\mathcal{A})$, let $P(n) := \text{Sub}(\mathbb{Z}^n)$. On morphisms?

Finitely-generated abelian groups

Recall the abelian category fgAb of finitely-generated abelian groups.

- Its most important object is $\mathbb{Z}$.
- Every other object is isomorphic to a quotient of a finite sum of $\mathbb{Z}$'s.

Let's use fgAb to build a lax monoidal po-functor $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$.

- For each $n \in \text{Ob}(\mathcal{A})$, let $P(n) := \text{Sub}(\mathbb{Z}^n)$. On morphisms?

- Define $P(\eta_!)\colon \text{Sub}(\mathbb{Z}^0) \to \text{Sub}(\mathbb{Z}^1)$ to be $1 \mapsto \perp$, "zero" subspace.

Finitely-generated abelian groups

Recall the abelian category fgAb of finitely-generated abelian groups.

- Its most important object is $\mathbb{Z}$.
- Every other object is isomorphic to a quotient of a finite sum of $\mathbb{Z}$'s.

Let's use fgAb to build a lax monoidal po-functor $P \colon \mathbb{A} \to \mathbb{P}\text{oset}$.

- For each $n \in \text{Ob}(\mathcal{A})$, let $P(n) := \text{Sub}(\mathbb{Z}^n)$. On morphisms?

- Define $P(\eta_!) \colon \text{Sub}(\mathbb{Z}^0) \to \text{Sub}(\mathbb{Z}^1)$ to be $1 \mapsto \bot$, "zero" subspace.
- Define $P(\mu_!) \colon \text{Sub}(\mathbb{Z}^2) \to \text{Sub}(\mathbb{Z})$ by $R \mapsto \{x + y \mid (x, y) \in R\}$.

Finitely-generated abelian groups

Recall the abelian category fgAb of finitely-generated abelian groups.

- Its most important object is $\mathbb{Z}$.
- Every other object is isomorphic to a quotient of a finite sum of $\mathbb{Z}$'s.

Let's use fgAb to build a lax monoidal po-functor $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$.

- For each $n \in \mathrm{Ob}(\mathcal{A})$, let $P(n) := \mathrm{Sub}(\mathbb{Z}^n)$. On morphisms?

- Define $P(\eta_!)\colon \mathrm{Sub}(\mathbb{Z}^0) \to \mathrm{Sub}(\mathbb{Z}^1)$ to be $1 \mapsto \bot$, "zero" subspace.
- Define $P(\mu_!)\colon \mathrm{Sub}(\mathbb{Z}^2) \to \mathrm{Sub}(\mathbb{Z})$ by $R \mapsto \{x + y \mid (x, y) \in R\}$.
- Of course $P(\eta^*)$ and $P(\epsilon_!)$ are the unique function $\mathrm{Sub}(\mathbb{Z}^1) \to 1$.

Finitely-generated abelian groups

Recall the abelian category fgAb of finitely-generated abelian groups.

- Its most important object is $\mathbb{Z}$.
- Every other object is isomorphic to a quotient of a finite sum of $\mathbb{Z}$'s.

Let's use fgAb to build a lax monoidal po-functor $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$.

- For each $n \in \text{Ob}(\mathcal{A})$, let $P(n) := \text{Sub}(\mathbb{Z}^n)$. On morphisms?

- Define $P(\eta_!)\colon \text{Sub}(\mathbb{Z}^0) \to \text{Sub}(\mathbb{Z}^1)$ to be $1 \mapsto \bot$, "zero" subspace.
- Define $P(\mu_!)\colon \text{Sub}(\mathbb{Z}^2) \to \text{Sub}(\mathbb{Z})$ by $R \mapsto \{x + y \mid (x, y) \in R\}$.
- Of course $P(\eta^*)$ and $P(\epsilon_!)$ are the unique function $\text{Sub}(\mathbb{Z}^1) \to 1$.
- Define $P(\mu^*)\colon \text{Sub}(\mathbb{Z}) \to \text{Sub}(\mathbb{Z}^2)$ by $R \mapsto \{(x, y) \mid x + y \in R\}$.
- Define $P(\delta_!)\colon \text{Sub}(\mathbb{Z}) \to \text{Sub}(\mathbb{Z}^2)$ by $R \mapsto \{(x, x) \mid x \in R\}$.
- Define $P(\epsilon^*)\colon 1 \to \text{Sub}(\mathbb{Z})$ by $1 \mapsto \top$.
- Define $P(\delta^*)\colon \text{Sub}(\mathbb{Z}^2) \to \text{Sub}(\mathbb{Z})$ by $R \mapsto \{x \mid (x, x) \in R\}$.

