Group Cohomology in Cubical Agda

Kelton OBrien [1]

[1]Indiana University

Synthetic Mathematics with Homotopy Type Theory

Homotopy Type Theory (HoTT) is an extension to Martin-Löf (dependent) Type Theory that adds a new axiom characterizing the structure of higher equalities (the univalence axiom), and new types equipped with constructors for the equality type called higher inductive types (HITs). These extensions allow us to interpret types as topological spaces, elements of types as points in those spaces, equalities as paths between those points, and equalities between equalities as homotopies of paths.

Instead of constructing our abstractions from base components that do not appear to have anything to with spaces until we organize them in the correct fashion (like sets), we are able to take these topological notions as primitive, and manipulate them directly. This, as well as the computational nature of type theory, allows for the formalization of homotopy theory and algebraic topology more generally in a way that was impossible before.

Cubical Agda is a proof assistant that uses a variant of HoTT known as cubical type theory, allowing formalization and computer checking of synthetic algebraic-topological proofs. There is a large body of verified work in HoTT based proof assistants, but despite this, **Group Cohomology**, a key notion in the field of homological algebra, has not been formalized.

Background: (Co)Homology

The **homology** of a topological space is an algebraic invariant that organizes the information relating to the n-dimensional holes in a space. To compute the homology of a topological space, one first must associate to it a *chain complex*. A chain complex is a sequence of abelian groups/R-modules C_n with homomorphisms $d_n : C_n \to C_{n-1}$ known as *boundary operators* between them, such that $d_n \circ d_{n+1} = 0$ for all n. We view elements of each of these groups C_n as (formal sums of) n-dimensional topological objects of some kind (simplicies, cells, ...), and the boundary operators as functions taking each of the elements to their (topological) boundary (e.g taking a simplex to its faces).

Once we have a chain complex, the definition of its nth homology group H_n is simply

$$H_n = ker(d_n)/im(d_{n+1})$$

We view elements of $ker(d_n)$ as n-dimensional cycles, and elements of $im(d_{n+1})$ as the boundaries of $n+1$ dimensional objects, so H_n then keeps track of all of the cycles, except those that are "filled" by an $n+1$ dimensional object — i.e, the holes.

If we have a chain complex that looks like

$$\cdots \to C_n(X) \xrightarrow{d_n} C_{n-1}(X) \xrightarrow{d_{n-1}} C_{n-2}(X) \xrightarrow{d_{n-2}} \cdots$$

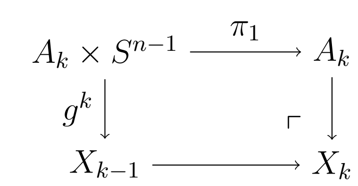
We can dualize into a *cochain complex* by fixing a group G, and then applying the Hom functor $\mathrm{Hom}(-, G)$ to the entire sequence, to obtain

$$\cdots \leftarrow \mathrm{Hom}(C_n(X), G) \xleftarrow{\partial^n} \mathrm{Hom}(C_{n-1}(X), G) \xleftarrow{\partial^{n-1}} \mathrm{Hom}(C_{n-2}(X), G) \xleftarrow{\partial_{n-2}} \cdots$$

The nth **cohomology group** is then the nth homology group of the cochain, which is denoted $H^n(X, G)$. We can view $H^n(X, G)$ as characterizing functions from cycles in X to elements of a group G. This grants significantly better algebraic properties than homology, like the existence of a product.

Background: CW complexes

A **CW complex** (also known as a cellular complex) is a topological space that can be built inductively by the attachment of open k-balls (known as cells) e^k_α by their boundary using continuous *gluing maps* $g^k_\alpha : \partial(e^k_\alpha) \to X_{k-1}$, where X_k is the result of gluing some number of n-cells to previous iteration X_{k-1}, known as the k-skeleton of the CW complex. We can define this formally as the following (homotopy) pushout:



We can then define a CW complex in HoTT as a type that is equivalent to a sequential colimit of a sequence of larger and larger n-skeletons. How to compute the cohomologies of CW complexes ("cellular cohomology") in HoTT is described in [1].

