Formalizing Double Groupoids and Cross Modules in the Lean Theorem Prover

Jakob von Raumer

University of Nottingham, United Kingdom
psxjv4@nottingham.ac.uk,
http://www.cs.nott.ac.uk/ psxjv4/

Abstract. Lean is a new open source dependently typed theorem prover which is mainly being developed by Leonardo de Moura at Microsoft Research. It is suited to be used for proof irrelevant reasoning as well as for proof relevant formalizations of mathematics. In my talk, I will present my experiences doing a formalization project in Lean. One of the interesting aspects of homotopy type theory is the ability to perform synthetic homotopy theory on higher types. While for the first homotopy group the choice of a suitable algebraic structure to capture the homotopic information is obvious – it's a group –, implementing a structure to capture the information about both the first and the second homotopy group (or groupoid) of a type and their interactions is more involved. Following Ronald Brown's book on Nonabelian Algebraic Topology, I formalized two structures: Double groupoids with thin structures and crossed modules on groupoids. I furthermore attempted to prove their equivalence. The project can be seen as a usability and performance test for the new theorem prover.

Keywords: Formalization of Mathematics, Algebraic Topology

1 Introduction

Making mathematical definitions and theorem proofs readable and verifiable by computers has become increasingly important in the last years, not only since there are proofs that are hard or impossible to be checked by a single person due to their size (one example being Tom Hales' proof of the Kepler conjecture). With the rise of formally verified software, one also wants the same level of trust for the mathematical theories whose soundness guarantee the correct functionality of the program. Fields where formal verification has been successfully used to certify computer programs include cryptography and aerospace industry. These rely heavily on results from algebra and calculus and differential equations.

Homotopy type theory (HoTT) can serve as a foundation of mathematics that is better suited to fit the needs of formalizing certain branches of mathematics, especially the ones of *topology.* In traditional, set-based approaches to formalizing the world of mathematical knowledge, topological spaces and their properties have to be modeled with much effort by referring to the type of real numbers.

In contrast to this, homotopy type theory, in a certain sense, contains topologically motivated objects like fibrations and homotopy types as primitives. This makes it much easier and more natural to reason about topological properties of these objects. Homotopy type theory is a relatively new field but it already has produced several useful implementations and libraries in interactive theorem provers like Agda and Coq. One important feature of homotopy type theory is that it is *constructive* and thus allows to extract programs from definitions and proofs.

Homotopy type theory is *proof relevant* which means that there can be distinct (and internally distinguishable) proofs for one statement. This leads to the fact that types in HoTT bear the structure of a higher groupoid in their identities. The essential problem in the field of *homotopy* is to analyze this structure of paths and iterated paths between paths in topological spaces or, in the world of HoTT, in higher types. This happens by considering the algebraic properties of the homotopy groups or *homotopy groupoids* of the spaces resp. types.

In his book "Nonabelian Algebraic Topology" [1], Ronald Brown introduces the notion of *double groupoids with thin structures* and *crossed modules over groupoids* to describe the interaction between the first and the second homotopy groupoid of a space algebraically. Brown's approach, preceding the discovery of homotopy type theory by a few decades, is formulated entirely classically and set-based.

I will describe how I translated some of the central definitions and lemmas from his book to dependently typed algebraic structures in homotopy type theory, made them applicable to the analysis of 2-truncated types by creating the notion of a *fundamental double groupoid of a presented 2-type*, and then formalized them in the newly built interactive theorem proving system Lean [2].

