A formalisation of the Serre finiteness theorem

Axel Ljungström

University of Nottingham

presenting j.w.w. Reid Barton, Owen Milner, Anders Mörtberg and Loïc Pujet

Introduction

- The homotopy groups of spheres are fundamental: they constitute the very building blocks of homotopy theory
  - Unfortunately: they are notoriously hard to compute
- Imagine if there was a formally verified algorithm which could quickly compute these...

Introduction

- The homotopy groups of spheres are fundamental: they constitute the very building blocks of homotopy theory
  - Unfortunately: they are notoriously hard to compute
- Imagine if there was a formally verified algorithm which $\overbrace{\textit{in theory}}$ could ~~quickly~~ compute these...

Introduction

- The homotopy groups of spheres are fundamental: they constitute the very building blocks of homotopy theory
  - Unfortunately: they are notoriously hard to compute
- Imagine if there was a formally verified algorithm which $\overbrace{\textit{in theory}}$ could ~~quickly~~ compute these...

$\Leftrightarrow$ Imagine if there was a formalisation of the Serre finiteness theorem in Cubical Agda...

Introduction

- The homotopy groups of spheres are fundamental: they constitute the very building blocks of homotopy theory
  - Unfortunately: they are notoriously hard to compute
- Imagine if there was a formally verified algorithm which $\overbrace{\text{in theory}}$ could ~~quickly~~ compute these...

$\Leftrightarrow$ Imagine if there was a formalisation of the Serre finiteness theorem in Cubical Agda...

- Good news: now there is! (and that is what this talk is about)

The Serre finiteness theorem

- A key concept is that of types which are *stably almost finite* (SAF). For now, let us black-box this definition.

```
postulate
  isSAF : Type → hProp
```

The Serre finiteness theorem

- A key concept is that of types which are *stably almost finite* (SAF). For now, let us black-box this definition.

```
postulate
  isSAF : Type → hProp
```

- The main theorem proved by Barton and Campion (and formalised by us) is the following

Theorem (Barton–Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The Serre finiteness theorem

- A key concept is that of types which are *stably almost finite* (SAF). For now, let us black-box this definition.

```
postulate
  isSAF : Type → hProp
```

- The main theorem proved by Barton and Campion (and formalised by us) is the following

Theorem (Barton–Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

  - Spheres in dimension ≥ 2 happen to be SAF and 1-connected (and in dimension < 2, their homotopy groups are well known)

The Serre finiteness theorem

- A key concept is that of types which are *stably almost finite* (SAF). For now, let us black-box this definition.

```
postulate
  isSAF : Type → hProp
```

- The main theorem proved by Barton and Campion (and formalised by us) is the following

Theorem (Barton–Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

  - Spheres in dimension ≥ 2 happen to be SAF and 1-connected (and in dimension < 2, their homotopy groups are well known)

Corollary (SFT)

$\pi_n(\mathbb{S}^m)$ is finitely presented for all n and m.

The Serre finiteness theorem

- A key concept is that of types which are *stably almost finite* (SAF). For now, let us black-box this definition.

```
postulate
  isSAF : Type → hProp
```

- The main theorem proved by Barton and Campion (and formalised by us) is the following

Theorem (Barton–Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

  - Spheres in dimension ≥ 2 happen to be SAF and 1-connected (and in dimension < 2, their homotopy groups are well known)

Corollary (SFT)

$\pi_n(\mathbb{S}^m)$ is finitely presented for all n and m.

- Goal now: give a rough idea of what the proof looks like
  - Need some definitions

Definitions: homotopy groups and connectedness

- **Homotopy groups:** Recall, for X pointed, we define $\pi_n(X) := \|\mathbb{S}^n \to_\star X\|_0$
  - or alternatively: $\pi_n(X) := \|\Omega^n X\|_0$

Definitions: homotopy groups and connectedness

- **Homotopy groups:** Recall, for X pointed, we define $\pi_n(X) := \|\mathbb{S}^n \to_\star X\|_0$
  - or alternatively: $\pi_n(X) := \|\Omega^n X\|_0$
- **Connectedness (of types):** A type X is said to be n-connected if $\|A\|_n$ is contractible
  - **Fact:** X is n-connected $\implies$ $\pi_k(X)$ vanishes for $k < n$.