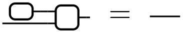
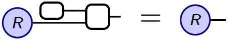
$\mathrm{Sub}(\mathbb{Z}^{-})\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ is a lax monoidal po-functor

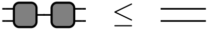
The assignment $P(n) := \mathrm{Sub}(\mathbb{Z}^n)$ as above is a lax monoidal po-functor.

- $\mathrm{Sub}(\mathbb{Z}^{-})$ is lax monoidal:
  - Product gives a map $\times\colon \mathrm{Sub}(\mathbb{Z}^m) \times \mathrm{Sub}(\mathbb{Z}^n) \to \mathrm{Sub}(\mathbb{Z}^{m+n})$.
  - There is a unique map $1 \to \mathrm{Sub}(\mathbb{Z}^0)$.

$\mathrm{Sub}(\mathbb{Z}^-)\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ is a lax monoidal po-functor

The assignment $P(n) \coloneqq \mathrm{Sub}(\mathbb{Z}^n)$ as above is a lax monoidal po-functor.

- $\mathrm{Sub}(\mathbb{Z}^-)$ is lax monoidal:
  - Product gives a map $\times\colon \mathrm{Sub}(\mathbb{Z}^m) \times \mathrm{Sub}(\mathbb{Z}^n) \to \mathrm{Sub}(\mathbb{Z}^{m+n})$.
  - There is a unique map $1 \to \mathrm{Sub}(\mathbb{Z}^0)$.
- $\mathrm{Sub}(\mathbb{Z}^-)$ is 2-functorial:
  - All equations in $\mathbb{A}$ are preserved.  translates to 

, i.e. $\{x + y \mid x \in R \text{ and } y = 0\} = R$.
  - Inequalities are preserved.  translates to $\{(x, y, x', y') \mid (x, y) \in R \wedge (x = y = x' = y')\} \subseteq R$

$\mathsf{Sub}(\mathbb{Z}^-)$ is bi-ajax and preserves involutions

$\mathsf{Sub}(\mathbb{Z}^-)\colon \mathbb{A} \to \mathbb{P}$oset is in fact *bi-ajax* (bi-adjoint lax monoidal).

- Not only is the assignment $n \mapsto \mathsf{Sub}(\mathbb{Z}^n)$ lax monoidal,...
- ...each of its laxators has both a left adjoint and a right adjoint.

$\mathrm{Sub}(\mathbb{Z}^-)$ is bi-ajax and preserves involutions

$\mathrm{Sub}(\mathbb{Z}^-)\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ is in fact *bi-ajax* (bi-adjoint lax monoidal).

- Not only is the assignment $n \mapsto \mathrm{Sub}(\mathbb{Z}^n)$ lax monoidal,...
- ...each of its laxators has both a left adjoint and a right adjoint.
  - Consider the functor $\times\colon \mathrm{Sub}(\mathbb{Z}^m) \times \mathrm{Sub}(\mathbb{Z}^n) \to \mathrm{Sub}(\mathbb{Z}^{m+n})$
  - It has a right adjoint: intersect $R \subseteq \mathbb{Z}^{m+n}$ with $\mathbb{Z}^m$ and $\mathbb{Z}^n$.

$\mathsf{Sub}(\mathbb{Z}^-)$ is bi-ajax and preserves involutions

$\mathsf{Sub}(\mathbb{Z}^-)\colon \mathbb{A} \to \mathbb{P}\mathsf{oset}$ is in fact *bi-ajax* (bi-adjoint lax monoidal).

