Computational Proofs in Cubical Type Theories

Anders Mörtberg

EuroProofNet WG6 Meeting, April 25, 2023

Outline

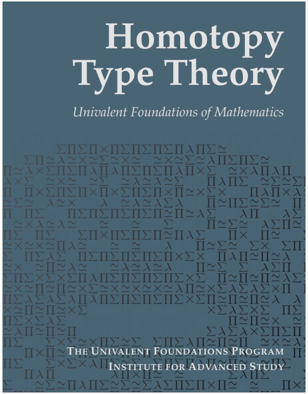
Outline

Homotopy Type Theory and Univalent Foundations

Aims to provide a *practical* foundations for computer formalization of mathematics

Builds on deep connections between type theory, homotopy theory and (higher) category theory

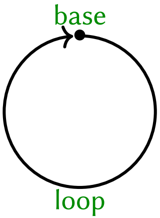
HoTT/UF = MLTT + Univalence + Higher Inductive Types



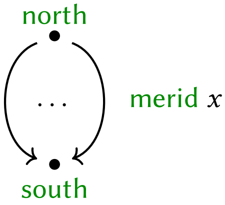
Higher Inductive Types (HITs)

Datatypes generated by regular "point" constructors and (higher) path constructors:

$\mathbb{S}^1$



$\mathsf{Susp}(\mathbb{S}^1)$



Higher spheres can either be defined by $\mathbb{S}^n := \mathsf{Susp}(\mathbb{S}^{n-1})$ or directly (for fixed n)

Synthetic algebraic topology

By representing spaces as types we can develop algebraic topology *synthetically* in HoTT/UF

Both homotopy and cohomology groups of types can be characterized using univalence

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

Problem: as univalence is added axiomatically to HoTT/UF we cannot compute with these results in proof assistants...

The Cubical paradigm in HoTT/UF

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:

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

Cubical proof assistants

There are by now a variety of different cubical type theories with native support for univalence and HITs, satisfying good metatheoretic properties (canonicity, normalization, decidable typechecking...)

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

Outline

Synthetic homotopy theory

In HoTT we define the nth *homotopy group* of a pointed type X by:

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

Synthetic homotopy theory

In HoTT we define the nth *homotopy group* of a pointed type X by:

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

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

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 hence be phrased as: *"there exists a number $\beta : \mathbb{Z}$ such that $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/\beta\mathbb{Z}$"*

The Brunerie number

The theorem can hence be phrased as: *"there exists a number $\beta : \mathbb{Z}$ such that $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}/\beta\mathbb{Z}$"*

In fact 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:

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

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

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

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 (Vikraman Choudhury, Paul Gustafson, Dan Licata, Ian Orton, and Jon 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

- 2018: various attempts to run parts of the computation in various cartesian cubical systems (`yacctt` and **redtt**) as well as in `Cubical Agda`, no luck
- 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 (Jon Sterling, Carlo Angiuli, Favonia, Dan Licata, Simon Huber, Ian Orton, Guillaume Brunerie) found various new optimizations to cubical evaluation ("Dagstuhl lemma"), did not help with computation

Computing the Brunerie number, a (probably incomplete) history

- 2018: various attempts to run parts of the computation in various cartesian cubical systems (`yacctt` and **redtt**) as well as in `Cubical Agda`, no luck
- 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 (Jon Sterling, Carlo Angiuli, Favonia, Dan Licata, Simon Huber, Ian Orton, Guillaume Brunerie) 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 Andrea Vezzosi)

Computing the Brunerie number, a (probably incomplete) history

- 2018: various attempts to run parts of the computation in various cartesian cubical systems (`yacctt` and **redtt**) as well as in `Cubical Agda`, no luck
- 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 (Jon Sterling, Carlo Angiuli, Favonia, Dan Licata, Simon Huber, Ian Orton, Guillaume Brunerie) 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 Andrea Vezzosi)
- 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

- 2018: various attempts to run parts of the computation in various cartesian cubical systems (`yacctt` and **`redtt`**) as well as in `Cubical Agda`, no luck
- 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 (Jon Sterling, Carlo Angiuli, Favonia, Dan Licata, Simon Huber, Ian Orton, Guillaume Brunerie) 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 Andrea Vezzosi)
- 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

- 2018: various attempts to run parts of the computation in various cartesian cubical systems (`yacctt` and **redtt**) as well as in `Cubical Agda`, no luck
- 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 (Jon Sterling, Carlo Angiuli, Favonia, Dan Licata, Simon Huber, Ian Orton, Guillaume Brunerie) 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 Andrea Vezzosi)
- 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`!