2 Double Categories and Double Groupoids

Seeing a (small) category as a tuple of object set, morphism set, domain and codomain functions, identity function and composition, Brown defines double categories similar to the following:

A **double category** D is given by the following data: Three sets D_0, D_1, and D_2, the elements of which are respectively called **0-, 1- and 2-cells**, together with maps ∂^-, ∂^+, ϵ, $\circ_D$, ∂_1^-, ∂_1^+, ϵ_1, $\circ_1$, ∂_2^-, ∂_2^+, ϵ_2, and $\circ_2$ that make these sets form three categories:

- a category $(D_0, D_1, \partial^-, \partial^+, \epsilon, \circ_D)$ on D_0, often called the **(1-)skeleton** of the double category,
- a **vertical category** $(D_1, D_2, \partial_1^-, \partial_1^+, \epsilon_1, \circ_1)$, and
- a **horizontal category** $(D_1, D_2, \partial_2^-, \partial_2^+, \epsilon_2, \circ_2)$.

The mentioned maps are required to satisfy the following **cubical identities**:

$$\partial^- \circ \partial_1^+ = \partial^- \circ \partial_2^-,$$
$$\partial^- \circ \partial_1^+ = \partial^+ \circ \partial_2^-,$$

$$\begin{aligned} \partial^+ \circ \partial_1^- &= \partial^- \circ \partial_2^+, \\ \partial^+ \circ \partial_1^+ &= \partial^+ \circ \partial_2^+, \end{aligned}$$

$$\begin{aligned} \partial_1^- \circ \epsilon_2 &= \epsilon \circ \partial^-, \\ \partial_1^+ \circ \epsilon_2 &= \epsilon \circ \partial^+, \\ \partial_2^- \circ \epsilon_1 &= \epsilon \circ \partial^-, \\ \partial_2^+ \circ \epsilon_1 &= \epsilon \circ \partial^+, \text{ and} \\ \epsilon_1 \circ \epsilon &= \epsilon_2 \circ \epsilon =: 0. \end{aligned}$$

The boundary and degeneracy maps of the vertical category are furthermore assumed to be a homomorphism with respect to the composition of the horizontal category, and vice versa:

$$\begin{aligned} \partial_2^-(v \circ_1 u) &= \partial_2^-(v) \circ_D \partial_2^-(u), \\ \partial_2^+(v \circ_1 u) &= \partial_2^+(v) \circ_D \partial_2^+(u), \\ \partial_1^-(v \circ_2 u) &= \partial_1^-(v) \circ_D \partial_1^-(u), \\ \partial_1^+(v \circ_2 u) &= \partial_1^+(v) \circ_D \partial_1^+(u), \\ \epsilon_2(g \circ_D f) &= \epsilon_2(g) \circ_1 \epsilon_2(f), \text{ and} \\ \epsilon_1(g \circ_D f) &= \epsilon_1(g) \circ_2 \epsilon_1(g), \end{aligned}$$

for each $f, g \in D_1$ and $u, v \in D_2$ where the compositions are defined.

A last condition, called the **interchange law**, has to be fulfilled: For each $u, v, w, x \in D_2$,

$$(x \circ_2 w) \circ_1 (v \circ_2 u) = (x \circ_1 v) \circ_2 (w \circ_1 u)$$

has to hold wherever it is well-defined.

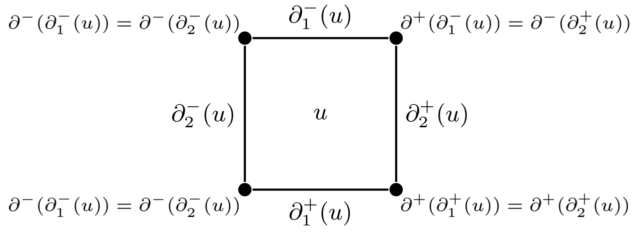
$\partial^-(\partial_1^-(u)) = \partial^-(\partial_2^-(u))$
$\partial_1^-(u)$
$\partial^+(\partial_1^-(u)) = \partial^-(\partial_2^+(u))$
$\partial_2^-(u)$
u
$\partial_2^+(u)$
$\partial^-(\partial_1^-(u)) = \partial^-(\partial_2^-(u))$
$\partial_1^+(u)$
$\partial^+(\partial_1^+(u)) = \partial^+(\partial_2^+(u))$

Fig. 1. A square $u \in D_2$, its faces, and its corners.