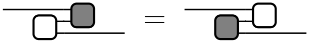
- Not only is the assignment $n \mapsto \mathsf{Sub}(\mathbb{Z}^n)$ lax monoidal,...
- ...each of its laxators has both a left adjoint and a right adjoint.
  - Consider the functor $\times\colon \mathsf{Sub}(\mathbb{Z}^m) \times \mathsf{Sub}(\mathbb{Z}^n) \to \mathsf{Sub}(\mathbb{Z}^{m+n})$
  - It has a right adjoint: intersect $R \subseteq \mathbb{Z}^{m+n}$ with $\mathbb{Z}^m$ and $\mathbb{Z}^n$.
  - It has a left adjoint: project $R \subseteq \mathbb{Z}^{m+n}$ onto $\mathbb{Z}^m$ and $\mathbb{Z}^n$.

$\mathrm{Sub}(\mathbb{Z}^-)$ is bi-ajax and preserves involutions

$\mathrm{Sub}(\mathbb{Z}^-)\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ is in fact *bi-ajax* (bi-adjoint lax monoidal).

- Not only is the assignment $n \mapsto \mathrm{Sub}(\mathbb{Z}^n)$ lax monoidal,...
- ...each of its laxators has both a left adjoint and a right adjoint.
    - Consider the functor $\times\colon \mathrm{Sub}(\mathbb{Z}^m) \times \mathrm{Sub}(\mathbb{Z}^n) \to \mathrm{Sub}(\mathbb{Z}^{m+n})$
    - It has a right adjoint: intersect $R \subseteq \mathbb{Z}^{m+n}$ with $\mathbb{Z}^m$ and $\mathbb{Z}^n$.
    - It has a left adjoint: project $R \subseteq \mathbb{Z}^{m+n}$ onto $\mathbb{Z}^m$ and $\mathbb{Z}^n$.

Further, $\mathrm{Sub}(\mathbb{Z}^-)$ preserves involutions.

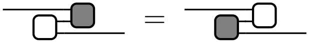
- $\mathbb{A}$ has a "negation involution"  .

$\mathrm{Sub}(\mathbb{Z}^-)$ is bi-ajax and preserves involutions

$\mathrm{Sub}(\mathbb{Z}^-)\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ is in fact *bi-ajax* (bi-adjoint lax monoidal).

- Not only is the assignment $n \mapsto \mathrm{Sub}(\mathbb{Z}^n)$ lax monoidal,...
- ...each of its laxators has both a left adjoint and a right adjoint.
  - Consider the functor $\times\colon \mathrm{Sub}(\mathbb{Z}^m) \times \mathrm{Sub}(\mathbb{Z}^n) \to \mathrm{Sub}(\mathbb{Z}^{m+n})$
  - It has a right adjoint: intersect $R \subseteq \mathbb{Z}^{m+n}$ with $\mathbb{Z}^m$ and $\mathbb{Z}^n$.
  - It has a left adjoint: project $R \subseteq \mathbb{Z}^{m+n}$ onto $\mathbb{Z}^m$ and $\mathbb{Z}^n$.

Further, $\mathrm{Sub}(\mathbb{Z}^-)$ preserves involutions.

- $\mathbb{A}$ has a "negation involution" 
- $\mathrm{Sub}(\mathbb{Z}^-)$ applied to the negation involution sends $R \mapsto \{x \mid -x \in R\}$.
- Subspaces of $\mathbb{Z}^n$ are closed under negation, so this is identity.
- $\mathrm{Sub}(\mathbb{Z}^-)$ applied to negation involution in $\mathbb{A}$ is identity in $\mathbb{P}\mathrm{oset}$.

The syntactic category of $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Such P's have *syntactic categories*, and these are abelian.

The syntactic category of $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$

Such P's have *syntactic categories*, and these are abelian.

- Let $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$ be a bi-ajax functor that preserves involutions.
- Define a category $\text{Syn}(P)$ as follows:

The syntactic category of $P \colon \mathbb{A} \to \mathbb{P}\text{oset}$

Such P's have *syntactic categories*, and these are abelian.