Definitions: homotopy groups and connectedness

- **Homotopy groups:** Recall, for X pointed, we define $\pi_n(X) := \|\mathbb{S}^n \to_\star X\|_0$
  - or alternatively: $\pi_n(X) := \|\Omega^n X\|_0$
- **Connectedness (of types):** A type X is said to be n-connected if $\|A\|_n$ is contractible
  - **Fact:** X is n-connected $\implies$ $\pi_k(X)$ vanishes for $k < n$.
- **Connectedness (of functions):** A function $f : X \to Y$ is said to be n-connected if its fibres are n-connected types
  - **Fact:** f is n-connected $\implies$ $f_* : \pi_k(X) \to \pi_k(X)$ is an iso for $k \geq n$ and surjective for $k = n - 1$

Definitions: connected covers

Definition

Given a pointed type X, we define its n-connected cover, denoted $X\langle n \rangle$, to be the fibre of the truncation map $X \to \|X\|_n$. This gives us a fibre sequence

$$X\langle n \rangle \xrightarrow{\mathsf{proj}_1} X \xrightarrow{|-|} \|X\|_n$$

Definitions: connected covers

Definition

Given a pointed type X, we define its n-connected cover, denoted $X\langle n\rangle$, to be the fibre of the truncation map $X \to \|X\|_n$. This gives us a fibre sequence

$$X\langle n\rangle \xrightarrow{\mathsf{proj}_1} X \xrightarrow{|-|} \|X\|_n$$

- **Fact 1**: $X\langle n\rangle$ is, as the name suggests, n-connected

Definitions: connected covers

Definition

Given a pointed type X, we define its n-connected cover, denoted $X\langle n\rangle$, to be the fibre of the truncation map $X \to \|X\|_n$. This gives us a fibre sequence

$$X\langle n\rangle \xrightarrow{\mathsf{proj}_1} X \xrightarrow{|-|} \|X\|_n$$

- **Fact 1**: $X\langle n\rangle$ is, as the name suggests, n-connected
- **Fact 2**: $\pi_n(X\langle n-1\rangle) \cong \pi_n(X)$

Definitions: connected covers

Definition

Given a pointed type X, we define its n-connected cover, denoted $X\langle n\rangle$, to be the fibre of the truncation map $X \to \|X\|_n$. This gives us a fibre sequence

$$X\langle n\rangle \xrightarrow{\mathrm{proj}_1} X \xrightarrow{|-|} \|X\|_n$$

- **Fact 1**: $X\langle n\rangle$ is, as the name suggests, n-connected
- **Fact 2**: $\pi_n(X\langle n-1\rangle) \cong \pi_n(X)$
- **Conclusion**: if interested in $\pi_n(X)$, we can always replace X with $X\langle n-1\rangle$ which has the additional property of being $(n-1)$-connected
  - Key idea in SFT: compute $\pi_n(X\langle n-1\rangle)$ instead of $\pi_n(X)$

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

- Want to know $\pi_n(X)$ where X is SAF (and 1-connected)

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

- Want to know $\pi_n(X)$ where X is SAF (and 1-connected)
- This would be ““‘easy’”” if X were $(n-1)$-connected and SAF
  - follows easily from a result by myself and Pujet – more on this later

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

- Want to know $\pi_n(X)$ where X is SAF (and 1-connected)
- This would be ‘‘‘‘easy’’’’ if X were $(n-1)$-connected and SAF
  - follows easily from a result by myself and Pujet – more on this later
- No problem
  - just replace X with $X\langle n-1\rangle$!

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

- Want to know $\pi_n(X)$ where X is SAF (and 1-connected)
- This would be ““‘easy’”” if X were $(n-1)$-connected and SAF
  - follows easily from a result by myself and Pujet – more on this later
- No problem (?)
  - just replace X with $X\langle n-1\rangle$!

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

- Want to know $\pi_n(X)$ where X is SAF (and 1-connected)
- This would be ““‘easy’”” if X were $(n-1)$-connected and SAF
  - follows easily from a result by myself and Pujet – more on this later
- No problem (?)
  - just replace X with $X\langle n-1\rangle$!

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

- Want to know $\pi_n(X)$ where X is SAF (and 1-connected)
- This would be ""easy"" if X were $(n-1)$-connected and SAF
  - follows easily from a result by myself and Pujet – more on this later
- No problem (?)
  - just replace X with $X\langle n-1\rangle$!

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

- Want to know $\pi_n(X)$ where X is SAF (and 1-connected)
- This would be ““‘easy’”” if X were $(n-1)$-connected and SAF
  - follows easily from a result by myself and Pujet – more on this later
- No problem (?)
  - just replace X with $X\langle n-1\rangle$!

