Graphical abelian logic

David I. Spivak* and Brendan Fong

July 19, 2019

Graphical languages in category theory

String diagrams for monoidal categories were invented by Joyal and Street.

Graphical languages in category theory

String diagrams for monoidal categories were invented by Joyal and Street.

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

Graphical languages in category theory

String diagrams for monoidal categories were invented by Joyal and Street.

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

Can also be defined combinatorially using lax monoidal functors.

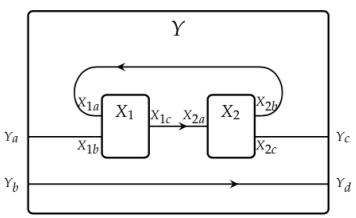
Graphical languages in category theory

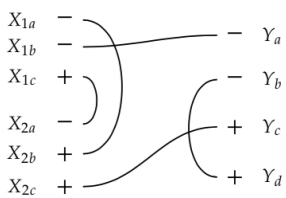
String diagrams for monoidal categories were invented by Joyal and Street.

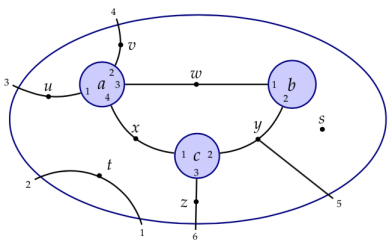
- See Selinger's "A survey of graphical languages for monoidal cats"
- Defined in terms of topological spaces and homotopies.

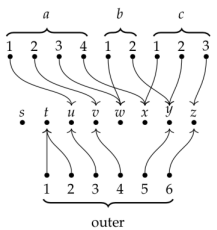
Can also be defined combinatorially using lax monoidal functors.

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









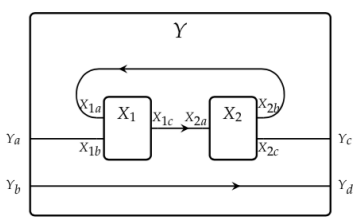
Graphical languages in category theory

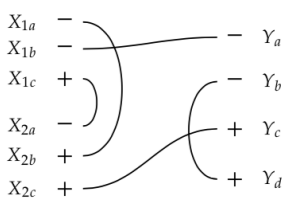
String diagrams for monoidal categories were invented by Joyal and Street.

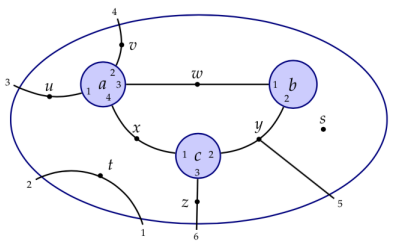
- See Selinger's "A survey of graphical languages for monoidal cats"
- Defined in terms of topological spaces and homotopies.

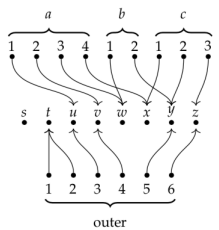
Can also be defined combinatorially using lax monoidal functors.

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









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

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.

Just think fgAb or $\mathrm{Vect}_{\mathbb{R}}$.

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* ($\bot$) "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* ($\bot$) "zero"
- Every morphism $f\colon A \to B$ in $\mathcal{A}$ has an image $A \twoheadrightarrow \mathrm{im}(f) \rightarrowtail B$.

Objects also have quotients. How to think of quotient A/Q for $Q \subseteq A$.

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 $A \twoheadrightarrow \mathrm{im}(f) \rightarrowtail B$.

Objects also have quotients. How to think of quotient A/Q for $Q \subseteq A$.

- The elements of A/Q are those of A, except with Q-noise added.
- Anonymize / wash out the Q dimension.

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: $\mathbb{P}(m, n) \in \mathrm{Poset}$.

Introducing the po-prop of abelian relations

We will be discussing a graphical syntax for abelian categories.

