Brief Summary of Research Activities

William DeMeo

1. Introduction

The last two decades have witnessed explosive growth in the number of applications of abstract mathematics, especially in computer science, and this trend continues on a steep upward trajectory. Mathematical fields like *universal algebra* and *category theory* have long had a substantial influence on the development of theoretical computer science, particularly in *domain theory*, *denotational semantics*, and *programming languages research* [**?**, **?**]. Dually, progress in theoretical computer science has informed and inspired a substantial amount of pure mathematics in the last half-century [**?**, **?**, **?**, **?**, **?**, **?**], just as physics and physical intuition motivated so many of the mathematical discoveries of the last two centuries.

Functional programming languages that support *dependent* and *(co)inductive types* have brought about new opportunities to apply abstract concepts from universal algebra and category theory to the practice of programming, to yield code that is more modular, reusable, and safer, and to express ideas that would be difficult or impossible to express in *imperative* or *procedural programming languages* [**?**, **?**, **?**, Chs. 5 & 10]. The *Lean Programming Language* [**?**] is one example of a functional language that supports dependent and (co)inductive types, and it is ideally suited for expressing the key concepts and result of universal algebra. As such, Lean is the primary language we have chosen for our most recently launched project, *Formal Foundations for Informal Mathematics Research*. This project is briefly mentioned in Section 2.1 below and described in detail in the document demeo_informal_foundations.pdf.

Universal algebra has also been invigorated by a recently discovered connection to complexity theory, and this connection was the primary focus of my work from 2015 to 2017. The theory of finite algebras has turned out to be broadly applicable in obtaining deep and definitive results about the complexity of algorithmic problems in the broad class of *constraint satisfaction problems* (CSPs). We call this area *algebraic CSP theory*. The tools of universal algebra, combined with combinatorial reasoning about polymorphisms (multi-variable endomorphisms) acting on finite graphs and other relational structures, has produced deep new results concerning the complexity of CSPs, and has explained and united older results. Furthermore, almost all of the new results have been turned around and used to produce startling new algebraic results of a kind never seen before in universal algebra.

There is now an extensive and growing research literature on algebraic CSP theory. Consequently, the field of universal algebra is more active now than at any other time in its brief eighty-five year history. My work in algebraic CSP is summarized in Section 2.4 below. Details can be found in the manuscript entitled "Universal Algebraic Methods for Constraint Satisfaction Problems," [**?**] that I authored with Cliff Bergman, available on the arXiv.

2. Research Projects

2.1. **Lean and Formal Foundations.** I am fascinated by the connections between programming languages and mathematics, and my most recently initiated research program aims to develop a library of all core definitions and theorems of universal algebra in the Lean proof assistant and programming language [**?**]. The title of this project is *Formal Foundations for Informal Mathematics Research* and a detailed project description is available in the document demeo_informal_foundations.pdf.

2.2. **New Characterizations of Bounded Lattices and Fiber Products.** About a year ago I began collaborating with Peter Mayr (CU Boulder) and Nik Ruskuc (University of St. Andrews) who were interested in knowing when a homomorphism $\varphi\colon \mathbf{F} \to \mathbf{L}$ from a finitely generated free lattice $\mathbf{F}$ onto a finite lattice $\mathbf{L}$ has a kernel $\ker \varphi$ that is a finitely generated sublattice of $\mathbf{F}^2$. We conjectured that this could be characterized by whether or not the homomorphism is *bounded*.[1] and I presented a proof of one direction of this conjecture at the Algebras and Lattices in Hawaii conference earlier this year. Last month we proved the converse and thus confirmed our conjecture. All along Mayr and Ruskuc have had in mind an application for the fact that our new result is equivalent to a characterization of *fiber products* of lattices. With the proof of our new characterization theorem complete, we expect to have a manuscript ready for submission by January 2019.

2.3. **Tractability of Deciding Existence of Special Terms.** Among my most recent research accomplishments was a surprising discovery that a certain decision problem about algebraic structures that previously seemed out of reach is actually tractable. We considered the following practical question: Given a finite algebra $\mathbf{A}$ in a finite language, can we efficiently decide whether the variety generated by $\mathbf{A}$ has a so called *difference term*? In a recently submitted paper we answer this question (positively) in the idempotent case and then describe algorithms for constructing difference term operations [**?**].