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

- Want to know $\pi_n(X)$ where X is SAF (and 1-connected)
- This would be ""easy"" if X were $(n-1)$-connected and SAF
  - follows easily from a result by myself and Pujet – more on this later
- No problem (?)
  - just replace X with $X\langle n-1\rangle$!

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

- Want to know $\pi_n(X)$ where X is SAF (and 1-connected)
- This would be ‘‘‘‘easy’’’’ if X were $(n-1)$-connected and SAF
  - follows easily from a result by myself and Pujet – more on this later
- No problem (?)
  - just replace X with $X\langle n-1\rangle$!

Proof idea of SFT

Recall the main theorem

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

The idea of the SFT proof is very simple:

- Want to know $\pi_n(X)$ where X is SAF (and 1-connected)
- This would be ‘‘‘‘easy’’’’ if X were $(n-1)$-connected and SAF
  - follows easily from a result by myself and Pujet – more on this later
- No problem (?)
  - just replace X with $X\langle n-1\rangle$!

Once we know $X\langle n-1\rangle$ is still SAF,

Proof idea of SFT

So, to summarise, we need two results:

Theorem A

If X is SAF and 1-connected, then $X\langle n\rangle$ is SAF for any $n \geq 1$

Theorem B

If X is SAF and $(n-1)$-connected, then $\pi_n(X)$ is finitely presentable

These two theorems guide the entire formalisation and also this presentation

Proof idea of SFT

Theorem A

If X is SAF and 1-connected, then $X\langle n \rangle$ is SAF for any $n \geq 1$

Let's take a look at how this one is proved ⇀

Part I: Proving Theorem A

Proving Theorem A: almost finiteness

- Elephant in the room: what does it mean for a type to be stably almost finite?
- This was one of the first things formalised, primarily by Owen
- The definition uses (finite) CW complexes. Black box these for now (this is what we did in the formalisation):

Proving Theorem A: almost finiteness

- Elephant in the room: what does it mean for a type to be stably almost finite?
- This was one of the first things formalised, primarily by Owen
- The definition uses (finite) CW complexes. Black box these for now (this is what we did in the formalisation):

```
postulate
  isFinCW : Type → hProp

  isFinCWFin : (n : ℕ) → isFinCW (Fin n)
  isFinCWPushout : (f : A → B) (g : A → C)
    → isFinCW A → isFinCW B → isFinCW C
    → isFinCW (Pushout f g)
  isFinCWΣ : {B : A → Type} → isFinCW A
    → ((a : A) → isFinCW (B a)) → isFinCW (Σ A B)
```

Proving Theorem A: almost finiteness

- Elephant in the room: what does it mean for a type to be stably almost finite?
- This was one of the first things formalised, primarily by Owen
- The definition uses (finite) CW complexes. Black box these for now (this is what we did in the formalisation):

```
postulate
  isFinCW : Type → hProp

  isFinCWFin : (n : ℕ) → isFinCW (Fin n)
  isFinCWPushout : (f : A → B) (g : A → C)
    → isFinCW A → isFinCW B → isFinCW C
    → isFinCW (Pushout f g)
  isFinCWΣ : {B : A → Type} → isFinCW A
    → ((a : A) → isFinCW (B a)) → isFinCW (Σ A B)


FinCW : Type₁ -- universe of finite CW complexes
FinCW = Σ[ A ∈ Type ] (isFinCW A)
```

Proving Theorem A: almost finiteness (contd)

Definition (n-finite types)

$$\text{is\_-finite} : (n : \mathbb{N}) \to \text{Type} \to \text{hProp}$$
$$\text{is } n \text{ -finite } X = \exists[\ C \in \text{CW}\ ]\ \Sigma[\ f \in (C \to X)\ ]\ (\text{isConnectedFun}\ (n \text{ - } 1)\ f)$$

Proving Theorem A: almost finiteness (contd)

Definition (n-finite types)

```
is_-finite : (n : ℕ) → Type → hProp
is n -finite X = ∃[ C ∈ CW ] Σ[ f ∈ (C → X) ] (isConnectedFun (n - 1) f)
                 × (is n -Dimensional C) -- may equivalently add
```

Proving Theorem A: almost finiteness (contd)

Definition (n-finite types)

```
is_-finite : (n : ℕ) → Type → hProp
is n -finite X = ∃[ C ∈ CW ] Σ[ f ∈ (C → X) ] (isConnectedFun (n - 1) f)
                 × (is n -Dimensional C) -- may equivalently add
```