- Let $P \colon \mathbb{A} \to \mathbb{P}\text{oset}$ be a bi-ajax functor that preserves involutions.
- Define a category $\text{Syn}(P)$ as follows:
  - $\text{Ob}(\text{Syn}P) \coloneqq \{(m, Q, S) \mid m \in \mathbb{N}, Q \leq S \in P(m)\}$.
  - Think "formal subquotients." We'll see this acts like S/Q.

The syntactic category of $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Such P's have *syntactic categories*, and these are abelian.

- Let $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ be a bi-ajax functor that preserves involutions.
- Define a category $\mathrm{Syn}(P)$ as follows:
  - $\mathrm{Ob}(\mathrm{Syn}P) \coloneqq \{(m, Q, S) \mid m \in \mathbb{N}, Q \leq S \in P(m)\}$.
  - Think "formal subquotients." We'll see this acts like S/Q.

$\mathrm{id}_{m,Q,S} \coloneqq$ 

Anonymize out Q's, select only S's.

The syntactic category of $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Such P's have *syntactic categories*, and these are abelian.

- Let $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ be a bi-ajax functor that preserves involutions.
- Define a category $\mathrm{Syn}(P)$ as follows:
  - $\mathrm{Ob}(\mathrm{Syn}P) \coloneqq \{(m, Q, S) \mid m \in \mathbb{N}, Q \leq S \in P(m)\}$.
  - Think "formal subquotients." We'll see this acts like S/Q.

$\mathrm{id}_{m,Q,S} \coloneqq$ 

    Anonymize out Q's, select only S's.

  - $(\mathrm{Syn}P)((m, Q, S), (m', Q', S')) \coloneqq$

The syntactic category of $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Such P's have *syntactic categories*, and these are abelian.

- Let $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ be a bi-ajax functor that preserves involutions.
- Define a category $\mathrm{Syn}(P)$ as follows:
  - $\mathrm{Ob}(\mathrm{Syn}P) \coloneqq \{(m, Q, S) \mid m \in \mathbb{N}, Q \leq S \in P(m)\}$.
  - Think "formal subquotients." We'll see this acts like S/Q.

    $\mathrm{id}_{m,Q,S} \coloneqq$ 

    Anonymize out Q's, select only S's.
  - $(\mathrm{Syn}P)((m, Q, S), (m', Q', S')) \coloneqq$

    $\{L \in P(m{+}m') \mid Q \boxplus Q' \leq L \leq S \boxplus S'$ and L is an internal left adjoint$\}$

The syntactic category of $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

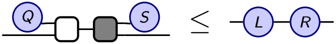
Such P's have *syntactic categories*, and these are abelian.

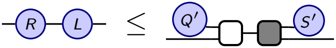
- Let $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ be a bi-ajax functor that preserves involutions.
- Define a category $\mathrm{Syn}(P)$ as follows:
  - $\mathrm{Ob}(\mathrm{Syn}P) \coloneqq \{(m, Q, S) \mid m \in \mathbb{N}, Q \leq S \in P(m)\}$.
  - Think "formal subquotients." We'll see this acts like S/Q.

    $\mathrm{id}_{m,Q,S} \coloneqq$ 

    Anonymize out Q's, select only S's.
  - $(\mathrm{Syn}P)((m, Q, S), (m', Q', S')) \coloneqq$

    $\{L \in P(m+m') \mid Q \boxplus Q' \leq L \leq S \boxplus S' \text{ and } L \text{ is an internal left adjoint}\}$





  - Think "group homomorphism $S/Q \to S'/Q'$."

The syntactic category of $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$

Such P's have *syntactic categories*, and these are abelian.

- Let $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ be a bi-ajax functor that preserves involutions.
- Define a category $\mathrm{Syn}(P)$ as follows:
  - $\mathrm{Ob}(\mathrm{Syn}P) \coloneqq \{(m, Q, S) \mid m \in \mathbb{N}, Q \leq S \in P(m)\}$.
  - Think "formal subquotients." We'll see this acts like S/Q.

$\mathrm{id}_{m,Q,S} \coloneqq$ 

  Anonymize out Q's, select only S's.

  - $(\mathrm{Syn}P)((m, Q, S), (m', Q', S')) \coloneqq$

  $\{L \in P(m+m') \mid Q \boxplus Q' \leq L \leq S \boxplus S' \text{ and } L \text{ is an internal left adjoint}\}$

  - Think "group homomorphism $S/Q \to S'/Q'$."

