Computational Synthetic Homotopy Theory

Anders Mörtberg

Homotopy Type Theory conference, May 25, 2023

Outline

1. Introduction: cubical methods in homotopy type theory
2. Proofs by computation in synthetic homotopy theory
3. Proofs by computation in synthetic cohomology theory
4. Computational challenges and future work

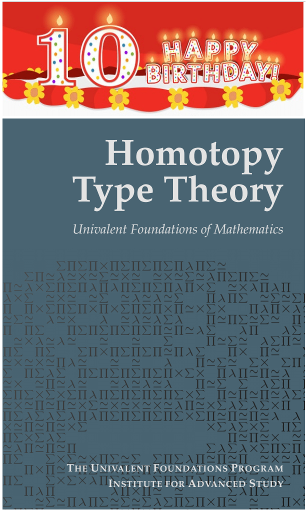
Outline

Synthetic algebraic topology

By representing spaces as higher inductive types we can develop algebraic topology *synthetically* using univalence

This is well-suited for computer formalization and leads to very compact and elegant proofs

But, as univalence is added axiomatically we cannot generally compute with these results in proof assistants...



Computational synthetic algebraic topology

Examples of computations that one might want to do in a formalization:

- Compute with functions

$$\phi : \pi_n(X) \to G \qquad\qquad \psi : H^n(X) \to G$$

  to e.g. compute winding numbers or prove that they are generated by particular elements

Computational synthetic algebraic topology

Examples of computations that one might want to do in a formalization:

- Compute with functions

  $$\phi : \pi_n(X) \longrightarrow G \qquad\qquad \psi : H^n(X) \longrightarrow G$$

  to e.g. compute winding numbers or prove that they are generated by particular elements

- Compute with group/ring operations on $\pi_n(X)$, $H^n(X)$, $H^*(X)$, ..., to distinguish spaces/types, e.g. $\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1 \neq \mathbb{T}$

Computational synthetic algebraic topology

Examples of computations that one might want to do in a formalization:

- Compute with functions

$$\phi : \pi_n(X) \to G \qquad\qquad \psi : H^n(X) \to G$$

  to e.g. compute winding numbers or prove that they are generated by particular elements

- Compute with group/ring operations on $\pi_n(X)$, $H^n(X)$, $H^*(X)$, ..., to distinguish spaces/types, e.g. $\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1 \neq \mathbb{T}$

- Characterize $\pi_n(X)$, $H^n(X)$, $H^*(X)$ by computation, e.g. the Brunerie number $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/\beta\mathbb{Z}$

Computational synthetic algebraic topology

Examples of computations that one might want to do in a formalization:

- Compute with functions

$$\phi : \pi_n(X) \to G \qquad\qquad \psi : H^n(X) \to G$$

  to e.g. compute winding numbers or prove that they are generated by particular elements

- Compute with group/ring operations on $\pi_n(X)$, $H^n(X)$, $H^*(X)$, ..., to distinguish spaces/types, e.g. $\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1 \neq \mathbb{T}$

- Characterize $\pi_n(X)$, $H^n(X)$, $H^*(X)$ by computation, e.g. the Brunerie number $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/\beta\mathbb{Z}$

- Prove tedious small proof steps by `refl` to shorten formal proofs

The Cubical paradigm in homotopy type theory

Theorem (Bezem-Coquand-Huber, 2013)

Univalent Type Theory has a constructive model in substructural Kan cubical sets ("BCH model").

This led to development of a variety of structural cubical set models and cubical type theories:

Theorem (Cohen-Coquand-Huber-M., 2015)

Univalent Type Theory has a constructive model in De Morgan Kan cubical sets ("CCHM model").

In cubical type theory we have a **univalence theorem** with computational content:

$$\mathsf{ua} : (A\ B : \mathcal{U}) \to (\mathsf{Path}_{\mathcal{U}}\ A\ B) \simeq (A \simeq B)$$

The many cubical models and type theories

	Structural	$\mathbb{I}$ operations	Kan operations	Diag. cofib.
BCH			$0 \to r, 1 \to r$	
CCHM	✓	$\wedge, \vee, \neg$ (DM alg.)	$0 \to 1$	
"Dedekind"	✓	$\wedge, \vee$ (dist. lattice)	$0 \to 1, 1 \to 0$	
Orton-Pitts	✓	$\wedge, \vee$ (conn. alg.)	$0 \to 1, 1 \to 0$	
Cartesian (A, AFH, ABCFHL)[1]	✓		$r \to s$	✓
Equivariant (ACCRS)[2]	✓		$\vec{r} \to \vec{s}$	✓
Cavallo-Sattler	✓	$\vee$ / $\wedge$	$0 \to r, 1 \to r$	✓

The last two are known to be equivalent to spaces

Comparison and unification of the Kan operations: *Unifying Cubical Models of Univalent Type Theory* (Cavallo-M.-Swan, CSL'20)

[1] Awodey, Angiuli-Favonia-Harper, Angiuli-Brunerie-Coquand-Favonia-Harper-Licata

[2] Awodey-Cavallo-Coquand-Riehl-Sattler

Cubical proof assistants

These different cubical type theories satisfy good metatheoretic properties: canonicity (Huber, AFH), normalization and decidable typechecking (Sterling-Angiuli)

There are also many cubical proof assistants: `cubical`, `cubicaltt`, `yacctt`, **RedPRL**, **redtt**, **cooltt**, `Cubical Agda`...