(Cellular) Group Cohomology

Since the cohomology of a chain complex is a purely algebraic construction, if we can associate a chain complex to a group in "the correct" way, we can compute its cohomology groups. There are 2 ways of doing this, in terms of the deloopings or derived functors. We focus on delooping. The *loop space* of a pointed space A, denoted ΩA, is the space/group of loops on the distinguished element $\star$ under composition. In the context of HoTT, we define $\Omega A = (\star =_A \star)$. The delooping of a Group G (the *classifying space* in the category **Top**) is a pointed connected space BG such that $\Omega BG = G$[4]. We define the cohomology groups of the group G as

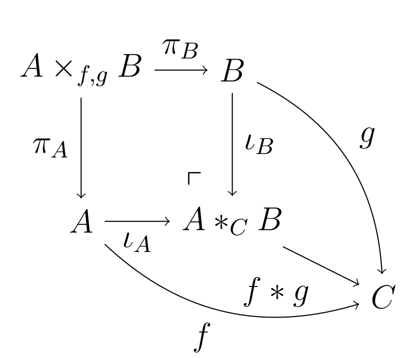
$$H^n(G, M) = H^n(BG, M)$$

There exists (pointed) models of BG[2] (which we use), but they have not been proven to be equivalent to a CW complex in general. We must prove this if to be able to use cellular cohomology to compute the cohomologies of groups. We provide a first step.

Theorem. For a finite group G, there exists a cellular delooping of G: a type CG such that CG is (homotopy) equivalent to BG and a CW complex.

Joins

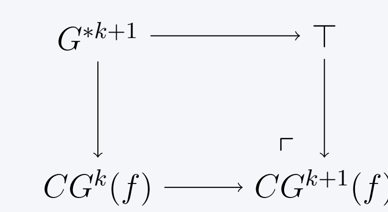
Definition. The **join** of types A and B, denoted $A * B$, is obtained by adding paths between every point $a : A$ and every point $b : B$, which can be defined as a pushout. Critically, as this is repeated, the types become more and more connected. We can in turn define the join of two maps $f : A \to C$ and $g : B \to C$ as the universal map out of the following pushout:



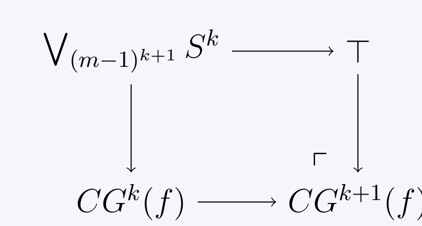
Proof Sketch

We break down the proof into 2 steps: constructing a space CG that is equivalent to BG, and then proving that it is in turn equivalent to a CW complex. Consider the map $f : \top \to BG$, mapping to the distinguished point of BG. We can show that the kernel of this map is equal to $\Omega BG = G$. This is equivalent to the data of a **fiber sequence**, which we can write $G \hookrightarrow \top \xrightarrow{f} BG$. We can take the repeated join of this map with itself $f^{*k} : \top *_{BG} \cdots *_{BG} \top \to BG$. We define CG^k to be the domain of f^{*k}. We prove that repeated join induces another fiber sequence: $G^{*k} \hookrightarrow CG^k \xrightarrow{f^{*k}} BG$. Taking the (sequential co)limit of this behavior, we get a final fiber sequence $G^{*\infty} \hookrightarrow CG \xrightarrow{f^{*\infty}} BG$ (defining $CG = CG^\infty$). But because spaces get more and more connected as you join them, we have $G^{*\infty}$ is contractible. We prove that in this situation, $f^{*\infty}$ is an equivalence.

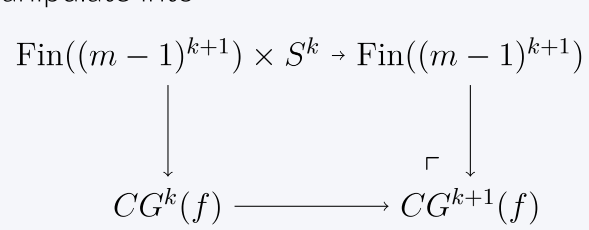
The proof of cellularity is a generalization of an argument from [3] and is significantly more involved. As such, we provide a higher level sketch. By the fact that G^{*k} is the kernel of f^{*k} (which can be defined as a pullback along f) and the definition of the join of maps as as pushout relative to this pullback, we get



Because G is finite (with cardinality, say, m), it can be written as the repeated wedge sum of S^0 (where the wedge sum ($\vee$) is the coproduct of 2 pointed types, but identifying their distinguished points). This allows us to apply a few important lemmas we prove (the most important of which is that join distributes over wedge sum) to get



which we can manipulate into



which is the k-skeleton of a CW complex.

References

[1] Ulrik Buchholtz and Kuen-Bang Hou Favonia. Cellular cohomology in homotopy type theory. In *Proceedings of the 33rd annual ACM/IEEE symposium on logic in computer science*, pages 521–529, 2018.

[2] Daniel R Licata and Eric Finster. Eilenberg-MacLane spaces in homotopy type theory. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–9, 2014.

[3] Samuel Mimram and Émile Oleon. Delooping cyclic groups with lens spaces in homotopy type theory. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–15, 2024.

[4] nLab authors. delooping. `https://ncatlab.org/nlab/show/delooping`, August 2025. Revision 40.