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 on the arxiv: `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. The computational proof relying on normalization

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

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 on the arxiv: `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. The computational proof relying on normalization

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

Let's look at what went into this...

Contents

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

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

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

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 $\mathbb{S}^3 \to \mathbb{S}^1 * \mathbb{S}^1 \to \mathbb{S}^2 \vee \mathbb{S}^2 \to \mathbb{S}^2$

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 1 (chapters 1–3)

The proof of this theorem uses:

- Hopf fibration
- LES of homotopy groups of a fibration
- Freudenthal suspension theorem
- James construction[1]
- The Blakers-Massey theorem
- Whitehead products

This is quite complicated synthetic HoTT, but all of it was formalizable and the proofs didn't contain any major surprises (except for a typo in the definition of Whitehead products)

[1]General form actually not needed, can do a direct encode-decode proof instead.

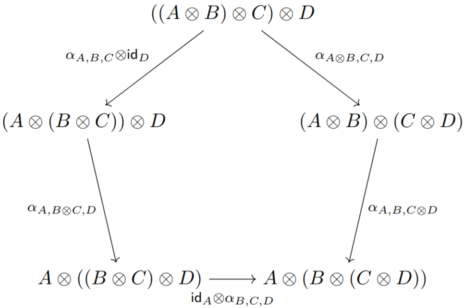
Brunerie's proof: part 2 (chapters 4–6)

The second half of the thesis is devoted to proving that $|\beta| = 2$ and this *a lot more complicated* than the first half. It uses the following classical theory:

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

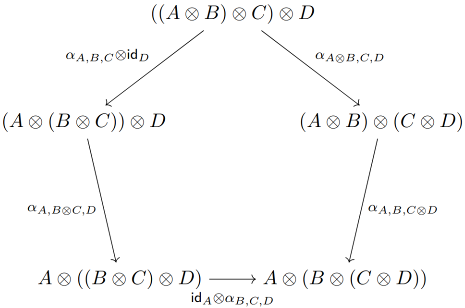
The second half of the thesis is devoted to proving that $|\beta| = 2$ and this *a lot more complicated* than the first half. It uses the following classical theory:

- Symmetric monoidal structure of smash products



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

- Symmetric monoidal structure of smash products



- This gives graded ring structure of the *cup product* $\smile : H^i(X) \to H^j(X) \to H^{i+j}(X)$

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

- The *Mayer-Vietoris sequence*:

$$\begin{tikzcd}
\widetilde{H}^{n+1}(D) \arrow[r, "i"] & \widetilde{H}^{n+1}(A) \times H^{n+1}(B) \arrow[r, "\Delta"] & H^{n+1}(C) \\
\widetilde{H}^{n}(D) \arrow[r, "i"] & \widetilde{H}^{n}(A) \times H^{n}(B) \arrow[r, "\Delta"] & H^{n}(C) \arrow[llu, "d"'] \\
\widetilde{H}^{n-1}(D) \arrow[r, "i"] & \widetilde{H}^{n-1}(A) \times H^{n-1}(B) \arrow[r, "\Delta"] & H^{n-1}(C) \arrow[llu, "d"']
\end{tikzcd}$$

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

- The *Mayer-Vietoris sequence*:

$$\begin{array}{ccccc} \widetilde{H}^{n+1}(D) & \xrightarrow{i} & \widetilde{H}^{n+1}(A) \times H^{n+1}(B) & \xrightarrow{\Delta} & H^{n+1}(C) \\ & \overset{d}{\longleftarrow} & & & \\ \widetilde{H}^{n}(D) & \xrightarrow{i} & \widetilde{H}^{n}(A) \times H^{n}(B) & \xrightarrow{\Delta} & H^{n}(C) \\ & \overset{d}{\longleftarrow} & & & \\ \widetilde{H}^{n-1}(D) & \xrightarrow{i} & \widetilde{H}^{n-1}(A) \times H^{n-1}(B) & \xrightarrow{\Delta} & H^{n-1}(C) \end{array}$$