In `Cubical Agda` we have explored how to do synthetic proofs *computationally*, in particular by computing a Brunerie number

All results have been formalized in: `https://github.com/agda/cubical/`

Outline

Synthetic homotopy theory

The *n*th *homotopy group* of a pointed type X can be defined as:[3]

$$\pi_n(X) = \|\mathbb{S}^n \to_\star X\|_0$$

[3] Equivalently: $\pi_n(X) = \|\Omega^n(X)\|_0$

Synthetic homotopy theory

The *n*th *homotopy group* of a pointed type X can be defined as:[3]

$$\pi_n(X) = \|\mathbb{S}^n \to_\star X\|_0$$

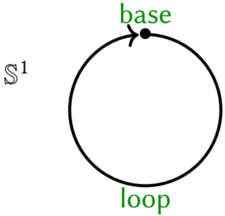
These groups constitute a topological invariant, making them a powerful tool for establishing whether two given spaces are homotopy equivalent

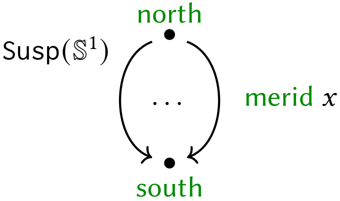
- $\pi_0(X)$ characterizes the connected components of X
- $\pi_1(X)$ characterizes equivalence classes the loops in X up to homotopy
- $\pi_n(X)$, for $n > 1$, characterizes of n-dimensional loops up to homotopy

[3] Equivalently: $\pi_n(X) = \|\Omega^n(X)\|_0$

Synthetic homotopy theory

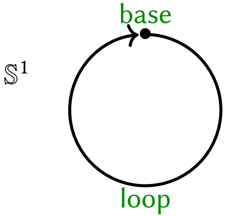
Using univalence we can prove properties of $\pi_n(X)$ for concrete spaces X represented using HITs

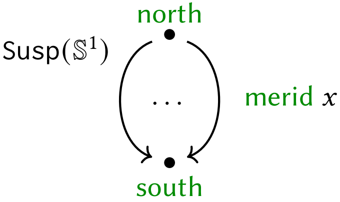




Synthetic homotopy theory

Using univalence we can prove properties of $\pi_n(X)$ for concrete spaces X represented using HITs





Example: $\pi_1(\mathbb{S}^1) \simeq \mathbb{Z}$ can be proved using the encode-decode method (Licata-Shulman '13)

Many other standard results allowing us to characterize homotopy groups of spheres can be found in the HoTT book: the Hopf fibration, Freudenthal suspension theorem, long exact sequence of homotopy groups, connectivity of spheres, ...

Homotopy groups of spheres synthetically

However, for many spaces, these groups tend to become increasingly esoteric and difficult to compute for large n

	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}
$\mathbb{S}^1$	$\mathbb{Z}$	0	0	0	0	0	0	0	0	0
$\mathbb{S}^2$	0	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_{15}$
$\mathbb{S}^3$	0	0	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_3$	$\mathbb{Z}_{15}$
$\mathbb{S}^4$	0	0	0	$\mathbb{Z}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z} \times \mathbb{Z}_{12}$	$\mathbb{Z}_2 \times \mathbb{Z}_2$	$\mathbb{Z}_2 \times \mathbb{Z}_2$	$\mathbb{Z}_{24} \times \mathbb{Z}_3$

The fourth homotopy group of the 3-sphere in HoTT

Guillaume Brunerie's PhD thesis contains a synthetic proof in Book HoTT of:

Theorem (Brunerie, 2016)

The fourth homotopy group of the 3-sphere is $\mathbb{Z}/2\mathbb{Z}$, that is, $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/2\mathbb{Z}$

The fourth homotopy group of the 3-sphere in HoTT

Guillaume Brunerie's PhD thesis contains a synthetic proof in Book HoTT of:

Theorem (Brunerie, 2016)

The fourth homotopy group of the 3-sphere is $\mathbb{Z}/2\mathbb{Z}$, that is, $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/2\mathbb{Z}$

The proof is one of the most impressive pieces of synthetic homotopy theory to date and uses lots of advanced classical machinery developed synthetically in HoTT:

The fourth homotopy group of the 3-sphere in HoTT

Guillaume Brunerie's PhD thesis contains a synthetic proof in Book HoTT of:

Theorem (Brunerie, 2016)

The fourth homotopy group of the 3-sphere is $\mathbb{Z}/2\mathbb{Z}$*, that is,* $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/2\mathbb{Z}$

The proof is one of the most impressive pieces of synthetic homotopy theory to date and uses lots of advanced classical machinery developed synthetically in HoTT: *symmetric monoidal structure of smash products, (integral) cohomology rings, the Mayer-Vietoris and Gysin sequences, Hopf invariant homomorphism, Whitehead products, the iterated Hopf construction, Blakers-Massey, ...*

The fourth homotopy group of the 3-sphere in HoTT

Guillaume Brunerie's PhD thesis contains a synthetic proof in Book HoTT of:

Theorem (Brunerie, 2016)

The fourth homotopy group of the 3-sphere is $\mathbb{Z}/2\mathbb{Z}$*, that is,* $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/2\mathbb{Z}$

The proof is one of the most impressive pieces of synthetic homotopy theory to date and uses lots of advanced classical machinery developed synthetically in HoTT: *symmetric monoidal structure of smash products, (integral) cohomology rings, the Mayer-Vietoris and Gysin sequences, Hopf invariant homomorphism, Whitehead products, the iterated Hopf construction, Blakers-Massey, ...*

Furthermore, the proof is fully constructive!

The Brunerie number

The theorem can be phrased as:

$$\textit{there exists a number } \beta : \mathbb{Z} \textit{ such that } \pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/\beta\mathbb{Z}$$

[4] Using only 1-HITs

The Brunerie number

The theorem can be phrased as:

there exists a number $\beta : \mathbb{Z}$ such that $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/\beta\mathbb{Z}$

Appendix B of Brunerie's thesis contains a complete and concise definition of β as the image of 1 under a sequence of 12 maps:[4]

$$\mathbb{Z} \longrightarrow \Omega(\mathbb{S}^1) \longrightarrow \Omega^2(\mathbb{S}^2) \longrightarrow \Omega^3(\mathbb{S}^3)$$

$$\Omega^3(\mathbb{S}^1 * \mathbb{S}^1) \xrightarrow{\Omega^3\alpha} \Omega^3(\mathbb{S}^2) \xrightarrow{h} \Omega^3(\mathbb{S}^1 * \mathbb{S}^1) \longrightarrow \Omega^3(\mathbb{S}^3)$$

$$\Omega^2\|\mathbb{S}^2\|_2 \longrightarrow \Omega\|\Omega(\mathbb{S}^2)\|_1 \longrightarrow \|\Omega^2(\mathbb{S}^2)\|_0 \longrightarrow \Omega(\mathbb{S}^1) \longrightarrow \mathbb{Z}$$

[4] Using only 1-HITs

The Brunerie number

On page 85 Brunerie says (for $n := |\beta|$):

> *This result is quite remarkable in that even though it is a constructive proof, it is not at all obvious how to actually compute this n. At the time of writing, we still haven't managed to extract its value from its definition. A complete and concise definition of this number n is presented in appendix B, for the benefit of someone wanting to implement it in a prospective proof assistant. In the rest of this thesis, we give a mathematical proof in homotopy type theory that n = 2.*

As the above cubical systems satisfy canonicity it should *in principle* be possible to use them to compute the Brunerie number...