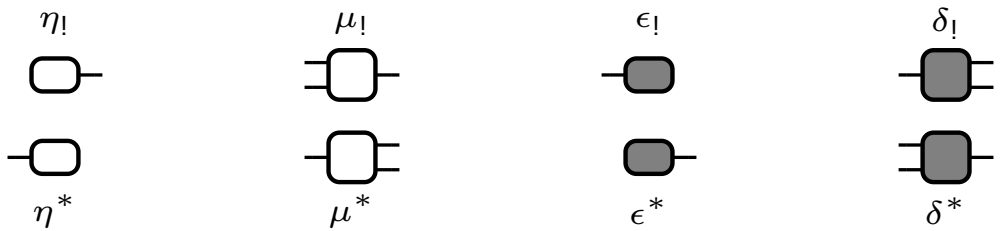
- The graphical syntax is governed by a certain kind of *monoidal theory*.
- A *prop* is a strict monoidal category whose object monoid is $(\mathbb{N}, 0, +)$.
- A *po-prop* $\mathbb{P}$ is a locally-posetal version: $\mathbb{P}(m, n) \in \mathrm{Poset}$.
- Think of the maps in $\mathbb{P}$ as icons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

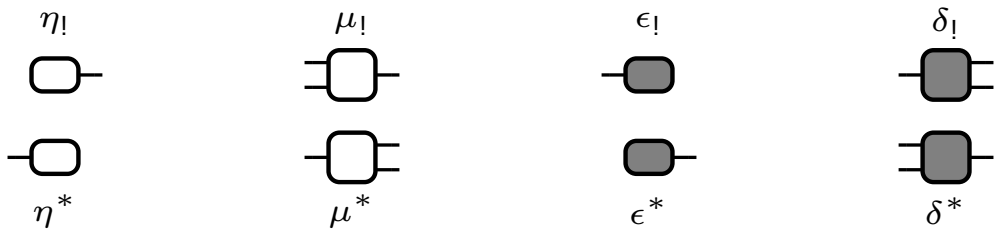


Introducing the po-prop of abelian relations

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: $\mathbb{P}(m, n) \in \mathrm{Poset}$.
- 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:



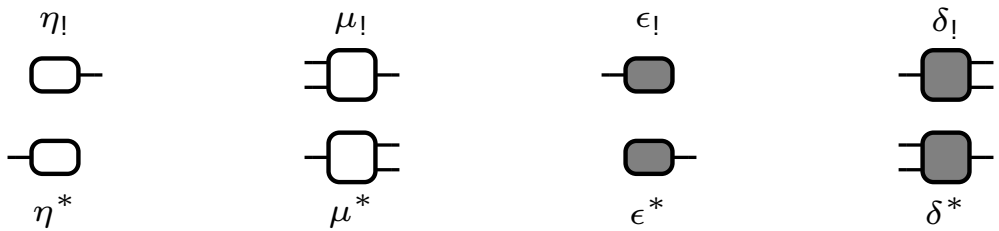
Intuition: icons $m \to n$ are maps between subsp's of $\mathbb{R}^m$ and subsp's of $\mathbb{R}^n$

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: $\mathbb{P}(m, n) \in \mathrm{Poset}$.
- 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 $\mathbb{R}^n$

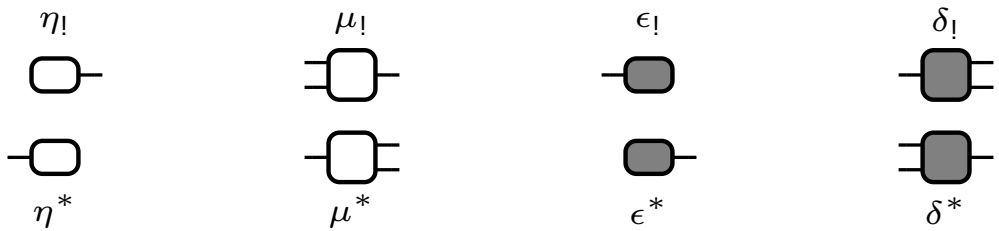
- All come in adjoint pairs;
- η is "0", μ is "+", ϵ is "everything", δ is "equality".

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: $\mathbb{P}(m, n) \in \mathrm{Poset}$.
- 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".

