Case Study on Displayed Monoidal Categories

Benedikt Ahrens[1,2], Ralph Matthes[3], and Kobe Wullaert[1]

[1] Delft University of Technology, The Netherlands
[2] University of Birmingham, United Kingdom
[3] IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse, France

A new formalization of monoidal categories A monoidal structure on a category C is given by a tensor product $\otimes$, that is, a functorial binary operation on the objects and morphisms of C. Furthermore there is a unit object I, that is neutral for the tensor operation modulo (natural) isomorphism—not "on the nose", i. e., not with propositional equality. The tensor is associative up to isomorphism, and a pentagon law holds for that isomorphism, as well as a triangle law connecting all three isomorphisms.

Monoidal categories abound in mathematics (a prime example being the vector spaces over a given field) and are also well present in theoretical computer science. Functors between categories are accordingly extended to monoidal functors, so as to ensure preservation of the extra structure. The operations doing this for tensor and unit have to interact properly with the aforementioned isomorphisms. *Strong* monoidal functors need these operations to be isomorphisms, while *strict* monoidal functors satisfy the laws "on the nose". Monoidal functors abstractly capture the notion of "homomorphism" in the case of one binary operation and one constant when not only objects but also their morphisms matter (which is typical of constructive logic).

In textbooks definitions, the tensor product is seen as a bifunctor on C, i. e., a functor from $\mathsf{C} \times \mathsf{C}$ to C. Our previous attempts at defining displayed monoidal categories (see next paragraph) on that basis suffered from major difficulties with transport along components of pairs arising with the use of this product category $\mathsf{C} \times \mathsf{C}$. Instead of working with two-place functions (encoded by pairing), one can move to a curried view that first takes the left argument and then is a function that expects the right-hand side argument—which is good for the object mapping. For the two-place morphism mapping, we employ a symmetric approach, by considering the one-place mappings where the left resp. right argument is fixed to the identity, which we call the left resp. right *whiskering*, respectively. This notion is not confined to the tensor of a monoidal category but is an alternative view for any bifunctor $\mathsf{A} \times \mathsf{B} \to \mathsf{C}$. However, calling it whiskering comes from the analogous treatment of horizontal composition in bicategories in the UniMath library.

As a benefit, the formal development of monoidal categories in this format is in close correspondence with bicategories (as they are formalized in UniMath [1, Definition 2.1])—mathematically, monoidal categories are just one-object bicategories. Still, the full definition of bicategories is much heavier than the definition of monoidal categories we are obtaining, and working with one-object instances of the general bicategorical theory did not seem an option.

For lack of space, we cannot detail the other steps to getting monoidal categories and their strict or strong functors, but the readers can consult the files with the string `Whiskered` in the name in the formalization (the 1.7kloc are approximately half vernacular and half proofs, according to `coqwc`).[1]

A formalization of displayed monoidal categories A displayed category D over some base category C ([2] involving the first author) has more than just the data of a category; it reflects the construction process done on the objects and morphisms of C. However, there is a

[1] https://github.com/UniMath/UniMath/blob/1580dab0/UniMath/CategoryTheory/Monoidal

generic construction of an "ordinary" category from D, the total category $\int \mathsf{D}$, and a forgetful functor π_1 down back to C. The notion of displayed category has led to "displayed versions" of numerous categorical concepts and is seen to have potential for much more [5]. In general, "displaying" means constructing out of the ingredients of the underlying structure in a concise way avoiding copying as much as possible.

Of special interest to our application is the identification of a class of functors F from C to $\int \mathsf{D}$ that have $\pi_1 \circ F = 1_{\mathsf{C}}$. This has been formalized in the UniMath library as "sections",[2] for which such a functor F can be generically constructed. Of course, this is not formulated in terms of the total category but gives adaptations of the functor laws to hold "over" C. By working with sections, we efficiently replace the very cumbersome use of $\pi_1 \circ F = 1_{\mathsf{C}}$ as an equation between functors for rewriting (functor equality is very bad for rewrites from the point of view of intensional type theory) by definitional equality. Let us mention that displayed notions in general provide better behaviour w. r. t. equality reasoning: much less transport operations along equational hypotheses are needed than when directly working with the total categories.

Given a monoidal category C in the whiskered format and a displayed category D over (the base category of) C, we add a tensor "over" the tensor of C (as a displayed bifunctor, similarly defined in whiskered format) and add the other ingredients and laws to have a definition of displayed monoidal category D^+. A corresponding total (curried) monoidal category $\int \mathsf{D}^+$ is then obtained, with a forgetful functor π_1 back to C that we show to be strict monoidal.

Then, we identify a class of strong monoidal functors F from C to $\int \mathsf{D}^+$ s. t. $\pi_1 \circ F = 1_{\mathsf{C}}$ with a concise description in terms of D^+, which gives the notion of strong monoidal sections.

Application scenario Recent work [3, §4.3] involving the first and second authors attempted to establish a bijection between a class of parameterized distributivities (given actions in the sense of Janelidze and Kelly [4] as strong monoidal functors into some functor category) and the monoidal sections for a specially crafted displayed monoidal category (the interested reader may consult the high-level description there—the details go far beyond the capacity of the present abstract). However, in that attempt, the authors tried to work with a naive definition of monoidal sections based on the "classical" bifunctorial view of the tensor; this definition generated many problems with transport, lack of implicit argument synthesis of Coq and an unpleasant need for re-packaging tuples. Consequently, for that work, only a function in one direction could be constructed.

Using our new definition of monoidal category and the resulting workable definition of displayed monoidal category, we have been able to construct the full bijection: we have constructed the missing function, using the aforementioned monoidal sections of that displayed monoidal category as its domain, and shown that the two functions are inverse to each other (for a bicategorical generalization). We have thus solved the open question of [3, §4.3]; furthermore, the compilation time of the respective file[3] is 53% less than for the original one.[4]

Conclusion We have introduced displayed monoidal categories, with a focus on implementation and use in the UniMath library. The case study of 2.1kloc (code mostly not written from scratch but adapted from the earlier approach) validates this approach. As future work, we see basing also the notion of action-based strength on our new format for monoidal categories.

[2] https://github.com/UniMath/UniMath/blob/1580dab0/UniMath/CategoryTheory/DisplayedCats/Constructions.v#L399

[3] https://github.com/UniMath/UniMath/blob/1580dab0/UniMath/Bicategories/MonoidalCategories/ActionBasedStrongFunctorsWhiskeredMonoidal.v

[4] 43s versus 92s wall clock time measured on current Intel processor in single-thread compilation.

References

[1] Benedikt Ahrens, Dan Frumin, Marco Maggesi, Niccolò Veltri, and Niels van der Weide. Bicategories in univalent foundations. *Mathematical Structures in Computer Science*, 2022. 38 pages, online publication at https://doi.org/10.1017/S0960129522000032.

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

[3] 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.

[4] George Janelidze and Gregory Maxwell Kelly. A note on actions of a monoidal category. *Theory and Applications of Categories*, 9(4):61–91, 2001. Online available at http://www.tac.mta.ca/tac/volumes/9/n4/9-04abs.html.

[5] Jonathan Sterling and Carlo Angiuli. Foundations of relative category theory. Online lecture course, available at https://www.jonmsterling.com/math/lectures/categorical-foundations.html, 2022.