The Brunerie number

On page 85 Brunerie says (for $n := |\beta|$):

> *This result is quite remarkable in that even though it is a constructive proof, it is not at all obvious how to actually compute this n. At the time of writing, we still haven't managed to extract its value from its definition. A complete and concise definition of this number n is presented in appendix B, for the benefit of someone wanting to implement it in a prospective proof assistant. In the rest of this thesis, we give a mathematical proof in homotopy type theory that $n = 2$.*

As the above cubical systems satisfy canonicity it should *in principle* be possible to use them to compute the Brunerie number... But this turned out to be **a lot** harder than expected!

Computing the Brunerie number, a (probably incomplete) history

- 2013: Guillaume presents informal definition of the Brunerie number at an IAS seminar
- December 2014: Guillaume visits Chalmers and tries to compute it with Thierry Coquand and Simon Huber using `cubical` (based on BCH model)
- Spring 2015: I join forces with them and spend a lot of time trying to benchmark and optimize the Haskell implementation of `cubical`
- 2016: Guillaume finishes thesis with definition in Appendix B (based on `cubical` code)

Computing the Brunerie number, a (probably incomplete) history

- 2013: Guillaume presents informal definition of the Brunerie number at an IAS seminar
- December 2014: Guillaume visits Chalmers and tries to compute it with Thierry Coquand and Simon Huber using `cubical` (based on BCH model)
- Spring 2015: I join forces with them and spend a lot of time trying to benchmark and optimize the Haskell implementation of `cubical`
- 2016: Guillaume finishes thesis with definition in Appendix B (based on `cubical` code)
- Spring/summer 2017: I port the proof to `cubicaltt` (based on CCHM), but computation runs out of memory (on Inria server with 64GB RAM)
- June 2017: another attempt in `cubicaltt` with the MRC group in Snowbird (Choudhury, Gustafson, Licata, Orton, Sterling). Optimizes the definition of the number, without luck
- Late 2017: I visit Guillaume repeatedly at the IAS and simplify the definition a lot, computation goes slightly further but still runs out of memory

Computing the Brunerie number, a (probably incomplete) history

- June 2018: Favonia tries running the `cubicaltt` computation on a super computer with 1TB of ram, computation terminated after $\sim$ 90 hours
- Summer 2018: Dagstuhl meeting where the cubical group (Sterling, Angiuli, Favonia, Licata, Huber, Orton, Brunerie, and I) found various new optimizations to cubical evaluation ("Dagstuhl lemma"), did not help with computation

Computing the Brunerie number, a (probably incomplete) history

- June 2018: Favonia tries running the `cubicaltt` computation on a super computer with 1TB of ram, computation terminated after $\sim$ 90 hours
- Summer 2018: Dagstuhl meeting where the cubical group (Sterling, Angiuli, Favonia, Licata, Huber, Orton, Brunerie, and I) found various new optimizations to cubical evaluation ("Dagstuhl lemma"), did not help with computation
- 2019: Evan Cavallo ports the definition to `Cubical Agda`, still running out of memory despite more optimizations (including `Cubical Agda` "ghcomp" trick of Vezzosi, inspired by AFH)

Computing the Brunerie number, a (probably incomplete) history

- June 2018: Favonia tries running the `cubicaltt` computation on a super computer with 1TB of ram, computation terminated after $\sim$ 90 hours
- Summer 2018: Dagstuhl meeting where the cubical group (Sterling, Angiuli, Favonia, Licata, Huber, Orton, Brunerie, and I) found various new optimizations to cubical evaluation ("Dagstuhl lemma"), did not help with computation
- 2019: Evan Cavallo ports the definition to `Cubical Agda`, still running out of memory despite more optimizations (including `Cubical Agda` "ghcomp" trick of Vezzosi, inspired by AFH)
- 2020-2021: No progress. I was convinced that the only way to make progress was to improve closed term evaluation for cubical type theories...

Computing the Brunerie number, a (probably incomplete) history

- June 2018: Favonia tries running the `cubicaltt` computation on a super computer with 1TB of ram, computation terminated after $\sim$ 90 hours
- Summer 2018: Dagstuhl meeting where the cubical group (Sterling, Angiuli, Favonia, Licata, Huber, Orton, Brunerie, and I) found various new optimizations to cubical evaluation ("Dagstuhl lemma"), did not help with computation
- 2019: Evan Cavallo ports the definition to `Cubical Agda`, still running out of memory despite more optimizations (including `Cubical Agda` "ghcomp" trick of Vezzosi, inspired by AFH)
- 2020-2021: No progress. I was convinced that the only way to make progress was to improve closed term evaluation for cubical type theories...
- 2022: Breakthrough with Axel Ljungström...

Computing the Brunerie number, a (probably incomplete) history

- June 2018: Favonia tries running the `cubicaltt` computation on a super computer with 1TB of ram, computation terminated after $\sim$ 90 hours
- Summer 2018: Dagstuhl meeting where the cubical group (Sterling, Angiuli, Favonia, Licata, Huber, Orton, Brunerie, and I) found various new optimizations to cubical evaluation ("Dagstuhl lemma"), did not help with computation
- 2019: Evan Cavallo ports the definition to `Cubical Agda`, still running out of memory despite more optimizations (including `Cubical Agda` "ghcomp" trick of Vezzosi, inspired by AFH)
- 2020-2021: No progress. I was convinced that the only way to make progress was to improve closed term evaluation for cubical type theories...
- 2022: Breakthrough with Axel Ljungström... A variation on the Brunerie number normalizes to -2 in just a few seconds in `Cubical Agda`!

