Lessons from Formalizing (Higher) Category Theory

Benedikt Ahrens[1] and Niels van der Weide[2]

[1] Delft University of Technology, Netherlands
B.P.Ahrens@tudelft.nl
[2] Radboud University Nijmegen, The Netherlands
nweide@cs.ru.nl

1 Summary

UniMath [VAG+] is a library of univalent mathematics written in Coq. In the past few years, it has seen growth mostly in the area of category theory and its applications to programming language semantics (e.g., [AMvdWW24, vdWRAN24, ANvdW23, AMM22, AHLM21, vdW23, VvdW21, BHL20].

We propose to give an overview of the UniMath library, some of the challenges that we are facing in developing and maintaining it, and our solutions to these challenges. We summarize these challenges here and give more detailed information below.

One challenge we encounter is the need to build large mathematical structures. For instance, among the objects we are studying are bicategories, which comprise 15 fields of data and 22 fields of properties. Verity double bicategories, formalized in [vdWRAN24], have 121 fields altogether. We have developed "displayed" machinery to build such large structures modularly.

Another challenge we encounter is the need to carefully distinguish between *weak* and *strict* mathematical structures. We discuss the difference between strict and weak (higher) categories, and how to formalize these notions in an intensional setting.

2 Modular Constructions

In category theory, one is frequently interested in categories whose objects and morphisms are given by, for instance, sets with additional structure. There are numerous examples of this, for instance, the category of groups or the category of rings. General examples are given by algebras for a functor on sets and by algebras for a monad.

There are some challenges when using such categories in a formalization. For instance, frequently one is interested in lifting functors F to the category of algebras (see, for instance, [HJ98, Theorem 2.14 and Corollary 2.15] and [vdWG19, Proposition 6.2]. In a formalization, the lifted functor should be defined in a modular way: we want to construct structure and properties of F from which one can obtain the lift directly.

Displayed categories [AL19] give a way to construct categories, functors, and natural transformations in a modular way. Basically, a displayed category over a category describes properties and structures to be added to the objects and morphisms of that category. We can represent the category of groups as a displayed category over the category of sets: the objects over a set is the collection of group structures, whereas the morphisms over a function between two group structures are given by proofs that this function is a homomorphism. Note that we can similarly define notions of displayed functors and displayed natural transformations.

The usage of displayed categories makes the notion of a "structure" explicit. Intuitively, the following is happening: there are two ways to define groups. We could define groups by describing them via a single record type containing the fields for the carrier, operations, and laws. However, we could also describe a record that is parameterized by a type, and that record represents a group over that type of elements.

Displayed categories are used extensively in UniMath. They are used to modularly construct categories and in the study of fibrations and the semantics of type theory. In addition, they are generalized to displayed bicategories as well, and they play a prominent role to construct bicategories in a modular fashion.

3 Weakness versus Strictness

In (higher) category theory, we distinguish between, on the one hand, *strict* structures, where sameness within a structure is expressed using equality, and, on the other hand, *weak* structures, where sameness is expressed using a notion of equivalence provided by the structure itself. Traditionally, strict structures have been considered to be easier to work with. The reason is that equality traditionally is *substitutive*; equals can be substituted for each other in any context. This means that we can always replace $f \circ (g \circ h)$ by $(f \circ g) \circ h$ or vice versa. In such a setting, coherence theorems are very useful, because it allows us to replace weak structures by strict structures.

The situation is different in intensional foundations. Instead of being substitutive, the Martin-Löf identity type is *transportational*: replacing something by something identical leaves a trace in form of a transport. This greatly reduces the usefulness of strict structures. More concretely, since we cannot replace $f \circ (g \circ h)$ by $(f \circ g) \circ h$ without leaving a trace in the resulting term, it is not possible to use strictness to simplify any coherence diagram.

Let us make this concrete for adjunctions. If we have an adjunction $F \dashv U$ then the unit η and counit ε satisfy the triangle equations, and one of them says $\varepsilon L \circ L\eta = \mathrm{id}$. In extensional foundations, this equation is well-typed, because $L = L \circ \mathrm{id}$ and because $L \circ (R \circ L) = (L \circ R) \circ L$. However, this equation is not well-typed in intensional foundations, because in that setting, one has to decorate the term with suitable unitors and associators. As a consequence, bicategories got more consideration than 2-categories in UniMath in contrast to classical foundations where it is the other way around.