- The *Gysin sequence*:

$$\mathbb{S}^{n-1} \longrightarrow E \xrightarrow{p} B$$

$$\ldots \longrightarrow H^{i-1}(E) \longrightarrow H^{i-n}(B) \xrightarrow{\smile e} H^{i}(B) \xrightarrow{p^*} H^{i}(E) \longrightarrow \ldots$$

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

- The *Hopf Invariant* homomorphism:

Definition 5.4.1. Given a pointed map $f : \mathbb{S}^{2n-1} \to \mathbb{S}^n$, we define

$$C_f := \mathbf{1} \sqcup^{\mathbb{S}^{2n-1}} \mathbb{S}^n,$$
$$\alpha_f := (i^*)^{-1}(\mathbf{c}_n) : H^n(C_f),$$
$$\beta_f := p^*(\mathbf{c}_{2n}) : H^{2n}(C_f),$$

Definition 5.4.2. The *Hopf invariant* of a pointed map $f : \mathbb{S}^{2n-1} \to \mathbb{S}^n$ is the integer $H(f) : \mathbb{Z}$ such that

$$\alpha_f^2 = H(f)\beta_f,$$

where α_f^2 is $\alpha_f \smile \alpha_f$.

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

- The *Iterated Hopf Construction*:

$$\begin{array}{ccccc}
A & \xleftarrow{\mathsf{fst}} & A \times (A \sqcup^{A \times A} A) & \xrightarrow{(a,x) \mapsto \nu'_a(x)} & \displaystyle\sum_{x:\Sigma A} H(x) \\
\big\downarrow \mathsf{id} & & \big\downarrow (a,x) \mapsto (a, \nu'_a(x)) & & \big\downarrow \mathsf{id} \\
A & \xleftarrow[\mathsf{fst}]{} & A \times \displaystyle\sum_{x:\Sigma A} H(x) & \xrightarrow[\mathsf{snd}]{} & \displaystyle\sum_{x:\Sigma A} H(x)
\end{array}$$

Brunerie's proof part 2

- 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

- **Symmetric monoidal structure of smash products**

  $\implies$ 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

- $(A \wedge B \to_\star C) \simeq (A \to_\star (B \to_\star C))$ **workaround** (Brunerie-Ljungström-M. CSL'22)
  $\implies$ 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

- **Symmetric monoidal structure of smash products** (Ljungström HoTT/UF'23 talk)
  $\Longrightarrow$ 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 $g : \mathbb{S}^3 \longrightarrow \mathbb{S}^2$. 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 $\pi_3^*(A) := ||\mathbb{S}^1 * \mathbb{S}^1 \longrightarrow_* A||_0$ 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

$$g_1 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^2 \qquad g_2 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^1 * \mathbb{S}^1 \qquad g_3 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^3$$

such that

$$e_1(|g|_0) = |g_1|_0 \qquad e_2(|g_1|_0) = |g_2|_0 \qquad e_3(|g_2|_0) = |g_3|_0 \qquad e_4(|g_3|_0) = -2$$

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

$$g_1 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^2 \qquad g_2 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^1 * \mathbb{S}^1 \qquad g_3 : \mathbb{S}^1 * \mathbb{S}^1 \longrightarrow \mathbb{S}^3$$

such that

$$e_1(|g|_0) = |g_1|_0 \qquad e_2(|g_1|_0) = |g_2|_0 \qquad e_3(|g_2|_0) = |g_3|_0 \qquad e_4(|g_3|_0) = -2$$

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

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)
2. Axel's new direct elementary proof which avoids part 2 of the thesis completely (600 LOC)
3. The new computational proof by normalizing one of these Brunerie numbers (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:[2]

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

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

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

Synthetic cohomology theory

In HoTT we can define cohomology as:[2]

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

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)

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

Side remark: relationship to homotopy groups of spheres

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

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

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

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 left-to-right map has better computational behavior than the one 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 2023)

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 2023)

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 $P : \mathsf{Type} \to \mathsf{Type}$:

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

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 $P : \mathsf{Type} \to \mathsf{Type}$:

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

We have the isomorphisms:

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

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

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

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

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 $P(\mathbb{T}^2)$ holds while $P(\mathbb{S}^2 \vee \mathbb{S}^1 \vee \mathbb{S}^1)$ doesn't, so these types are not equivalent

Further computations with cohomology rings

For a more ambitious computation involving $\smile$ consider Chapter 6 of Guillaume 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 Guillaume 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 Guillaume 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 Guillaume 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