Computing the Brunerie number, a (probably incomplete) history

- June 2018: Favonia tries running the `cubicaltt` computation on a super computer with 1TB of ram, computation terminated after $\sim$ 90 hours
- Summer 2018: Dagstuhl meeting where the cubical group (Sterling, Angiuli, Favonia, Licata, Huber, Orton, Brunerie, and I) found various new optimizations to cubical evaluation ("Dagstuhl lemma"), did not help with computation
- 2019: Evan Cavallo ports the definition to `Cubical Agda`, still running out of memory despite more optimizations (including `Cubical Agda` "ghcomp" trick of Vezzosi, inspired by AFH)
- 2020-2021: No progress. I was convinced that the only way to make progress was to improve closed term evaluation for cubical type theories...
- 2022: Breakthrough with Axel Ljungström... A variation on the Brunerie number normalizes to -2 in just a few seconds in `Cubical Agda`!
- 2023: slightly more complex Brunerie number computes in Kovács' `cctt` (which takes closed term evaluation seriously!)
- 2023: Tom Jack's new number computes to 2 in `cubicaltt` and `cctt`
- ...

Formalizing $\pi_4(\mathbb{S}^3) \cong \mathbb{Z}/2\mathbb{Z}$ and Computing a Brunerie Number in Cubical Agda

We have a write-up: `https://arxiv.org/abs/2302.00151`

This was recently accepted to LICS'23 and the paper contains 3 fully formalized proofs:

1. Streamlined and complete version of Brunerie's original proof
2. Axel's new proof
3. A computational proof relying on normalization of the variation on β

Proofs 1 and 2 work in Book HoTT, proof 3 relies on cubical normalization

Contents

Brunerie's theorem: part 1 (chapters 1–3)

In the first half of the thesis (chapters 1–3) Guillaume constructs a map $g : \mathbb{S}^3 \to \mathbb{S}^2$

g is defined as the composition of a sequence of (pointed) maps $\mathbb{S}^3 \to \mathbb{S}^1 * \mathbb{S}^1 \to \mathbb{S}^2 \vee \mathbb{S}^2 \to \mathbb{S}^2$

Brunerie's theorem: part 1 (chapters 1–3)

In the first half of the thesis (chapters 1–3) Guillaume constructs a map $g : \mathbb{S}^3 \longrightarrow \mathbb{S}^2$

g is defined as the composition of a sequence of (pointed) maps $\mathbb{S}^3 \longrightarrow \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^2 \vee \mathbb{S}^2 \longrightarrow \mathbb{S}^2$

Let $e : \pi_3(\mathbb{S}^2) \simeq \mathbb{Z}$ and define $\beta := e(| g |_0)$, the first main theorem is then that:

Theorem (Brunerie, Corollary 3.4.5)

We have $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/\beta\mathbb{Z}$

Brunerie's proof: part 2 (chapters 4–6)

- Symmetric monoidal structure of smash products
  $\Longrightarrow$ The graded ring structure of the cup product $\smile : H^i(X) \times H^j(X) \to H^{i+j}(X)$
- The *Mayer-Vietoris* sequence
- The *Gysin Sequence*
- The *Hopf Invariant* homomorphism
- The *Iterated Hopf Construction*

Brunerie's proof: part 2 (chapters 4–6)

- **Symmetric monoidal structure of smash products**

  $\Longrightarrow$ The graded ring structure of the cup product $\smile : H^i(X) \times H^j(X) \to H^{i+j}(X)$
- The *Mayer-Vietoris* sequence
- The *Gysin Sequence*
- The *Hopf Invariant* homomorphism
- The *Iterated Hopf Construction*

Brunerie's proof: part 2 (chapters 4–6)

- $(A \wedge B \to_\star C) \simeq (A \to_\star (B \to_\star C))$ **workaround** (Brunerie-Ljungström-M. CSL'22)
  $\Longrightarrow$ The graded ring structure of the cup product
  $\smile : H^i(X) \times H^j(X) \to H^{i+j}(X)$
- The *Mayer-Vietoris* sequence
- The *Gysin Sequence*
- The *Hopf Invariant* homomorphism
- The *Iterated Hopf Construction*

Brunerie's proof: part 2 (chapters 4–6)

- **Symmetric monoidal structure of smash products** (Ljungström HoTT'23)
  $\Longrightarrow$ The graded ring structure of the cup product $\smile : H^i(X) \times H^j(X) \to H^{i+j}(X)$
- The *Mayer-Vietoris* sequence
- The *Gysin Sequence*
- The *Hopf Invariant* homomorphism
- The *Iterated Hopf Construction*

New proof

Having finished the formalization of chapters 4–6 Axel realized that one can actually simplify the proof a lot and completely avoid the second half of Brunerie's thesis

The new proof is very elementary – doesn't use any complicated theory!

New proof

Having finished the formalization of chapters 4–6 Axel realized that one can actually simplify the proof a lot and completely avoid the second half of Brunerie's thesis

The new proof is very elementary – doesn't use any complicated theory!

Idea: trace the maps by hand using clever tricks and choices

Sketch of new proof

Recall that $\beta := e(|g|_0)$ for $e : \pi_3(\mathbb{S}^2) \simeq \mathbb{Z}$ and $g : \mathbb{S}^3 \to \mathbb{S}^2$. The goal is to show that $|\beta| = 2$

Sketch of new proof

Recall that $\beta := e(|g|_0)$ for $e : \pi_3(\mathbb{S}^2) \simeq \mathbb{Z}$ and $g : \mathbb{S}^3 \longrightarrow \mathbb{S}^2$. The goal is to show that $|\beta| = 2$

In fact, g is defined as the precomposition of a not very complicated map $\mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^2$ with the somewhat complicated equivalence $f : \mathbb{S}^3 \simeq \mathbb{S}^1 * \mathbb{S}^1$

Sketch of new proof

Recall that $\beta := e(|g|_0)$ for $e : \pi_3(\mathbb{S}^2) \simeq \mathbb{Z}$ and $g : \mathbb{S}^3 \longrightarrow \mathbb{S}^2$. The goal is to show that $|\beta| = 2$

