ABELIAN CALCULI PRESENT ABELIAN CATEGORIES

DAVID I. SPIVAK* AND BRENDAN FONG

Abelian categories are "good places to do computation," e.g. homological algebra. The category of (finitely generated) abelian groups, of vector spaces, and of chain complexes are examples. The definition of abelian category is:

(*) a category with a zero object, finite products and coproducts, a kernel and cokernel for every morphism, and with the property that each monic is a kernel and each epic is a cokernel.

From these simple axioms follow many interesting consequences: each abelian category A has all finite limits and colimits, its finite products and coproducts coincide, its opposite is abelian, and—most interesting for this talk—it is a regular category.

The fact that each abelian category A is regular implies that it has a "nice" theory of relations, a monoidal 2-category $\mathbb{A}$, whose monoidal structure is inherited from A and from which A can be recovered as the category of left adjoints. The relations in an abelian category can be considered as formulas for a kind of regular logic called *abelian logic*, where roughly speaking, new formulas can be built from old using existential quantification ($\exists$), meet ($\wedge$), true, equals ($=$), join ($\vee$), zero (0), and sum ($+$). For example, if $R(y,z)$ and $S(z,z')$ have type $Y \times Z$ and $Z \times Z$ respectively, then $\exists z.\, (R(y,z) \wedge S(z,z') \wedge z + z = z') \vee (y = 0)$ has type $Y \times Z$.

In this talk we will discuss a new presentation language—a syntax we call abelian calculus—for abelian categories, that looks nothing like (*). Let Mat denote the category of matrices with integer coefficients (the full subcategory of finitely-generated abelian groups spanned by the free ones); it is a regular category. Let $\mathbb{M}\mathsf{at}$ denote the monoidal 2-category of relations in Mat, and let $\mathbb{P}\mathsf{oset}$ denote the monoidal 2-category of posets under Cartesian product, monotone maps, and natural transformations. An *abelian calculus with one type* is a lax monoidal 2-functor

$$C\colon \mathbb{M}\mathsf{at} \to \mathbb{P}\mathsf{oset}$$

such that each lax coherence map has both a left and a right adjoint. We abbreviate the condition "C is a bi-adjoint lax monoidal 2-functor" as "C is *bi-ajax*." An abelian calculus is a generalization (T, C) of the above, where T is a set, $\mathsf{Mat}_T := \coprod_T \mathsf{Mat}$ is the T-indexed coproduct prop, and $C\colon \mathbb{M}\mathsf{at}_T \to \mathbb{P}\mathsf{oset}$ is bi-ajax.

From any abelian category A, one obtains an abelian calculus $\mathrm{Rel}(A)$ as the relations in A, i.e. take $T := \mathrm{Ob}(A)$ and $C(t_1, \ldots, t_n) := \mathrm{Sub}(t_1 \times \cdots \times t_n)$. Each such poset is in fact a lattice, and its meet and join are roughly where the bi-ajax condition arises. Conversely, given an abelian calculus (T, C), one can form its *syntactic category* $\mathrm{Syn}(T, C)$ and prove that it is abelian. It is in this sense that abelian calculi form a presentation language for abelian categories: there is an adjunction

$$\mathrm{Syn} : \mathsf{AbCalc} \leftrightarrows \mathsf{AbCat} : \mathrm{Rel}$$

which is an essential reflection in the sense that, for every abelian category A, the functor $\mathrm{Syn}(\mathrm{Rel}(A)) \to A$ is an equivalence.

It may at first be surprising that there is a connection between abelian categories and functors from integer matrices to posets. But as functors $C\colon \mathbb{M}\mathsf{at} \to \mathbb{P}\mathsf{oset}$, abelian calculi offer completely different sorts of "knobs to turn" for generalization, e.g. replace $\mathbb{M}\mathsf{at}$ with matrices of natural numbers or replace $\mathbb{P}\mathsf{oset}$ with $\mathbb{C}\mathsf{at}$. Another interesting feature is that $\mathbb{M}\mathsf{at}$ has a rich graphical language developed by Zanasi, Sobociński, and others, for which C provides a semantic interpretation. The formal terms $\exists z.\, R(y,z) \vee (y=0)$ in the logical language discussed above can be reinterpreted as graphical terms in a wiring diagram language.

We will discuss the above ideas and briefly consider homology in this context, e.g. showing how the "connecting homomorphism" from the snake lemma looks as a graphical term.