One can prove that the result $\mathrm{Syn}(P)$ is a category and that it's abelian.

Aside: a sequence being a complex is its homology

Suppose given a sequence $A \xrightarrow{f} B \xrightarrow{g} C$ of abelian group homomorphisms.

- If $\mathrm{im}(f) \subseteq \ker(g)$, as subobjects of B, we say it's a complex.

Aside: a sequence being a complex is its homology

Suppose given a sequence $A \xrightarrow{f} B \xrightarrow{g} C$ of abelian group homomorphisms.

- If $\mathrm{im}(f) \subseteq \ker(g)$, as subobjects of B, we say it's a complex.
- But in the syntactic category such relationships $Q \subseteq S$ define objects.
- Namely, $\mathrm{im}(f) \subseteq \ker(g)$ *just is* the quotient object $\ker(g)/\,\mathrm{im}(f)$.
- This is the homology of the complex there.

Aside: a sequence being a complex is its homology

Suppose given a sequence $A \xrightarrow{f} B \xrightarrow{g} C$ of abelian group homomorphisms.

- If $\mathrm{im}(f) \subseteq \ker(g)$, as subobjects of B, we say it's a complex.
- But in the syntactic category such relationships $Q \subseteq S$ define objects.
- Namely, $\mathrm{im}(f) \subseteq \ker(g)$ *just is* the quotient object $\ker(g)/\,\mathrm{im}(f)$.
- This is the homology of the complex there.

The homology at B *is* the assertion that the sequence is a complex at B.

Outline

The notion of supply

"Every object is compatibly equipped with algebraic structure from $\mathbb{P}$."

The notion of supply

"Every object is compatibly equipped with algebraic structure from $\mathbb{P}$."

Definition

Let $\mathcal{C}$ be a symmetric monoidal category (SMC) and $\mathbb{P}$ a prop.

The notion of supply

"Every object is compatibly equipped with algebraic structure from $\mathbb{P}$."

Definition

Let $\mathcal{C}$ be a symmetric monoidal category (SMC) and $\mathbb{P}$ a prop. A *supply of* $\mathbb{P}$ *in* $\mathcal{C}$ consists of a strict monoidal functor $s_c \colon \mathbb{P} \to \mathcal{C}$ with $s(1) = c$ for every object $c \in \mathcal{C}$, such that the following diagrams commute:

$$
\begin{array}{ccc}
c^{\otimes m} \otimes d^{\otimes m} & \xrightarrow{s_c(\mu) \otimes s_d(\mu)} & c^{\otimes n} \otimes d^{\otimes n} \\
{\scriptstyle \cong}\downarrow & & \downarrow{\scriptstyle \cong} \\
(c \otimes d)^{\otimes m} & \xrightarrow[s_{c \otimes d}(\mu)]{} & (c \otimes d)^{\otimes n}
\end{array}
\qquad
\begin{array}{ccc}
I & = & I \\
{\scriptstyle \cong}\downarrow & & \downarrow{\scriptstyle \cong} \\
I^{\otimes m} & \xrightarrow[s_I(\mu)]{} & I^{\otimes n}
\end{array}
$$

The notion of supply

"Every object is compatibly equipped with algebraic structure from $\mathbb{P}$."

Definition

Let $\mathcal{C}$ be a symmetric monoidal category (SMC) and $\mathbb{P}$ a prop. A *supply of* $\mathbb{P}$ *in* $\mathcal{C}$ consists of a strict monoidal functor $s_c \colon \mathbb{P} \to \mathcal{C}$ with $s(1) = c$ for every object $c \in \mathcal{C}$, such that the following diagrams commute:

$$\begin{array}{ccc} c^{\otimes m} \otimes d^{\otimes m} & \xrightarrow{s_c(\mu) \otimes s_d(\mu)} & c^{\otimes n} \otimes d^{\otimes n} \\ \cong\downarrow & & \downarrow\cong \\ (c \otimes d)^{\otimes m} & \xrightarrow[s_{c \otimes d}(\mu)]{} & (c \otimes d)^{\otimes n} \end{array} \qquad \begin{array}{ccc} I & = & I \\ \cong\downarrow & & \downarrow\cong \\ I^{\otimes m} & \xrightarrow[s_I(\mu)]{} & I^{\otimes n} \end{array}$$

Same definition for SM po-categories and po-props.