In fact, g is defined as the precomposition of a not very complicated map $\mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^2$ with the somewhat complicated equivalence $f : \mathbb{S}^3 \simeq \mathbb{S}^1 * \mathbb{S}^1$

One of Axel's tricks in the proof is to define $\pi_3^*(A) := ||\mathbb{S}^1 * \mathbb{S}^1 \longrightarrow_\star A||_0$ and work with it instead so that f can be avoided

Sketch of new proof

We can now decompose $e : \pi_3(\mathbb{S}^2) \simeq \mathbb{Z}$ as:

$$\pi_3(\mathbb{S}^2) \overset{e_1}{\simeq} \pi_3^*(\mathbb{S}^2) \overset{e_2}{\simeq} \pi_3^*(\mathbb{S}^1 * \mathbb{S}^1) \overset{e_3}{\simeq} \pi_3^*(\mathbb{S}^3) \overset{e_4}{\simeq} \mathbb{Z}$$

Sketch of new proof

We can now decompose $e : \pi_3(\mathbb{S}^2) \simeq \mathbb{Z}$ as:

$$\pi_3(\mathbb{S}^2) \overset{e_1}{\simeq} \pi_3^*(\mathbb{S}^2) \overset{e_2}{\simeq} \pi_3^*(\mathbb{S}^1 * \mathbb{S}^1) \overset{e_3}{\simeq} \pi_3^*(\mathbb{S}^3) \overset{e_4}{\simeq} \mathbb{Z}$$

We can also give explicit definitions of

$$\eta_1 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^2 \qquad \eta_2 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^1 * \mathbb{S}^1 \qquad \eta_3 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^3$$

such that

$$e_1(|g|_0) = |\eta_1|_0 \qquad e_2(|\eta_1|_0) = |\eta_2|_0 \qquad e_3(|\eta_2|_0) = |\eta_3|_0 \qquad e_4(|\eta_3|_0) = -2$$

Sketch of new proof

We can now decompose $e : \pi_3(\mathbb{S}^2) \simeq \mathbb{Z}$ as:

$$\pi_3(\mathbb{S}^2) \overset{e_1}{\simeq} \pi_3^*(\mathbb{S}^2) \overset{e_2}{\simeq} \pi_3^*(\mathbb{S}^1 * \mathbb{S}^1) \overset{e_3}{\simeq} \pi_3^*(\mathbb{S}^3) \overset{e_4}{\simeq} \mathbb{Z}$$

We can also give explicit definitions of

$$\eta_1 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^2 \qquad \eta_2 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^1 * \mathbb{S}^1 \qquad \eta_3 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^3$$

such that

$$e_1(|g|_0) = |\eta_1|_0 \qquad e_2(|\eta_1|_0) = |\eta_2|_0 \qquad e_3(|\eta_2|_0) = |\eta_3|_0 \qquad e_4(|\eta_3|_0) = -2$$

The first 3 equalities are not definitional and requires some clever choices, but (surprisingly) a variation of the last one holds by `refl` in `Cubical Agda`!

New Brunerie numbers

$\eta_1 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^2$
η_1 (inl x) = north
η_1 (inr y) = north
η_1 (push (x , y) i) = ($\sigma\ y \cdot \sigma\ x$) i

$\eta_2 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^1 * \mathbb{S}^1$
η_2 (inl x) = inr (- x)
η_2 (inr y) = inr y
η_2 (push (x , y) i) =
(push (y - x , - x) $^{-1} \cdot$ push (y - x , y)) i

$\eta_3 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^3$
η_3 (inl x) = north
η_3 (inr y) = north
η_3 (push (x , y) i) =
$(\sigma\ (x \smile_1 y)^{-1} \cdot \sigma\ (x \smile_1 y)^{-1})\ i$

η_3' $: \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^3$
η_3' (inl x) = north
η_3' (inr y) = north
η_3' (push (x , y) i) =
$(\sigma\ (x \smile_1 y) \cdot \sigma\ (x \smile_1 y))\ i$

New Brunerie numbers

$\eta_1 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^2$
η_1 (inl x) = north
η_1 (inr y) = north
η_1 (push (x , y) i) = (σ y • σ x) i

$\eta_2 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^1 * \mathbb{S}^1$
η_2 (inl x) = inr (- x)
η_2 (inr y) = inr y
η_2 (push (x , y) i) =
  (push (y - x , - x) $^{-1}$ • push (y - x , y)) i

$\eta_3 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^3$
η_3 (inl x) = north
η_3 (inr y) = north
η_3 (push (x , y) i) =
  (σ (x $\smile_1$ y) $^{-1}$ • σ (x $\smile_1$ y) $^{-1}$) i

η_3' : $\mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^3$
η_3' (inl x) = north
η_3' (inr y) = north
η_3' (push (x , y) i) =
  (σ (x $\smile_1$ y) • σ (x $\smile_1$ y)) i

This gives a sequence of new Brunerie numbers:

$$\beta_1 := e_4(e_3(e_2(|\eta_1|_0))) \qquad \beta_2 := e_4(e_3(|\eta_2|_0)) \qquad \beta_3 := e_4(|\eta_3|_0) \qquad \beta' := e_4(|\eta_3'|_0)$$

File Edit Options Buffers Tools Agda Help

```
-- We also have a much more direct proof in Cubical.Homotopy.Group.Pi4S3.DirectProof,
-- not relying on any of the more advanced constructions in chapters
-- 4-6 in Brunerie's thesis (but still using chapters 1-3 for the
-- construction). For details see the header of that file.

π₄S³≃ℤ/2ℤ-direct : GroupEquiv (π 4 𝕊³) (ℤGroup/ 2)
π₄S³≃ℤ/2ℤ-direct = DirectProof.BrunerieGroupEquiv

-- This direct proof allows us to define a much simplified version of
-- the Brunerie number:
β' : ℤ
β' = fst DirectProof.computer η₃'

-- This number computes definitionally to -2 in a few seconds!
β'≡-2 : β' ≡ -2
β'≡-2 = refl

-- Combining all of this gives us the desired equivalence of groups by
-- computation as conjectured in Brunerie's thesis:
π₄S³≃ℤ/2ℤ-computation : GroupEquiv (π 4 𝕊³) (ℤGroup/ 2)
π₄S³≃ℤ/2ℤ-computation = DirectProof.BrunerieGroupEquiv''
```