Write $\Sigma\hat{}$: $(n : \mathbb{N}) \to \text{Type} \to \text{Type}$ for iterated suspension

Proving Theorem A: almost finiteness (contd)

Definition (n-finite types)

$$\begin{aligned}&\text{is\_-finite} : (n : \mathbb{N}) \to \text{Type} \to \text{hProp}\\&\text{is } n \text{ -finite } X = \exists[\ C \in \text{CW}\ ]\ \Sigma[\ f \in (C \to X)\ ]\ (\text{isConnectedFun } (n \text{ - } 1)\ f)\\&\qquad\qquad \times (\text{is } n \text{ -Dimensional } C)\ \texttt{-- may equivalently add}\end{aligned}$$

Write $\Sigma\hat{}\ : (n : \mathbb{N}) \to \text{Type} \to \text{Type}$ for iterated suspension

Definition (Stably n-finite types)

$$\begin{aligned}&\text{is-stably\_-finite} : (n : \mathbb{N}) \to \text{Type} \to \text{hProp}\\&\text{is-stably } n \text{ -finite } X = \exists[\ m \in \mathbb{N}\ ]\ (\text{is } (n + m) \text{ -finite } (\Sigma\hat{}\ m\ X))\end{aligned}$$

Proving Theorem A: almost finiteness (contd)

Definition (n-finite types)

$$\begin{aligned}&\text{is\_-finite} : (n : \mathbb{N}) \to \text{Type} \to \text{hProp}\\&\text{is } n \text{ -finite } X = \exists[\ C \in \text{CW}\ ]\ \Sigma[\ f \in (C \to X)\ ]\ (\text{isConnectedFun } (n \text{ - } 1)\ f)\\&\qquad\qquad \times (\text{is } n \text{ -Dimensional } C)\ \texttt{-- may equivalently add}\end{aligned}$$

Write $\Sigma\hat{}\ : (n : \mathbb{N}) \to \text{Type} \to \text{Type}$ for iterated suspension

Definition (Stably n-finite types)

$$\begin{aligned}&\text{is-stably\_-finite} : (n : \mathbb{N}) \to \text{Type} \to \text{hProp}\\&\text{is-stably } n \text{ -finite } X = \exists[\ m \in \mathbb{N}\ ]\ (\text{is } (n + m) \text{ -finite } (\Sigma\hat{}\ m\ X))\end{aligned}$$

Definition (Stably almost finite types)

$$\begin{aligned}&\text{isSAF} : \text{Type} \to \text{hProp}\\&\text{isSAF } X = (n : \mathbb{N}) \to \text{is-stably } n \text{ -finite } X\end{aligned}$$

Proving Theorem A: intuition behind SAF

- The intuition of SAF types is that their homology is finitely presentable
- Indeed, we have $H_n(X) \cong H_{n+m}(\Sigma^m X) \cong H_{n+m}(C)$ for some m and CW complex C. The latter group is always finitely presented.
- From this point of view, 'all we need to do' is to hope to be able to relate homology groups to homotopy groups (via tools like the Hurewicz theorem) in order to prove the main theorem
- Turns out: by talking about SAF types only, we never actually have to mention homology in the proof(!)

Proving Theorem A: Closure properties

- Recall, our goal is to prove that $X\langle n \rangle$ of a SAF type is SAF
- This will be deduced via a sequence of closure properties for SAF types

Lemma (Closure of n-finiteness under cofibres)

cofibClosure : $(f : X \to Y) \to$ is $(n$ - $1)$ -finite $X \to$ is n -finite Y
$\to$ is n -finite (cofib f)

Proving Theorem A: Closure properties

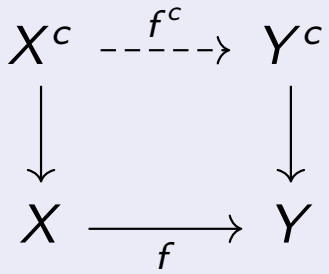
- Recall, our goal is to prove that $X\langle n \rangle$ of a SAF type is SAF
- This will be deduced via a sequence of closure properties for SAF types

Lemma (Closure of n-finiteness under cofibres)

cofibClosure : $(f : X \to Y) \to$ is $(n$ - $1)$ -finite $X \to$ is n -finite Y
$\to$ is n -finite (cofib f)

Proof sketch.