A *difference term* for a variety $\mathcal{V}$ is a ternary term d in the language of $\mathcal{V}$ that satisfies the following: if $\mathbf{A} = \langle A, \dots \rangle \in \mathcal{V}$, then for all $a, b \in A$ we have

$$d^{\mathbf{A}}(a,a,b) = b \quad \text{and} \quad d^{\mathbf{A}}(a,b,b)\ [\theta,\theta]\ a, \tag{1}$$

where θ is any congruence containing (a, b) and $[\cdot,\cdot]$ denotes the *commutator*. When the relations in (1) hold for a single algebra $\mathbf{A}$ and term d we call $d^{\mathbf{A}}$ a *difference term operation* for $\mathbf{A}$.

Difference terms are studied extensively in the general algebra literature. (See, for example, [**?**, **?**, **?**, **?**, **?**].) There are many reasons to study difference terms, but one obvious reason is the following: if we know that a variety has a difference term, this fact allows us to deduce that the algebras inhabiting this variety must satisfy certain interesting properties. Perhaps the most important property can be summarized in

[1] See [**?**] for the definition of a bounded lattice homomorphism.

the following heuristic slogan: *varieties with a difference term have a commutator that behaves nicely.*

The class of varieties that have a difference term is fairly broad and includes those varieties that are congruence modular or congruence meet-semidistributive. Since the commutator of two congruences of an algebra in a congruence meet-semidistributive variety is just their intersection [**?**], it follows that the term $d(x, y, z) := z$ is a difference term for such varieties. A special type of difference term $d(x, y, z)$ is one that satisfies the equations $d(x, x, y) = y$ and $d(x, y, y) = x$. Such terms are called *Maltsev terms*. So if $\mathbf{A}$ lies in a variety that has a difference term $d(x, y, z)$ and if $\mathbf{A}$ is *abelian* (i.e., $[1_A, 1_A] = 0_A$), then d will be a Maltsev term for $\mathbf{A}$.

Difference terms also play a role in recent work of Keith Kearnes, Agnes Szendrei, and Ross Willard. In [**?**] these authors give a positive answer Jónsson's famous question—whether a variety of finite residual bound must be finitely axiomatizable—for the special case in which the variety has a difference term.[2]

Computers have become invaluable as a research tool and have helped to broaden and deepen our understanding of algebraic structures and the varieties they inhabit. This is largely due to the efforts of researchers who, over the last three decades, have found ingenious ways to coax computers into solving challenging abstract algebraic decision problems, and to do so very quickly. To give a couple of examples related to our own work, it is proved in [**?**] (respectively, [**?**]) that deciding whether a finite idempotent algebra generates a variety that is congruence-n-permutable (respectively, congruence-modular) is *tractable*. Our work continues this effort by presenting an efficient algorithm for deciding whether a finitely generated idempotent variety has a difference term.

2.4. **Algebras and Algorithms, Structure and Complexity.** In 2015, I joined a group of eight other scientists to form arguably one of the strongest active group of researchers working in universal algebra at American universities today. Our group was awarded a three-year NSF grant for the project, "Algebras and Algorithms, Structure and Complexity Theory." Our focus is on fundamental problems at the confluence of mathematical logic, algebra, and computer science, and the main goal of this effort is to deepen understanding of how to determine the complexity of certain types of computational problems. We focus primarily on classes of mathematical problems whose solutions yield new information about the complexity of CSPs. These include scheduling problems, resource allocation problems, and problems reducible to solving systems of linear equations. CSPs are theoretically solvable, but some are not solvable efficiently. Our work seeks to establish a clear boundary between the tractable and intractable cases, and to develop efficient algorithms for solutions in the tractable cases. My work on this project has culminated in the 50-page manuscript that I authored with Cliff Bergman entitled "Universal Algebraic Methods for Constraint Satisfaction Problems," which is available on the arXiv [**?**].

[2] To say a variety has *finite residual bound* is to say there is a finite bound on the size of the subdirectly irreducible members of the variety.