4 Transport along Equivalences

UniMath uses univalent foundations [Uni13], and univalent categories play a prominent role in it [AKS15]. In univalent categories, isomorphism of objects is the same as identity of objects. In addition, equivalences of univalent categories are the same as identities of univalent categories. As such, statements about isomorphisms and equivalences can be proven by induction.

A concrete application of univalence is given by transporting structure and properties of categories. For instance, suppose we have categories C and D, an equivalence e from C to D, and suppose that D is locally Cartesian closed. Proving that C is locally Cartesian closed usually requires a technical and tedious proof. However, if both C and D are univalent, then the proof becomes trivial. because by induction, we can assume that e is the identity.

Acknowledgements We gratefully acknowledge the work by the Coq development team in providing the Coq proof assistant and surrounding infrastructure, as well as their support in keeping UniMath compatible with Coq.

References

[AHLM21] Benedikt Ahrens, André Hirschowitz, Ambroise Lafont, and Marco Maggesi. Presentable signatures and initial semantics. *Log. Methods Comput. Sci.*, 17(2), 2021.

[AKS15] Benedikt Ahrens, Krzysztof Kapulkin, and Michael Shulman. Univalent categories and the rezk completion. *Math. Struct. Comput. Sci.*, 25(5):1010–1039, 2015.

[AL19] Benedikt Ahrens and Peter LeFanu Lumsdaine. Displayed categories. *Log. Methods Comput. Sci.*, 15(1), 2019.

[AMM22] Benedikt Ahrens, Ralph Matthes, and Anders Mörtberg. Implementing a category-theoretic framework for typed abstract syntax. In Andrei Popescu and Steve Zdancewic, editors, *CPP '22: 11th ACM SIGPLAN International Conference on Certified Programs and Proofs, Philadelphia, PA, USA, January 17 - 18, 2022*, pages 307–323. ACM, 2022.

[AMvdWW24] Benedikt Ahrens, Ralph Matthes, Niels van der Weide, and Kobe Wullaert. Displayed monoidal categories for the semantics of linear logic. In Amin Timany, Dmitriy Traytel, Brigitte Pientka, and Sandrine Blazy, editors, *Proceedings of the 13th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2024, London, UK, January 15-16, 2024*, pages 260–273. ACM, 2024.

[ANvdW23] Benedikt Ahrens, Paige Randall North, and Niels van der Weide. Bicategorical type theory: semantics and syntax. *Math. Struct. Comput. Sci.*, 33(10):868–912, 2023.

[BHL20] Peio Borthelle, Tom Hirschowitz, and Ambroise Lafont. A cellular howe theorem. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 273–286. ACM, 2020.

[HJ98] Claudio Hermida and Bart Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. Comput.*, 145(2):107–152, 1998.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* `https://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.

[VAG+] Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson, et al. Unimath — a computer-checked library of univalent mathematics. available at `http://unimath.org`.

[vdW23] Niels van der Weide. The formal theory of monads, univalently. In Marco Gaboardi and Femke van Raamsdonk, editors, *8th International Conference on Formal Structures for Computation and Deduction, FSCD 2023, July 3-6, 2023, Rome, Italy*, volume 260 of *LIPIcs*, pages 6:1–6:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[vdWG19] Niels van der Weide and Herman Geuvers. The construction of set-truncated higher inductive types. In Barbara König, editor, *Proceedings of the Thirty-Fifth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2019, London, UK, June 4-7, 2019*, volume 347 of *Electronic Notes in Theoretical Computer Science*, pages 261–280. Elsevier, 2019.

[vdWRAN24] Niels van der Weide, Nima Rasekh, Benedikt Ahrens, and Paige Randall North. Univalent double categories. In Amin Timany, Dmitriy Traytel, Brigitte Pientka, and Sandrine Blazy, editors, *Proceedings of the 13th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2024, London, UK, January 15-16, 2024*, pages 246–259. ACM, 2024.

[VvdW21] Niccolò Veltri and Niels van der Weide. Constructing higher inductive types as groupoid quotients. *Log. Methods Comput. Sci.*, 17(2), 2021.