We get a diagram



Proving Theorem A: Closure properties

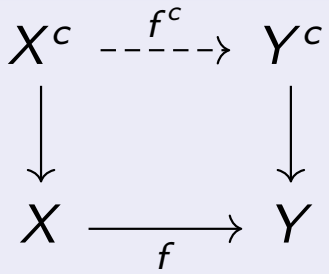
- Recall, our goal is to prove that $X\langle n \rangle$ of a SAF type is SAF
- This will be deduced via a sequence of closure properties for SAF types

Lemma (Closure of n-finiteness under cofibres)

cofibClosure : $(f : X \to Y) \to$ is $(n$ - $1)$ -finite $X \to$ is n -finite Y
$\to$ is n -finite (cofib f)

Proof sketch.

We get a diagram



- Here, X^c and Y^c are CW complexes and f^c can be deduced to (merely) exist via a basic connectedness argument

Proving Theorem A: Closure properties

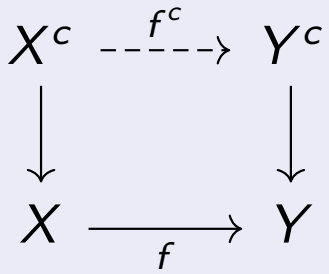
- Recall, our goal is to prove that $X\langle n\rangle$ of a SAF type is SAF
- This will be deduced via a sequence of closure properties for SAF types

Lemma (Closure of n-finiteness under cofibres)

cofibClosure : $(f : X \to Y) \to$ is $(n$ - $1)$ -finite $X \to$ is n -finite Y
$\to$ is n -finite (cofib f)

Proof sketch.

We get a diagram



- Here, X^c and Y^c are CW complexes and f^c can be deduced to (merely) exist via a basic connectedness argument
- We get an induced $(n-1)$-connected map of cofibres cofib $f^c \to$ cofib f

Proving Theorem A: Closure properties

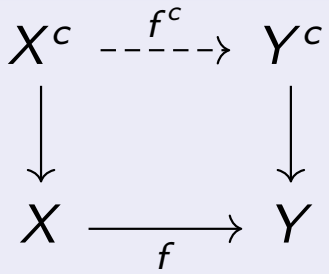
- Recall, our goal is to prove that $X\langle n\rangle$ of a SAF type is SAF
- This will be deduced via a sequence of closure properties for SAF types

Lemma (Closure of n-finiteness under cofibres)

cofibClosure : $(f : X \to Y) \to$ is $(n$ - $1)$ -finite $X \to$ is n -finite Y
$\to$ is n -finite (cofib f)

Proof sketch.

We get a diagram



- Here, X^c and Y^c are CW complexes and f^c can be deduced to (merely) exist via a basic connectedness argument
- We get an induced $(n-1)$-connected map of cofibres cofib $f^c \to$ cofib f
- We conclude by noting that cofib f^c is a finite CW complex by closure of (finite) CW complexes under pushouts □

Proving Theorem A: closure properties of SAFness

Suspensions commute with cofibres $\Longrightarrow$

Lemma

$$\text{isSAF-cofib} : (f : X \to Y) \to \text{isSAF}\ X \to \text{isSAF}\ Y \to \text{isSAF}\ (\text{cofib}\ f)$$

- Even better (and in non-Agda notation...):

Lemma (2 out of 3)

Let $X \xrightarrow{f} Y \to Z$ be a cofibre sequence (i.e. $Z \simeq \text{cofib}\ f$). If 2 types are SAF, then so is the third.

Lemma

Cartesian products, wedge sums, smash products and **joins** *of SAF types are SAF.*

Proving Theorem A: Ganea construction

- The previous page: closure properties for colimit construction
- Actually want: closure properties for limit constructions (fibres/loop spaces)
- They are proved using the above together with the *Ganea construction*

Theorem (Ganea)

Let $F \xrightarrow{j} E \to B$ *be a fibre sequence with* B *pointed. There is a fibre sequence* $F * \Omega B \to C_j \to B$

Proving Theorem A: Ganea construction

- The previous page: closure properties for colimit construction
- Actually want: closure properties for limit constructions (fibres/loop spaces)
- They are proved using the above together with the *Ganea construction*

Theorem (Ganea)

Let $F \xrightarrow{j} E \to B$ *be a fibre sequence with* B *pointed. There is a fibre sequence* $F * \Omega B \to C_j \to B$