In more pictorial words, double categories do not only contain objects and morphisms (lines), but also square-shaped two cells (see figure 1). These can be composed vertically or horizontally, given that their edges match. There are

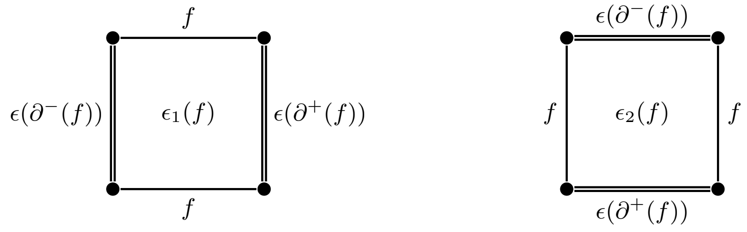
f
$\epsilon(\partial^-(f))$
$\epsilon_1(f)$
$\epsilon(\partial^+(f))$
f
$\epsilon(\partial^-(f))$
f
$\epsilon_2(f)$
f
$\epsilon(\partial^+(f))$

Fig. 2. Degenerate squares of the vertical and horizontal category for a given line $f \in D_1$. Degenerate lines are drawn as double lines.

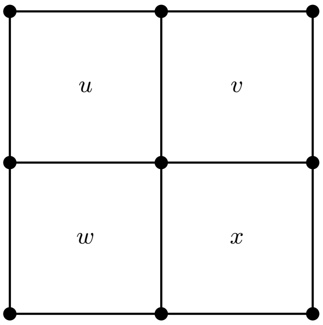

Fig. 3. The grid we use to illustrate the composition $(x \circ_2 w) \circ_1 (v \circ_2 u)$ as well as $(x \circ_1 v) \circ_2 (w \circ_1 u)$, which are identical by the interchange law.

squares which act as the identity with respect one of the composition (see figure 2), and when composing in a 2-by-2 grid, it doesn't matter whether we give precedence to vertical or to horizontal composition (see figure 3) To prevent the necessity of the composition of a partial function, we make the the type of two-cells depend on its boundary when we translate the definition to one in type theory:

We define a **double category** to be a record containing the following:

- The object set $D_0 : \mathsf{Set}$,
- A *precategory* (here, "pre" means that isomorphic in that category are not necessarily equal) on D_0, consisting of:
  - A type family of morphisms $D_1 : \prod_{(a,b:D_0)} \mathsf{Set}$.
  - The composition of morphisms
$$\circ : \prod_{a,b,c:D_0} D_1(b,c) \to D_1(a,b) \to D_1(a,c).$$
  - An identity operator $\mathrm{id} : \prod_{(a:D_0)} D_1(a,a)$.
  - A witness ensuring associativity for all morphisms:
$$\prod_{a,b,c,d:D_0} \prod_{h:D_1(c,d)} \prod_{g:D_1(b,c)} \prod_{f:D_1(a,b)} h \circ (g \circ f) = (h \circ g) \circ f$$

- Witnesses that the identity morphisms are neutral with respect to composition from the left and from the right:

$$\prod_{a,b:D_0} \prod_{f:D_1(a,b)} (\mathrm{id}(b) \circ f = f) \times (f \circ \mathrm{id}(a) = f)$$

- A set family of two-cells:

$$D_2 : \prod_{a,b,c,d:D_0} \prod_{f:D_1(a,b)} \prod_{g:D_1(c,d)} \prod_{h:D_1(a,c)} \prod_{i:D_1(b,d)} \mathsf{Set}$$

We will always leave the first four parameters implicit and write $D_2(f, g, h, i)$ for the type of two-cells with f as their upper face, g as their bottom face, h as their left face, and i as their right face.