U:--- Summary.agda Bot (112,0) Git:inducedstruct (Agda:Checked +2)

U:%*- *All Done* All (1,0) (AgdaInfo)

The three formalized proofs

We have three fully formalized synthetic proofs that $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/2\mathbb{Z}$:

1. Streamlined and complete proof following Guillaume's thesis (17000 LOC + 8000 from library)
2. Axel's new direct elementary proof which avoids part 2 of the thesis completely (600 LOC)
3. The new computational proof by normalizing β' (400 LOC)

Common part to all proofs (Brunerie Chapters 1-3): 9000 LOC

The first two proofs are expressable in Book HoTT, while the third crucially relies on normalization of terms involving univalence and HITs (so expressable in cubical systems, and maybe H.O.T.T.)

Outline

Synthetic cohomology theory

In HoTT we can define cohomology as:[5]

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

In *Synthetic Integral Cohomology in Cubical Agda* (Brunerie-Ljungström-M., CSL'22) we equip $H^n(X,\mathbb{Z})$ with a very concrete group structure that computes quite well

[5]Buchholtz, Brunerie, Cavallo, Favonia, Finster, Licata, Shulman, van Doorn, ...

Synthetic cohomology theory

In HoTT we can define cohomology as:[5]

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

In *Synthetic Integral Cohomology in Cubical Agda* (Brunerie-Ljungström-M., CSL'22) we equip $H^n(X, \mathbb{Z})$ with a very concrete group structure that computes quite well

We also compute cohomology groups for many classical spaces: spheres, torus, Klein bottle, wedge sums, real and complex projective planes

Many of these proofs are direct by analyzing function spaces, but some require more elaborate classical techniques (Eilenberg-Steenrod axioms, Mayer-Vietoris sequence)

[5] Buchholtz, Brunerie, Cavallo, Favonia, Finster, Licata, Shulman, van Doorn, ...

Side remark: relationship to homotopy groups of spheres

Integral cohomology gives a nice map $\pi_n(\mathbb{S}^n) \to \mathbb{Z}$. Note the similarity in:

$$\pi_n(\mathbb{S}^n) = \|\mathbb{S}^n \to_\star \mathbb{S}^n\|_0$$

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

This is used in the new Brunerie number computation: it is quite straightforward to prove that $H^3(\mathbb{S}^3, \mathbb{Z}) \simeq \mathbb{Z}$ and the maps have better computational behavior than the ones in $\pi_3(\mathbb{S}^3) \simeq \mathbb{Z}$ obtained by iterated Freudenthal suspension theorem

Computations in proofs of cohomology groups

Proofs by computation also pop up in synthetic cohomology theory:

Computations in proofs of cohomology groups

Proofs by computation also pop up in synthetic cohomology theory:

- Base cases when verifying the group laws for $H^n(X, \mathbb{Z})$ involve path algebra in loop spaces over the spheres which can typically be reduced to integer computations

Computations in proofs of cohomology groups

Proofs by computation also pop up in synthetic cohomology theory:

- Base cases when verifying the group laws for $H^n(X, \mathbb{Z})$ involve path algebra in loop spaces over the spheres which can typically be reduced to integer computations
- When showing that $H^n(X, G)$ or $\pi_n(X)$ is generated by a particular element e we can use that the group is equivalent to some nice group G (e.g. $\mathbb{Z}$) and check that e is mapped to a generator of G (e.g. ± 1))

Computations in proofs of cohomology groups

Proofs by computation also pop up in synthetic cohomology theory:

- Base cases when verifying the group laws for $H^n(X, \mathbb{Z})$ involve path algebra in loop spaces over the spheres which can typically be reduced to integer computations
- When showing that $H^n(X, G)$ or $\pi_n(X)$ is generated by a particular element e we can use that the group is equivalent to some nice group G (e.g. $\mathbb{Z}$) and check that e is mapped to a generator of G (e.g. ± 1))
- Various computations involving the group operations

Some of these are fast, some are slow, and some do not terminate in a reasonable amount of time (minutes on a normal laptop)

Cohomology benchmarks

For every equivalence $\phi : H^n(X, \mathbb{Z}) \simeq G$ that we have formalized, two benchmarks have been run in `Cubical Agda`:

- **Test 1**: can $\phi\,(\phi^{-1}(g)) \equiv g$ be proved by refl for different values of $g : G$?
- **Test 2** can $\phi\,(\phi^{-1}(g_1) +_H \phi^{-1}(g_2)) \equiv g_1 +_G g_2$ be proved by refl for $g_1, g_2 : G$?

Cohomology benchmarks

Type A	**Cohomology**	**Group** G	**Test 1**	**Test 2**
$\mathbb{S}^1$	H^1	$\mathbb{Z}$	✓	✓
$\mathbb{S}^2$	H^2	$\mathbb{Z}$	✓	✓
$\mathbb{S}^3$	H^3	$\mathbb{Z}$	✓	✗
$\mathbb{S}^4$	H^4	$\mathbb{Z}$	✗	✗
$\mathbb{T}^2$	H^1	$\mathbb{Z} \times \mathbb{Z}$	✓	✓
	H^2	$\mathbb{Z}$	✓	✓
$\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1$	H^1	$\mathbb{Z} \times \mathbb{Z}$	✓	✓
	H^2	$\mathbb{Z}$	✓	✓
$\mathbb{K}^2$	H^1	$\mathbb{Z}$	✓	✓
	H^2	$\mathbb{Z}/2\mathbb{Z}$	✗	✗
$\mathbb{R}P^2$	H^2	$\mathbb{Z}/2\mathbb{Z}$	✗	✗
$\mathbb{C}P^2$	H^2	$\mathbb{Z}$	✓	✓
	H^4	$\mathbb{Z}$	✗	✗

