Computing Cohomology Rings in Cubical Agda

Thomas Lamiaux
thomas.lamiaux@ens-paris-saclay.fr
Ens Paris-Saclay, Universite Paris-Saclay

Axel Ljungström
axel.ljungstrom@math.su.se
Stockholm University

Anders Mörtberg
anders.mortberg@math.su.se
Stockholm University

1 Introduction

1.1 Motivation

Out of the many different branches of algebraic topology we may study in HoTT, cohomology theory has the advantage of having a particularly simple and practical definition. In classical algebraic topology, it also constitutes a powerful tool for distinguishing spaces and is often easier to handle than, for instance, the theory of homotopy groups. For those reasons, it has been thoroughly investigated over the last years, starting with the IAS year [7] and culminating in developments and proofs of of e.g. the Eilenberg-Steenrod axioms [4], the Serre spectral sequence [9], cellular cohomology [3], and $\pi_4(\mathbb{S}^3) \cong \mathbb{Z}/2\mathbb{Z}$ [1].

More recently, formalizations and computations of several $\mathbb{Z}$ cohomology groups were carried out in Cubical Agda [2]. This talk aims to share our continuation of this work in [5]. We will explain how we have completed the first formalization of cohomology rings (with arbitrary ring coefficients) and computed different examples of these in Cubical Agda [10]. Doing so, we will illustrate how data structures influence the formalization by considering the cases of the direct sum and multivariate polynomials, essential objects in the formalization and computation of cohomology rings. Moreover, we will explain how these data structures can yield a data refinement problem — not in computer science but in mathematics — and how this can be handled using univalence and the Structure Identity Principle (SIP)[8, Section 9.8]. In addition, we will also discuss how the power of computation in constructive mathematics can already be partially used as aid in the computation of cup products.

1.2 Cohomology Rings

A guiding principle in algebraic topology is that spaces are best understood through the lens of algebraic invariants. In the case of cohomology, this is a sequence of abelian groups thought of as encoding the "holes" of the space in questions. The idea is that these cohomology groups often are easier to compute and compare than topological spaces directly. In HoTT, the definition of cohomology groups is very compact: given a type X and an abelian groups G, the *nth cohomology group of X with coefficients in G* is defined by:

$$H^n(X, G) := \| \, X \longrightarrow K(G, n) \, \|_0$$

where $K(G, n)$ denotes the n:th Eilenberg-MacLane space, which is defined using HITs following [6]. This definition simplifies further when $G = \mathbb{Z}$. In this case, for $n > 0$, the cohomology groups may be defined by:

$$H^n(X, \mathbb{Z}) := H^n(X) := \| \, X \longrightarrow \|\mathbb{S}^n\|_n \, \|_0$$

Using only the $\mathbb{Z}$ case, it is already possible to distinguish many spaces, such as the n-sphere $\mathbb{S}^n$, the real and complex projective plan, $\mathbb{R}P^2$ and $\mathbb{C}P^2$, or the Klein Bottle $\mathbb{K}^2$ [2]. However, not all spaces can be distinguished using cohomology groups alone. For instance, for all $n : \mathbb{N}$ and abelian groups G, we have a group isomorphism $H^n(\mathbb{C}P^2, G) \cong H^n(\mathbb{S}^2 \vee \mathbb{S}^4, G)$.

Fortunately, the situation is often salvageable. Cohomology groups, when taken with coefficients in a ring R, have additional structure when considered all together. Indeed, there is a graded multiplication on cohomology groups called the cup product $\smile : H^i(X, R) \to H^j(X, R) \to H^{i+j}(X, R)$. Using the cup product, it is possible to turn the direct sum of the groups into a ring (graded commutative when R is commutative) called the cohomology ring:

$$H^*(X, R) := \bigoplus_{n:\mathbb{N}} H^i(X, R)$$

The point is that this cohomology ring *is an additional invariant* and, indeed, spaces like $\mathbb{C}P^2$, and $\mathbb{S}^2 \bigvee \mathbb{S}^4$ have different cohomology rings and cannot be equivalent, even though all of their cohomology groups are isomorphic.

The main goal of the talk is to explain how to formalize them and how to compute those rings in the framework of synthetic mathematics and homotopy type theory.