The notion of supply

"Every object is compatibly equipped with algebraic structure from $\mathbb{P}$."

Definition

Let $\mathcal{C}$ be a symmetric monoidal category (SMC) and $\mathbb{P}$ a prop. A *supply of* $\mathbb{P}$ *in* $\mathcal{C}$ consists of a strict monoidal functor $s_c \colon \mathbb{P} \to \mathcal{C}$ with $s(1) = c$ for every object $c \in \mathcal{C}$, such that the following diagrams commute:

$$\begin{array}{ccc} c^{\otimes m} \otimes d^{\otimes m} & \xrightarrow{s_c(\mu) \otimes s_d(\mu)} & c^{\otimes n} \otimes d^{\otimes n} \\ \cong\downarrow & & \downarrow\cong \\ (c \otimes d)^{\otimes m} & \xrightarrow[s_{c \otimes d}(\mu)]{} & (c \otimes d)^{\otimes n} \end{array} \qquad \begin{array}{ccc} I & = & I \\ \cong\downarrow & & \downarrow\cong \\ I^{\otimes m} & \xrightarrow[s_I(\mu)]{} & I^{\otimes n} \end{array}$$

Same definition for SM po-categories and po-props.

Examples:

- If $\mathcal{C}$ has finite products, it has a supply of comonoids.

The notion of supply

"Every object is compatibly equipped with algebraic structure from $\mathbb{P}$."

Definition

Let $\mathcal{C}$ be a symmetric monoidal category (SMC) and $\mathbb{P}$ a prop. A *supply of* $\mathbb{P}$ *in* $\mathcal{C}$ consists of a strict monoidal functor $s_c \colon \mathbb{P} \to \mathcal{C}$ with $s(1) = c$ for every object $c \in \mathcal{C}$, such that the following diagrams commute:

$$\begin{array}{ccc} c^{\otimes m} \otimes d^{\otimes m} & \xrightarrow{s_c(\mu) \otimes s_d(\mu)} & c^{\otimes n} \otimes d^{\otimes n} \\ \cong\downarrow & & \downarrow\cong \\ (c \otimes d)^{\otimes m} & \xrightarrow[s_{c \otimes d}(\mu)]{} & (c \otimes d)^{\otimes n} \end{array} \qquad \begin{array}{ccc} I & = & I \\ \cong\downarrow & & \downarrow\cong \\ I^{\otimes m} & \xrightarrow[s_I(\mu)]{} & I^{\otimes n} \end{array}$$

Same definition for SM po-categories and po-props.

Examples:

- If $\mathcal{C}$ has finite products, it has a supply of comonoids.
- There is always a canonical supply of $\mathbb{P}$ in $\mathbb{P}$.

The notion of supply

"Every object is compatibly equipped with algebraic structure from $\mathbb{P}$."

Definition

Let $\mathcal{C}$ be a symmetric monoidal category (SMC) and $\mathbb{P}$ a prop. A *supply of* $\mathbb{P}$ *in* $\mathcal{C}$ consists of a strict monoidal functor $s_c \colon \mathbb{P} \to \mathcal{C}$ with $s(1) = c$ for every object $c \in \mathcal{C}$, such that the following diagrams commute:

$$\begin{array}{ccc} c^{\otimes m} \otimes d^{\otimes m} & \xrightarrow{s_c(\mu) \otimes s_d(\mu)} & c^{\otimes n} \otimes d^{\otimes n} \\ \cong\downarrow & & \downarrow\cong \\ (c \otimes d)^{\otimes m} & \xrightarrow[s_{c \otimes d}(\mu)]{} & (c \otimes d)^{\otimes n} \end{array} \qquad \begin{array}{ccc} I & = & I \\ \cong\downarrow & & \downarrow\cong \\ I^{\otimes m} & \xrightarrow[s_I(\mu)]{} & I^{\otimes n} \end{array}$$

Same definition for SM po-categories and po-props.

