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* [11, 13]. Dually, progress in theoretical computer science has informed and inspired a substantial amount of pure mathematics in the last half-century [2, 3, 5, 6, 12, 13], 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* [4, 10, 8, Chs. 5 & 10]. The *Lean Programming Language* [1] 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_formal_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 2.3 below. Details can be found in the manuscript entitled "Universal Algebraic Methods for Constraint Satisfaction Problems," [7] 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 [1]. The title of this project is *Formal Foundations for Informal Mathematics Research* and a detailed project description is available in the document demeo_formal_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, working together with Myer and Ruskuc, 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. **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 [7].

[1]See [9] for the definition of a bounded lattice homomorphism.

References

[1] Jeremy Avigad, Mario Carneiro, Leonardo de Moura, Johannes Hölzl, and Sebastian Ullrich. The lean theorem proving language. GitHub.com, 2018. URL: https://leanprover.github.io/.

[2] Andrej Bauer. Five stages of accepting constructive mathematics. *Bull. Amer. Math. Soc. (N.S.)*, 54(3):481–498, 2017. doi:10.1090/bull/1556.

[3] Andrej Bauer. On fixed-point theorems in synthetic computability. *Tbilisi Math. J.*, 10(3):167–181, 2017. doi:10.1515/tmj-2017-0107.

[4] Andrej Bauer. Algebraic effects and handlers. OPLSS 2018, June 2018. Lecture notes available at: https://github.com/OPLSS/introduction-to-algebraic-effects-and-handlers; Recorded lecture available at: https://youtu.be/atYp386EGo8.

[5] Andrej Bauer and Jens Blanck. Canonical effective subalgebras of classical algebras as constructive metric completions. *J.UCS*, 16(18):2496–2522, 2010.

[6] Andrej Bauer, Gordon D. Plotkin, and Dana S. Scott. Cartesian closed categories of separable Scott domains. *Theoret. Comput. Sci.*, 546:17–29, 2014. doi:10.1016/j.tcs.2014.02.042.

[7] Clifford Bergman and William DeMeo. Universal algebraic methods for constraint satisfaction problems. Submitted to LMCS, 2016. URL: https://arxiv.org/abs/1611.02867.

[8] P. Chiusano and R. Bjarnason. *Functional Programming in Scala*. Manning Publications, 2014. URL: https://books.google.com/books?id=bmTRlwEACAAJ.

[9] Ralph Freese, Jaroslav Ježek, and J. B. Nation. *Free lattices*, volume 42 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 1995. URL: http://dx.doi.org/10.1090/surv/042, doi:10.1090/surv/042.

[10] J. Hughes. Why functional programming matters. *Comput. J.*, 32(2):98–107, April 1989. URL: http://dx.doi.org/10.1093/comjnl/32.2.98, doi:10.1093/comjnl/32.2.98.

[11] Martin Hyland and John Power. The category theoretic understanding of universal algebra: Lawvere theories and monads. In *Computation, meaning, and logic: articles dedicated to Gordon Plotkin*, volume 172 of *Electron. Notes Theor. Comput. Sci.*, pages 437–458. Elsevier Sci. B. V., Amsterdam, 2007. URL: https://doi-org.colorado.idm.oclc.org/10.1016/j.entcs.2007.02.019, doi:10.1016/j.entcs.2007.02.019.

[12] Don Pigozzi and Antonino Salibra. Lambda abstraction algebras: representation theorems. *Theoret. Comput. Sci.*, 140(1):5–52, 1995. Selected papers of AMAST '93 (Enschede, 1993). URL: http://dx.doi.org/10.1016/0304-3975(94)00203-U, doi:10.1016/0304-3975(94)00203-U.

[13] Dana S. Scott. A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theoret. Comput. Sci.*, 121(1-2):411–440, 1993. A collection of contributions in honour of Corrado Böhm on the occasion of his 70th birthday. URL: http://dx.doi.org/10.1016/0304-3975(93)90095-B, doi:10.1016/0304-3975(93)90095-B.