2 Contribution

Our primary contribution is the formalization of cohomology rings and computation of several concrete examples. In the cases that we are interested in, cohomology rings can be expressed as a quotient of a multivariate polynomial ring, for instance:

$$
\begin{array}{lcccl}
H^*(\mathbb{C}P^2, \mathbb{Z}) & := & \bigoplus_{n:\mathbb{N}} H^n(\mathbb{C}P^2, \mathbb{Z}) & \cong & \mathbb{Z}[X]/(X^3) \\
H^*(\mathbb{S}^2 \bigvee \mathbb{S}^4, \mathbb{Z}) & := & \bigoplus_{n:\mathbb{N}} H^n(\mathbb{S}^2 \bigvee \mathbb{S}^4, \mathbb{Z}) & \cong & \mathbb{Z}[X,Y]/(X^2, XY, Y^2)
\end{array}
$$

To do this, we have

(1) formalized a notion of direct sum practical enough to be able to deal with graded operations and to define graded rings.

(2) formalized a practical notions of multivariate polynomials which are easy to map in and out of.

(3) formalized cohomology rings as an instance of (1), lifting the cup product to be defined over the whole direct sum rather than just single instances of cohomology groups.

We had to main obstacles to overcome concerning formalization. Concerning (1), we have to stay constructive as the entire framework is and we want to preserve the computational aspect of synthetic mathematics. Concerning (2), the Cubical Agda library does not use tactics. As we do not wish to change that, we can not rely on tactics to get us through bureaucratic details, which significantly complicates the work.

3 Summary of the Talk

We will first discuss why trying to directly adapt the classical definition of the direct sum is not suitable for our purposes. In the general case, the indexing set needs to have decidable equality for the direct sum to be well-behaved. Moreover, even when the index is $\mathbb{N}$, lifting a graded operation such as the cup product to the direct sum is not that easy. The intensional feature of our language and type dependency gets in the way and we quickly descend into transport hell.

We will see how, in order to circumvent these issues, we have used quotient inductive types (set truncated HITs) to define the direct sum. These allow us to better lift graded operations to get graded rings, and work for any indexing set — not just $\mathbb{N}$. Using this definition, we will also show how, using the SIP, it is possible to get around the very tedious proofs of the classical definition and how it yields a data refinement problem in mathematics. Then how we can derive from the quotient inductive definition a directly defined notion of multivariate polynomials.

Combining these notions, we will show how to calculate some ring isomorphisms. In particular, we will show how the well chosen data structures enables us to only have to deal with the mathematical problems of interest — algebraic topology — by avoiding much of the bureaucratic work that one often need to do when formalizing.

Lastly, we will discuss the computation of cup products, which is needed in the computation of cohomology rings. This should technically, in concrete cases, be entirely doable by normalization in Cubical Agda, but in practice it is still limited as terms tend to blow up in size during normalization. Nevertheless, we will discuss how it is already possible to use normalization to help us characterize the cup product in some cases.

References

[1] G. Brunerie. *On the homotopy groups of spheres in homotopy type theory.* PhD thesis, Université Nice Sophia Antipolis, 2016.

[2] G. Brunerie, A. Ljungström, and A. Mörtberg. Synthetic Integral Cohomology in Cubical Agda. In F. Manea and A. Simpson, editors, *30th EACSL Annual Conference on Computer Science Logic (CSL 2022)*, volume 216 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[3] U. Buchholtz and K.-B. Hou Favonia. Cellular Cohomology in Homotopy Type Theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 521–529, New York, NY, USA, 2018. Association for Computing Machinery.

[4] E. Cavallo. Synthetic Cohomology in Homotopy Type Theory. Master's thesis, Carnegie Mellon University, 2015.

[5] T. Lamiaux, A. Ljungström, and A. Mörtberg. Computing cohomology rings in cubical agda. In *Proceedings of the 12th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2023, page 239–252, New York, NY, USA, 2023. Association for Computing Machinery.

[6] D. R. Licata and E. 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)*, CSL-LICS '14, New York, NY, USA, 2014. Association for Computing Machinery.

[7] M. Shulman. Cohomology, 2013. post on the Homotopy Type Theory blog: `http://homotopytypetheory.org/2013/07/24/`.

[8] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* Self-published, Institute for Advanced Study, 2013.

[9] F. van Doorn. *On the Formalization of Higher Inductive Types and Synthetic Homotopy Theory.* PhD thesis, Carnegie Mellon University, May 2018.

[10] A. Vezzosi, A. Mörtberg, and A. Abel. Cubical agda: A dependently typed programming language with univalence and higher inductive types. *Journal of Functional Programming*, 31:e8, 2021.