

The Agda UALib and Birkhoff's Theorem in Martin-Löf Dependent Type Theory

William DeMeo   

Department of Algebra, Charles University in Prague

Abstract

The Agda Universal Algebra Library (UALib) is a library of types and programs (theorems and proofs) we developed to formalize the foundations of universal algebra in Martin-Löf dependent type theory using the Agda programming language and proof assistant. This paper describes the UALib and demonstrates how it makes Agda more accessible to working mathematicians (like ourselves) as a tool for formally verifying “known” results in general algebra and related fields, as well as for discovering new theorems in these areas. The library already includes a substantial collection of definitions, theorems, and proofs from universal algebra and as such provides many examples that exhibit the power of inductive and dependent types for representing and reasoning about algebraic and relational structures.

The first major milestone of the UALib project is a complete proof of Birkhoff's HSP Theorem. To the best of our knowledge, this is the first time Birkhoff's Theorem has been formulated and proved in dependent type theory and verified with a proof assistant.

In this paper we describe the UALib and the formal proof of Birkhoff's theorem, discussing some of the challenges we faced and how these hurdles were overcome. In so doing, we illustrate the effectiveness of dependent type theory, Agda, and the UALib for proving and verifying theorems in universal algebra.

2012 ACM Subject Classification Theory of computation → Constructive mathematics; Theory of computation → Type theory; Theory of computation → Logic and verification; Computing methodologies → Representation of mathematical objects; Theory of computation → Type structures

Keywords and phrases Universal algebra, Equational logic, Martin-Löf Type Theory, Birkhoff's HSP Theorem, Formalization of mathematics, Agda, Proof assistant

Supplementary Material

Documentation: <https://ualib.org>

Software: <https://gitlab.com/ualib/ualib.gitlab.io.git>

Acknowledgements The author wishes to thank Hyeyoung Shin and Siva Somayyajula for their contributions to this project and Martin Hötzel Escardo for creating the Type Topology library and teaching a course on Univalent Foundations of Mathematics with Agda at the 2019 Midlands Graduate School in Computing Science.

Contents

1 Introduction

To support formalization in type theory of research level mathematics in universal algebra and related fields, we present the Agda Universal Algebra Library (Agda UALib), a software library containing formal statements and proofs of the core definitions and results of universal algebra. The Agda UALib is written in Agda [?], a programming language and proof assistant based on Martin-Löf Type Theory that not only supports dependent and inductive types, but also provides powerful *proof tactics* for proving things about the objects that inhabit these types.

There have been a handful of other successful efforts to formalize parts of universal algebra in type theory, most notably

- Capretta [?] (1999) formalized the basics of universal algebra in the Calculus of Inductive Constructions using the Coq proof assistant;
- Spitters and van der Weegen [?] (2011) formalized the basics of universal algebra and some classical algebraic structures, also in the Calculus of Inductive Constructions using the Coq proof assistant, and promoted the use of type classes as a preferable alternative to setoids.
- Gunther, et al [?] developed what seems to have been the most extensive library formalizing universal algebra to date, including the formalization of some equational logic; this project (like the UALib) uses Martin-Löf dependent type theory and the Agda proof assistant.

Some other projects aimed at formalizing mathematics generally, and algebra in particular, have developed into very extensive libraries that include definitions, theorems, and proofs about algebraic structures, such as groups, rings, fields, and modules. However, the goals of these other projects do not seem to be a formal library of definitions, theorems, and proofs about *general* (universal) algebra.

Although the UALib project was initiated relatively recently (by the author of this paper, with valuable contributions from Siva Somayyajula), the part of universal algebra and equational logic that we formalize already extends beyond the scope of prior efforts such as those mentioned above. In particular, the UALib includes the only formal, constructive, machine-checked proof of Birkhoff’s variety theorem that we know of. We remark that, with the exception of [?], all other proofs of Birkhoff’s theorem are informal and not known to be constructive.

The seminal idea for this project was the observation that, on the one hand, a number of basic and important constructs in universal algebra can be defined recursively, and theorems about them have easy proofs by structural induction, while, on the other hand, inductive types (of dependent type theory) make possible very precise formal representations of recursively defined objects, and often yield very short, elegant and constructive proofs of properties of such objects. An important feature of such proofs in type theory is that they are *total functional programs* and, as such, they are computable and composable. These observations suggest that much can be gained from implementing universal algebra in a language, like Martin-Löf type theory, that supports dependent and inductive types.

1.1 Objectives

The first goal of the project is to express the foundations of universal algebra constructively, in type theory, and to formally verify the foundations using the Agda proof assistant. Thus we aim to codify the edifice upon which our mathematical research stands, and demonstrate that advancements in our field can be expressed in type theory and formally verified in a way that we, and other working mathematicians, can easily understand and check the results. We hope the library inspires and encourages others to formalize and verify their own mathematics research so that we may more easily understand and verify their results.

Our field is deep and broad, so codifying all of its foundations may seem like a daunting task and a risky investment of time and resources. However, we believe the subject is well served by a new, modern, *constructive* presentation of its foundations. Finally, the mere act of reinterpreting the foundations in an alternative system offers a fresh perspective, and this often leads to deeper insights and new discoveries.

Indeed, we wish to emphasize that our ultimate objective is not merely to translate existing results into a new more modern and formal language. Rather, an important goal of the UALib project is a system that is useful for conducting research in mathematics, and that is how we intend to use our library now that we have achieved our initial objective of implementing a substantial part of the foundations of universal algebra in Agda.

In our own mathematics research, experience has taught us that a proof assistant equipped with specialized libraries for universal algebra, as well as domain-specific tactics to automate proof idioms of our field, would be extremely valuable and powerful tool. Thus, we aim to build a library that serves as an indispensable part of our research tool kit.

1.2 Contributions

Apart from the library itself, we describe the formal implementation and proof of a deep result in universal algebra, which was among the first major results of our subject—namely, Garrett Birkhoff’s celebrated HSP Theorem [?]. This theorem says that a *variety* (a class of algebras closed under quotients, subalgebras, and products) is an equational class. More precisely, a class of algebras is closed under the taking of quotients, subalgebras, and arbitrary products if and only if it is the class of all algebras that satisfy some set of equations.

The fact that we now have a proof of Birkhoff’s Theorem in Agda is noteworthy, not only because this is the first time the theorem has been proved in dependent type theory and verified with a proof assistant, but also because our proof is *constructive*. Judging from the paper [?] by Carlström, for example, it is evidently a nontrivial matter to take a well-known informal proof of a theorem like Birkhoff’s (as presented in, e.g., [?]) and show that it can be formalized using only constructive logical assumptions—that is, without appealing to the Law of the Excluded Middle and (therefore) without the Axiom of Choice.