Cup product and cohomology ring

Cohomology allows us to distinguish many spaces, but it is sometimes a bit too coarse. We can equip cohomology groups also with a graded multiplication operations

$$\smile\ : H^n(X) \to H^m(X) \to H^{n+m}(X)$$

This can be organized into a graded commutative ring $H^*(X)$

Cup product and cohomology ring

Cohomology allows us to distinguish many spaces, but it is sometimes a bit too coarse. We can equip cohomology groups also with a graded multiplication operations

$$\smile : H^n(X) \to H^m(X) \to H^{n+m}(X)$$

This can be organized into a graded commutative ring $H^*(X)$

These rings are often equivalent to quotients of multivariate polynomial rings and we computed some of these in *Computing Cohomology Rings in Cubical Agda* (Lamiaux-Ljungström-M., CPP'23)

Cup product and cohomology ring

Cohomology allows us to distinguish many spaces, but it is sometimes a bit too coarse. We can equip cohomology groups also with a graded multiplication operations

$$\smile : H^n(X) \to H^m(X) \to H^{n+m}(X)$$

This can be organized into a graded commutative ring $H^*(X)$

These rings are often equivalent to quotients of multivariate polynomial rings and we computed some of these in *Computing Cohomology Rings in Cubical Agda* (Lamiaux-Ljungström-M., CPP'23)

Application: $\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1$ has the same cohomology groups as $\mathbb{T}^2$, but they are not equivalent as the cohomology rings differ

Computing with the cohomology ring

To distinguish $\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1$ and $\mathbb{T}^2$ we define a predicate $P : \mathsf{Type} \to \mathsf{Type}$:

$$P(A) :\equiv (x\ y : H^1(A)) \to x \smile y \equiv 0_h$$

Computing with the cohomology ring

To distinguish $\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1$ and $\mathbb{T}^2$ we define a predicate P : Type $\longrightarrow$ Type:

$$P(A) := (x\ y : H^1(A)) \rightarrow x \smile y \equiv 0_h$$

We have the isomorphisms:

$$f_1 : H^1(\mathbb{T}^2) \cong \mathbb{Z} \times \mathbb{Z}$$
$$f_2 : H^2(\mathbb{T}^2) \cong \mathbb{Z}$$

$$g_1 : H^1(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1) \cong \mathbb{Z} \times \mathbb{Z}$$
$$g_2 : H^2(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1) \cong \mathbb{Z}$$

Computing with the cohomology ring

To distinguish $\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1$ and $\mathbb{T}^2$ we define a predicate $P : \mathsf{Type} \to \mathsf{Type}$:

$$P(A) := (x\ y : H^1(A)) \to x \smile y \equiv 0_h$$

We have the isomorphisms:

$$f_1 : H^1(\mathbb{T}^2) \cong \mathbb{Z} \times \mathbb{Z} \qquad g_1 : H^1(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1) \cong \mathbb{Z} \times \mathbb{Z}$$
$$f_2 : H^2(\mathbb{T}^2) \cong \mathbb{Z} \qquad g_2 : H^2(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1) \cong \mathbb{Z}$$

We will now disprove $P(\mathbb{T}^2)$ and prove $P(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$, which establishes that they are not equivalent

Computing with the cohomology ring

To disprove $P(\mathbb{T}^2)$ we need $x, y : H^1(\mathbb{T}^2)$ such that $x \smile y \not\equiv 0_h$.

Computing with the cohomology ring

To disprove $P(\mathbb{T}^2)$ we need $x, y : H^1(\mathbb{T}^2)$ such that $x \smile y \not\equiv 0_h$. Let

$$x = f_1^{-1}(0, 1) \qquad y = f_1^{-1}(1, 0)$$

Computing with the cohomology ring

To disprove $P(\mathbb{T}^2)$ we need $x, y : H^1(\mathbb{T}^2)$ such that $x \smile y \not\equiv 0_h$. Let

$$x = f_1^{-1}(0, 1) \qquad\qquad y = f_1^{-1}(1, 0)$$

Now $f_2(x \smile y) \equiv 1$ holds by refl and thus $x \smile y \not\equiv 0_h$

Computing with the cohomology ring

To disprove $P(\mathbb{T}^2)$ we need $x, y : H^1(\mathbb{T}^2)$ such that $x \smile y \not\equiv 0_h$. Let

$$x = f_1^{-1}(0, 1) \qquad\qquad y = f_1^{-1}(1, 0)$$

Now $f_2(x \smile y) \equiv 1$ holds by refl and thus $x \smile y \not\equiv 0_h$

To prove $P(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$ we let $x, y : H^1(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$.

Computing with the cohomology ring

To disprove $P(\mathbb{T}^2)$ we need $x, y : H^1(\mathbb{T}^2)$ such that $x \smile y \not\equiv 0_h$. Let

$$x = f_1^{-1}(0, 1) \qquad\qquad y = f_1^{-1}(1, 0)$$

Now $f_2(x \smile y) \equiv 1$ holds by refl and thus $x \smile y \not\equiv 0_h$

To prove $P(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$ we let $x, y : H^1(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$. We have that $g_2(g_1^{-1}(g_1\ x) \smile g_1^{-1}(g_1\ y)) \equiv 0$, again by refl (modulo truncation elimination).

Computing with the cohomology ring

To disprove $P(\mathbb{T}^2)$ we need $x, y : H^1(\mathbb{T}^2)$ such that $x \smile y \not\equiv 0_h$. Let

$$x = f_1^{-1}(0, 1) \qquad\qquad y = f_1^{-1}(1, 0)$$

Now $f_2(x \smile y) \equiv 1$ holds by refl and thus $x \smile y \not\equiv 0_h$

To prove $P(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$ we let $x, y : H^1(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$. We have that $g_2(g_1^{-1}(g_1\ x) \smile g_1^{-1}(g_1\ y)) \equiv 0$, again by refl (modulo truncation elimination). Thus $g_1^{-1}(g_1\ x) \smile g_1^{-1}(g_1\ y) \equiv x \smile y \equiv 0_h$.