Examples:

- If $\mathcal{C}$ has finite products, it has a supply of comonoids.
- There is always a canonical supply of $\mathbb{P}$ in $\mathbb{P}$.
- If $\mathcal{A}$ is abelian, its relations po-cat $\mathbb{R}\mathrm{el}_{\mathcal{A}}$ supplies abelian relations $\mathbb{A}$.

Abelian calculi

Definition

An *abelian calculus* is a pair $(\mathcal{C}, P)$, where $\mathcal{C}$ supplies abelian relations and $P\colon \mathcal{C} \to \mathbb{P}\mathrm{oset}$ is bi-ajax and preserves involutions.

Abelian calculi

Definition

An *abelian calculus* is a pair $(\mathcal{C}, P)$, where $\mathcal{C}$ supplies abelian relations and $P\colon \mathcal{C} \to \mathbb{P}\mathrm{oset}$ is bi-ajax and preserves involutions.

Theorem (Fong-S.)

Abelian categories are reflective in the 2-category of abelian calculi.

$$\mathbb{A}\mathrm{bCalc} \underset{\mathrm{Prd}}{\overset{\mathrm{Syn}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\Rightarrow}}}} \mathbb{A}\mathrm{bCat}$$

In particular for $\mathcal{A} \in \mathbb{A}\mathrm{bCat}$, *the unit* $\mathcal{A} \xrightarrow{\cong} \mathrm{SynPrd}(\mathcal{A})$ *is an equivalence.*

The predicates functor

The inclusion "predicates" functor $\mathsf{Prd}\colon \mathbb{A}\mathsf{bCat} \to \mathbb{A}\mathsf{bCalc}$ is given by

$$\mathsf{Prd}(\mathcal{A}) \coloneqq (\mathbb{R}\mathsf{el}_{\mathcal{A}}, \mathbb{R}\mathsf{el}_{\mathcal{A}}(0, -)).$$

The predicates functor

The inclusion “predicates” functor $\mathsf{Prd}\colon \mathbb{A}\mathsf{bCat} \to \mathbb{A}\mathsf{bCalc}$ is given by

$$\mathsf{Prd}(\mathcal{A}) \coloneqq (\mathbb{R}\mathsf{el}_{\mathcal{A}}, \mathbb{R}\mathsf{el}_{\mathcal{A}}(0, -)).$$

- Since $\mathcal{A}$ is abelian, $\mathbb{R}\mathsf{el}_{\mathcal{A}}$ supplies abelian relations.
- $\mathbb{R}\mathsf{el}_{\mathcal{A}}(0, -))\colon \mathbb{R}\mathsf{el}_{\mathcal{A}} \to \mathbb{P}\mathsf{oset}$ sends $A \mapsto \mathbb{R}\mathsf{el}_{\mathcal{A}}(0, A) = \mathsf{Sub}(A)$.
  - Easy to see that $\mathbb{R}\mathsf{el}_{\mathcal{A}}(0, -)$ is a po-functor; it's represented by 0.
  - It is also bi-ajax and preserves involutions.

The predicates functor

The inclusion "predicates" functor $\mathrm{Prd}\colon \mathbb{A}\mathrm{bCat} \to \mathbb{A}\mathrm{bCalc}$ is given by

$$\mathrm{Prd}(\mathcal{A}) \coloneqq (\mathbb{R}\mathrm{el}_{\mathcal{A}}, \mathbb{R}\mathrm{el}_{\mathcal{A}}(0,-)).$$

- Since $\mathcal{A}$ is abelian, $\mathbb{R}\mathrm{el}_{\mathcal{A}}$ supplies abelian relations.
- $\mathbb{R}\mathrm{el}_{\mathcal{A}}(0,-))\colon \mathbb{R}\mathrm{el}_{\mathcal{A}} \to \mathbb{P}\mathrm{oset}$ sends $A \mapsto \mathbb{R}\mathrm{el}_{\mathcal{A}}(0,A) = \mathrm{Sub}(A)$.
    - Easy to see that $\mathbb{R}\mathrm{el}_{\mathcal{A}}(0,-)$ is a po-functor; it's represented by 0.
    - It is also bi-ajax and preserves involutions.