Thomas Lamiaux's talk at the HoTT/UF workshop contained 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

Thomas Lamiaux's talk at the HoTT/UF workshop contained 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 the ideal that we quotient by

Computations with cohomology rings

Thomas Lamiaux's talk at the HoTT/UF workshop contained 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 the ideal that we quotient by

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

Synthetic computations in homotopy and cohomology theory

Some reflections on the above proofs by computation:

- Why does only the new Brunerie number $e_4(|g_3|_0)$ terminate? What about the other Brunerie numbers (especially Brunerie's original definition without optimizations)?
- Many computations are not very stable, composition with refl in certain places can make it run seemingly forever... Why?!
- Is it possible to get more computations to terminate in reasonable time? Maybe in other cubical type theories or faster implementations (taking closed term evaluation seriously)?
- What do the proofs actually tell us?

Outline

Cubical set models of HoTT/UF

Ideally these cubical proofs should be interpretable in spaces (Kan complexes), or in a suitably structured ∞-topos

Cubical set models of HoTT/UF

Ideally these cubical proofs should be interpretable in spaces (Kan complexes), or in a suitably structured ∞-topos

Unfortunately the model structure induced by the standard model of CCHM cubical type theory (on which `Cubical Agda` is based) is *not* Quillen equivalent to spaces

Cubical set models of HoTT/UF

Ideally these cubical proofs should be interpretable in spaces (Kan complexes), or in a suitably structured ∞-topos

Unfortunately the model structure induced by the standard model of CCHM cubical type theory (on which `Cubical Agda` is based) is *not* Quillen equivalent to spaces

Luckily, there at least is the *equivariant* cartesian cubical model which is equivalent to spaces (Awodey-Cavallo-Coquand-Riehl-Sattler)

More recently Cavallo and Sattler has proved that cartesian cubical sets with one connection is also equivalent to space: `https://arxiv.org/abs/2211.14801`

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 (AFH, ABCFHL)	✓		$r \to s$	✓
Cavallo-Sattler	✓	$\vee$	$0 \to r, 1 \to r$	✓
Equivariant (ACCRS)	✓		$\vec{r} \to \vec{s}$	✓

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 (AFH, ABCFHL)	✓		$r \to s$	✓
Cavallo-Sattler	✓	$\vee$	$0 \to r$, $1 \to r$	✓
Equivariant (ACCRS)	✓		$\vec{r} \to \vec{s}$	✓

The cartesian cubical model can be interpreted into Cavallo-Sattler model or the equivariant model, and hence any proof in cartesian cubical model has meaning in spaces

How do the other cubical type theories and models relate? Can we translate between them?

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 (AFH, ABCFHL)	✓		$r \to s$	✓
Cavallo-Sattler	✓	$\vee$	$0 \to r, 1 \to r$	✓
Equivariant (ACCRS)	✓		$\vec{r} \to \vec{s}$	✓

The cartesian cubical model can be interpreted into Cavallo-Sattler model or the equivariant model, and hence any proof in cartesian cubical model has meaning in spaces

How do the other cubical type theories and models relate? Can we translate between them?

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

Cubical set models of HoTT/UF

Translating the various definitions used in the computations to other cubical type theories should be easy, but translating the proofs is much harder...

Cubical set models of HoTT/UF

Translating the various definitions used in the computations to other cubical type theories should be easy, but translating the proofs is much harder...

Dream: `Cubical Agda` is conservative over cartesian cubical type theory

Seems very hard to prove, but it would mean that we do not have to change and redo all proofs in `Cubical Agda` if we want them to be interpretable into spaces

Cubical set models of HoTT/UF

Translating the various definitions used in the computations to other cubical type theories should be easy, but translating the proofs is much harder...

Dream: `Cubical Agda` is conservative over cartesian cubical type theory

Seems very hard to prove, but it would mean that we do not have to change and redo all proofs in `Cubical Agda` if we want them to be interpretable into spaces

Work in progress with Cavallo and Di Liberti: is cartesian cubical type theory wih *one* connection conservative over cartesian cubical type theory?

Outline

Future work

- Can we make implementations of cubical type theory faster and compute more things?
- How do the many cubical type theories relate? Are some conservative over others?
- 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)
- Very ambitious: Serre finiteness theorem for homotopy groups of spheres (following Barton-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?