Some terms in the graphical language

Given a map $f : X \to Y$, 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 $f : X \to Y$, 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 $f : X \to Y$, 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...

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

Some terms in the graphical language

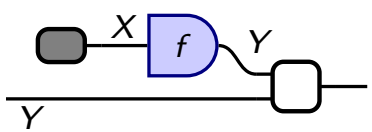
Given a map $f : X \to Y$, its cokernel and kernel are canonical maps:

- 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."

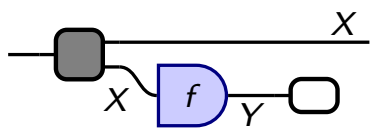
In pictures...

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

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



$\ker(f) \to X$

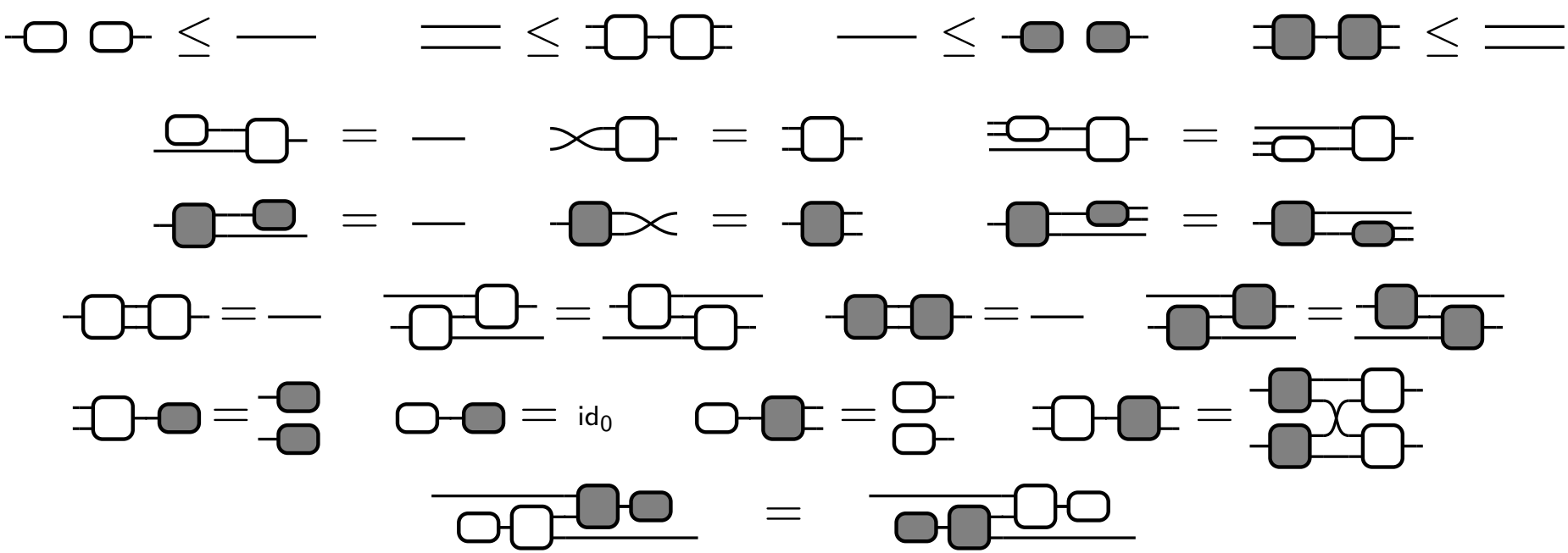


The po-prop of abelian relations

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:



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

Main interest: lax monoidal po-functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$.

- String diagrams from $\mathbb{A}$ govern abelian categories.
- I.e. functors $P\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ correspond to abelian cats.

Example: Finitely-generated abelian groups

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

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

Example: Finitely-generated abelian groups

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

- For each $n \in \mathrm{Ob}(\mathbb{A})$, 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.

Example: Finitely-generated abelian groups

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

- For each $n \in \mathrm{Ob}(\mathbb{A})$, 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 $R \mapsto \{x + y \mid (x, y) \in R\}$.