$\mathrm{Prd}\colon \mathbb{A}\mathrm{bCat} \to \mathbb{A}\mathrm{bCalc}$ is fully faithful and locally fully faithful.

Outline

Summary

Abelian calculi give a "sketch" approach to abelian categories.

- The 2-cat of abelian categories embeds into that of abelian calculi.
- This embedding is full, and it has a reflector $\mathrm{Syn}\colon \mathbb{A}\mathrm{bCalc} \to \mathbb{A}\mathrm{bCat}$.

An abelian calculus, e.g. $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$, is graphical.

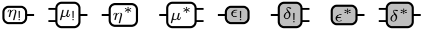
- Have access to icons: $\eta_!$ $\mu_!$ η^* μ^* $\epsilon_!$ $\delta_!$ ϵ^* δ^*

Summary

Abelian calculi give a “sketch” approach to abelian categories.

- The 2-cat of abelian categories embeds into that of abelian calculi.
- This embedding is full, and it has a reflector Syn: $\mathbb{A}$bCalc $\to$ $\mathbb{A}$bCat.

An abelian calculus, e.g. $P\colon \mathbb{A} \to \mathbb{P}\text{oset}$, is graphical.

- Have access to icons: 

- These icons and their relations are used in constructing Syn(P)...
    - ...namely to define identity, composition, kernels, and cokernels.

Summary

Abelian calculi give a "sketch" approach to abelian categories.

- The 2-cat of abelian categories embeds into that of abelian calculi.
- This embedding is full, and it has a reflector $\mathrm{Syn}\colon \mathbb{A}\mathrm{bCalc} \to \mathbb{A}\mathrm{bCat}$.

An abelian calculus, e.g. $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$, is graphical.

- Have access to icons: $\eta_!$ $\mu_!$ η^* μ^* $\epsilon_!$ $\delta_!$ ϵ^* δ^*
- These icons and their relations are used in constructing $\mathrm{Syn}(P)$...
  - ...namely to define identity, composition, kernels, and cokernels.

Different knobs to turn.

- Knobs in $\mathbb{A}\mathrm{bCat}$: four axioms (coprods, prods, kernels, cokernels).

Summary

Abelian calculi give a "sketch" approach to abelian categories.

- The 2-cat of abelian categories embeds into that of abelian calculi.
- This embedding is full, and it has a reflector Syn: $\mathbb{A}$bCalc $\to$ $\mathbb{A}$bCat.

An abelian calculus, e.g. $P\colon \mathbb{A} \to \mathbb{P}$oset, is graphical.

- Have access to icons: $\eta_!$ $\mu_!$ η^* μ^* $\epsilon_!$ $\delta_!$ ϵ^* δ^*
- These icons and their relations are used in constructing Syn(P)...
  - ...namely to define identity, composition, kernels, and cokernels.

Different knobs to turn.

- Knobs in $\mathbb{A}$bCat: four axioms (coprods, prods, kernels, cokernels).
- Knobs in $\mathbb{A}$bCalc: "bi-ajax functor to $\mathbb{P}$oset, preserving involutions".

Summary

Abelian calculi give a “sketch” approach to abelian categories.

- The 2-cat of abelian categories embeds into that of abelian calculi.
- This embedding is full, and it has a reflector $\mathrm{Syn}\colon \mathbb{A}\mathrm{bCalc} \to \mathbb{A}\mathrm{bCat}$.

An abelian calculus, e.g. $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$, is graphical.

- Have access to icons: $\eta_!$ $\mu_!$ η^* μ^* $\epsilon_!$ $\delta_!$ ϵ^* δ^*
- These icons and their relations are used in constructing $\mathrm{Syn}(P)$...
  - ...namely to define identity, composition, kernels, and cokernels.

Different knobs to turn.

- Knobs in $\mathbb{A}\mathrm{bCat}$: four axioms (coprods, prods, kernels, cokernels).
- Knobs in $\mathbb{A}\mathrm{bCalc}$: “bi-ajax functor to $\mathbb{P}\mathrm{oset}$, preserving involutions”.

Thanks! Questions and comments welcome!