- The vertical composition operation: For all $a, b, c_1, d_1, c_2, d_2 : D_0$ and $f_1 : D_1(a, b)$, $g_1 : D_1(c_1, d_1)$, $h_1 : D_1(a, c_1)$, $i_1 : D_1(b, d_1)$, $g_2 : D_1(c_2, d_2)$, $h_2 : D_1(c_1, c_2)$, and $i_2 : D_1(d_1, d_2)$ the composition of two cells

$$v \circ_1 u : D_2(g_1, g_2, h_2, i_2) \to D_2(f, g_1, h_1, i_1) \to D_2(f_1, g_2, h_2 \circ h_1, i_2 \circ i_1).$$

- The vertical identity $\mathrm{id}_1 : \prod_{(a,b:D_0)} \prod_{(f:D_1(a,b))} D_2(f, f, \mathrm{id}(a), \mathrm{id}(b))$.
- For all $w : D_2(g_2, g_3, h_3, i_3)$, $v : D_2(g_1, g_2, h_2, i_2)$, and $u : D_2(f, g_1, h_1, i_1)$ a witness for the associativity of the vertical composition $\mathsf{assoc}_1(w, v, u)$ in

$$\mathsf{assoc}(i_3, i_2, i_1)_*(\mathsf{assoc}(h_3, h_2, h_1)_*(w \circ_1 (v \circ_1 u))) = (w \circ_1 v) \circ_1 u,$$

where assoc is the associativity proof in the 1-skeleton. The transport is required since the cells at the left and right side of the equation do not definitionally have the same set of faces.

- Horizontal composition $\circ_2$ and horizontal identity id_2.
- Finally, we need witnesses that the axioms of a double category, as stated in the definition above, hold. Note that there are only four of these rules which are not yet expressed by the types of composition and identity.

3 The Fundamental Double Groupoid

How can we now use this structure to characterize types in homotopy type theory? The role of the basepoint in the consideration of the fundamental group of a type or a set of basepoints for the fundamental double groupoid, we need a *presentation* relative to which we express the fundmental double groupoid:

We define a **presented 2-type** to be a triple (X, A, C) of types $X, A, C : \mathcal{U}$ together with functions $\iota : C \to A$ and $\iota' : A \to X$ where X is a 2-type, A is a 1-type, and C is a set.

From each presented 2-type (X, A, C) we receive its **fundamental double category** G by defining

$$\begin{aligned} G_0 &:\equiv C \\ G_1(a, b) &:\equiv \iota(a) =_A \iota(b) \\ G_2(f, g, h, i) &:\equiv \mathsf{ap}_{\iota'}(h) \cdot \mathsf{ap}_{\iota'}(g) =_X \mathsf{ap}_{\iota'}(f) \cdot \mathsf{ap}_{\iota'}(i) \end{aligned}$$

for all $a, b : C$, $f : \iota(a) = \iota(b)$, $g : \iota(c) = \iota(d)$, $h : \iota(a) = \iota(c)$, and $i : \iota(b) = \iota(d)$.

Omitted from this abstract, we can then account for the symmetry of the identity relation by extending the definition to the one of a **(weak) double groupoid** and for the fact that each commuting square boundary gives rise to a homotopically degenerate square filler by equipping the double groupoid with a **thin structure** or, equivalently, **connections**. From the category of double groupoids we can then switch to a more "flat" representation by transforming these into **crossed modules**. Future work will include a formulation of the statement and proof of a Seifert-van Kampen theorem which yields the fundamental double groupoid of certain pushouts of types.

References

1. Ronald Brown, Philip J Higgins, and Rafael Sivera. *Nonabelian Algebraic Topology: Filtered spaces,crossed complexes,cubical homotopy groupoids.* European Mathematical Society, 2011.
2. Leonardo de Moura, Soonho Kong, Floris van Doorn, and Jakob von Raumer. *The Lean Theorem Prover (System Description).* In Automated Deduction-CADE-25 (pp. 378-388). Springer International Publishing.