Example: Finitely-generated abelian groups

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

- For each $n \in \mathrm{Ob}(\mathbb{A})$, 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 $\mathrm{Sub}(\mathbb{Z}^1) \to 1$.

Example: Finitely-generated abelian groups

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

- For each $n \in \mathrm{Ob}(\mathbb{A})$, 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 $\mathrm{Sub}(\mathbb{Z}^1) \to 1$.
- Define $P(\mu^*)\colon \mathrm{Sub}(\mathbb{Z}) \to \mathrm{Sub}(\mathbb{Z}^2)$ by $R \mapsto \{(x, y) \mid x + y \in R\}$.
- Define $P(\delta_!)\colon \mathrm{Sub}(\mathbb{Z}) \to \mathrm{Sub}(\mathbb{Z}^2)$ by $R \mapsto \{(x, x) \mid x \in R\}$.
- Define $P(\epsilon^*)\colon 1 \to \mathrm{Sub}(\mathbb{Z})$ by $1 \mapsto \top$.
- Define $P(\delta^*)\colon \mathrm{Sub}(\mathbb{Z}^2) \to \mathrm{Sub}(\mathbb{Z})$ 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 $P(n) \coloneqq \mathrm{Sub}(\mathbb{Z}^n)$ 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 $1 \to \mathrm{Sub}(\mathbb{Z}^0)$.

$\mathrm{Sub}(\mathbb{Z}^-)\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ 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 (preserves equations and ineq's).

$\mathrm{Sub}(\mathbb{Z}^-)$ 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.

$\mathrm{Sub}(\mathbb{Z}^-)$ 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 $\mathbb{Z}^n$.

$\mathrm{Sub}(\mathbb{Z}^-)$ 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).

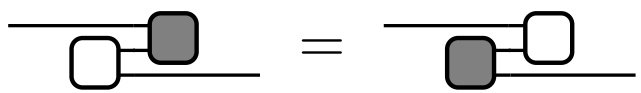
- Not only is the assignment $n \mapsto \mathrm{Sub}(\mathbb{Z}^n)$ lax monoidal,...
- ...each of its laxators has both a left adjoint and a right adjoint.
  - Consider the functor $\times\colon \mathrm{Sub}(\mathbb{Z}^m) \times \mathrm{Sub}(\mathbb{Z}^n) \to \mathrm{Sub}(\mathbb{Z}^{m+n})$
  - It has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

$\mathrm{Sub}(\mathbb{Z}^-)$ 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.

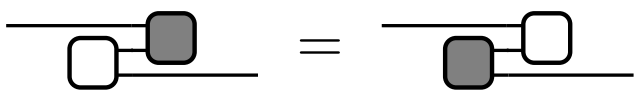
- $\mathbb{A}$ has a "negation involution"  .

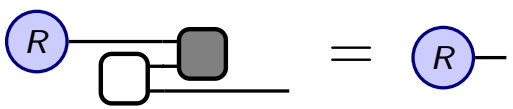
$\mathrm{Sub}(\mathbb{Z}^-)$ is bi-ajax and preserves involutions

$\mathrm{Sub}(\mathbb{Z}^-)\colon \mathbb{A} \to \mathbb{P}\mathrm{oset}$ 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 $\mathbb{P}\mathrm{oset}$.



Main theorem

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.

Main theorem

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.

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

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

Main theorem

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.

$$\mathbb{A}\mathrm{bCalc} \underset{\mathsf{Prd}}{\overset{\mathsf{Syn}}{\rightleftarrows}} \mathbb{A}\mathrm{bCat}$$

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

Rather than four axioms, bi-ajax functors to poset: Different knobs to turn.

Main theorem

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.

$$\mathbb{A}\mathrm{bCalc} \underset{\mathsf{Prd}}{\overset{\mathsf{Syn}}{\rightleftarrows}} \mathbb{A}\mathrm{bCat}$$

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

Rather than four axioms, bi-ajax functors to poset: Different knobs to turn.

Thanks! Questions and comments welcome!