Computing with the cohomology ring

To disprove $P(\mathbb{T}^2)$ we need $x, y : H^1(\mathbb{T}^2)$ such that $x \smile y \not\equiv 0_h$. Let

$$x = f_1^{-1}(0, 1) \qquad\qquad y = f_1^{-1}(1, 0)$$

Now $f_2(x \smile y) \equiv 1$ holds by refl and thus $x \smile y \not\equiv 0_h$

To prove $P(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$ we let $x, y : H^1(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$. We have that $g_2(g_1^{-1}(g_1\ x) \smile g_1^{-1}(g_1\ y)) \equiv 0$, again by refl (modulo truncation elimination). Thus $g_1^{-1}(g_1\ x) \smile g_1^{-1}(g_1\ y) \equiv x \smile y \equiv 0_h$.

So $P(\mathbb{T}^2)$ does not hold while $P(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$ does, so these types are not equivalent

Further computations with cohomology rings

For a more ambitious computation involving $\smile$ consider Chapter 6 of Brunerie's PhD thesis. This chapter is devoted to proving that the generator $e : H^2(\mathbb{C}P^2)$ when multiplied with itself yields a generator of $H^4(\mathbb{C}P^2)$

Further computations with cohomology rings

For a more ambitious computation involving $\smile$ consider Chapter 6 of Brunerie's PhD thesis. This chapter is devoted to proving that the generator $e : H^2(\mathbb{C}P^2)$ when multiplied with itself yields a generator of $H^4(\mathbb{C}P^2)$

Let $g : \mathbb{Z} \to \mathbb{Z}$ be the map given by composing:

$$\mathbb{Z} \xrightarrow{\cong} H^2(\mathbb{C}P^2) \xrightarrow{\lambda\, x \to x \smile x} H^4(\mathbb{C}P^2) \xrightarrow{\cong} \mathbb{Z}$$

Further computations with cohomology rings

For a more ambitious computation involving $\smile$ consider Chapter 6 of Brunerie's PhD thesis. This chapter is devoted to proving that the generator $e : H^2(\mathbb{C}P^2)$ when multiplied with itself yields a generator of $H^4(\mathbb{C}P^2)$

Let $g : \mathbb{Z} \to \mathbb{Z}$ be the map given by composing:

$$\mathbb{Z} \xrightarrow{\cong} H^2(\mathbb{C}P^2) \xrightarrow{\lambda\, x \to x \smile x} H^4(\mathbb{C}P^2) \xrightarrow{\cong} \mathbb{Z}$$

The number $g(1)$ should reduce to ± 1 for $e \smile e$ to generate $H^4(\mathbb{C}P^2)$ and by evaluating it in `Cubical Agda` we should be able to reduce the whole chapter to a single computation...

Further computations with cohomology rings

For a more ambitious computation involving $\smile$ consider Chapter 6 of Brunerie's PhD thesis. This chapter is devoted to proving that the generator $e : H^2(\mathbb{C}P^2)$ when multiplied with itself yields a generator of $H^4(\mathbb{C}P^2)$

Let $g : \mathbb{Z} \to \mathbb{Z}$ be the map given by composing:

$$\mathbb{Z} \xrightarrow{\cong} H^2(\mathbb{C}P^2) \xrightarrow{\lambda\, x \to x \,\smile\, x} H^4(\mathbb{C}P^2) \xrightarrow{\cong} \mathbb{Z}$$

The number $g(1)$ should reduce to ± 1 for $e \smile e$ to generate $H^4(\mathbb{C}P^2)$ and by evaluating it in `Cubical Agda` we should be able to reduce the whole chapter to a single computation... However, `Cubical Agda` is currently stuck on computing $g(1)$

So this is yet another Brunerie number

Computations with cohomology rings

In *Computing Cohomology Rings in Cubical Agda* we have some more examples where it would be nice if things computed faster for characterizing $H^*(X, R)$ as quotients of polynomial rings

Computations with cohomology rings

In *Computing Cohomology Rings in Cubical Agda* we have some more examples where it would be nice if things computed faster for characterizing $H^*(X, R)$ as quotients of polynomial rings

For example, to show that $H^*(\mathbb{K}, \mathbb{Z}) \cong \mathbb{Z}[X, Y]/(X^2, XY, 2Y, Y^2)$ some computations are involved to show that the map $f : \mathbb{Z}[X, Y] \to H^*(\mathbb{K}, \mathbb{Z})$ is zero on the generators of $(X^2, XY, 2Y, Y^2)$

Computations with cohomology rings

In *Computing Cohomology Rings in Cubical Agda* we have some more examples where it would be nice if things computed faster for characterizing $H^*(X, R)$ as quotients of polynomial rings

For example, to show that $H^*(\mathbb{K}, \mathbb{Z}) \cong \mathbb{Z}[X, Y]/(X^2, XY, 2Y, Y^2)$ some computations are involved to show that the map $f : \mathbb{Z}[X, Y] \to H^*(\mathbb{K}, \mathbb{Z})$ is zero on the generators of $(X^2, XY, 2Y, Y^2)$

This gives even more examples of computations that are fast, slow, and some that don't terminate in a reasonable amount of time

Outline

Computational challenges

We now have lots of computational challenges:

- The new Brunerie numbers: β_1, β_2, and β_3
- The original Brunerie number β (and various reformulations of it)
- Our cohomology benchmarks
- Brunerie's $g(1)$ number
- Various cohomology ring computations
- ...

Computational challenges

We now have lots of computational challenges:

- The new Brunerie numbers: β_1, β_2, and β_3
- The original Brunerie number β (and various reformulations of it)
- Our cohomology benchmarks
- Brunerie's $g(1)$ number
- Various cohomology ring computations
- ...

β_3 seems stuck in `Cubical Agda`, but computes instantly in `cctt`! What about the other numbers?

Future/ongoing work

- Can we make implementations of cubical type theory faster and compute more things?
- Can we compute with univalence and HITs in non-cubical systems, e.g. H.O.T.T.? How does this compare in terms of efficiency?
- Can we get faster cohomology computations using synthetic cellular cohomology following Buchholtz-Favonia? Should allow us to reduce computations to linear algebra!
- Formalize more classical computational tools from algebraic topology (e.g. spectral sequences following van Doorn PhD)
- Serre finiteness theorem for homotopy groups of spheres (following Barton and Campion's synthetic proof). Gives that homotopy groups of spheres are finitely presented. Can we effectively compute these presentations?

Thank you for your attention!

Questions?