- Apply Ganea to $\Omega B \to 1 \to B$: get a new fibre sequence

$$\Omega B * \Omega B \to C_{t:\Omega B \to 1} \to B$$

Proving Theorem A: Ganea construction

- The previous page: closure properties for colimit construction
- Actually want: closure properties for limit constructions (fibres/loop spaces)
- They are proved using the above together with the *Ganea construction*

Theorem (Ganea)

Let $F \xrightarrow{j} E \to B$ *be a fibre sequence with* B *pointed. There is a fibre sequence* $F * \Omega B \to C_j \to B$

- Apply Ganea to $\Omega B \to 1 \to B$: get a new fibre sequence

$$\Omega B * \Omega B \to C_{t:\Omega B \to 1} \to B$$

and again

$$\Omega B * \Omega B * \Omega B \to C_{t:\dots} \to B$$

and so on... We get a fibre sequence

$$*^n \Omega B \to E_n \xrightarrow{p_n} B$$

where E_n is an iterated cofibre (a type we understand by lemmas on previous slide).

Proving Theorem A: Ganea construction

- The previous page: closure properties for colimit construction
- Actually want: closure properties for limit constructions (fibres/loop spaces)
- They are proved using the above together with the *Ganea construction*

Theorem (Ganea)

Let $F \xrightarrow{j} E \to B$ be a fibre sequence with B pointed. There is a fibre sequence $F * \Omega B \to C_j \to B$

- Apply Ganea to $\Omega B \to 1 \to B$: get a new fibre sequence

$$\Omega B * \Omega B \to C_{t:\Omega B \to 1} \to B$$

and again

$$\Omega B * \Omega B * \Omega B \to C_{t:\dots} \to B$$

and so on... We get a fibre sequence

$$*^n \Omega B \to E_n \xrightarrow{p_n} B$$

where E_n is an iterated cofibre (a type we understand by lemmas on previous slide).

- B connected $\implies$ p_n is $(n-1)$ connected. This is enough to deduce the results on the following page.

Proving Theorem A: (more) closure properties of SAFness

In what follows, B is connected we assume there is a fibre sequence $F \to E \to B$.

Lemma (Spaces with SAF loop spaces are SAF)

$\mathsf{isSAF}\Omega\to\mathsf{isSAF} : \mathsf{isSAF}\ (\Omega\ B) \to \mathsf{isSAF}\ B$

Proving Theorem A: (more) closure properties of SAFness

In what follows, B is connected we assume there is a fibre sequence $F \to E \to B$.

Lemma (Spaces with SAF loop spaces are SAF)

isSAFΩ→isSAF : isSAF (Ω B) → isSAF B

Lemma (Same thing, other direction)

isSAFΩ : isConnected 1 B → isSAF B → isSAF (Ω B)

Proving Theorem A: (more) closure properties of SAFness

In what follows, B is connected we assume there is a fibre sequence $F \to E \to B$.

Lemma (Spaces with SAF loop spaces are SAF)

isSAFΩ→isSAF : isSAF (Ω B) → isSAF B

Lemma (Same thing, other direction)

isSAFΩ : isConnected 1 B → isSAF B → isSAF (Ω B)

- Consequence: $K(G, n)$ is SAF if G is finitely presentable

Proving Theorem A: (more) closure properties of SAFness

In what follows, B is connected we assume there is a fibre sequence $F \to E \to B$.

Lemma (Spaces with SAF loop spaces are SAF)

isSAFΩ→isSAF : isSAF (Ω B) → isSAF B

Lemma (Same thing, other direction)

isSAFΩ : isConnected 1 B → isSAF B → isSAF (Ω B)

- Consequence: $K(G, n)$ is SAF if G is finitely presentable

Lemma (Total spaces are SAF)

isSAF-total : isSAF B → isSAF (Ω B) → isSAF F → isSAF E

Proving Theorem A: (more) closure properties of SAFness

In what follows, B is connected we assume there is a fibre sequence $F \to E \to B$.

Lemma (Spaces with SAF loop spaces are SAF)

isSAFΩ→isSAF : isSAF (Ω B) → isSAF B

Lemma (Same thing, other direction)

isSAFΩ : isConnected 1 B → isSAF B → isSAF (Ω B)

- Consequence: $K(G, n)$ is SAF if G is finitely presentable

Lemma (Total spaces are SAF)

isSAF-total : isSAF B → isSAF (Ω B) → isSAF F → isSAF E

Lemma (Fibres are SAF)

isSAF-fibre : isConnected 1 B → isConnected 0 E
→ isSAF B → isSAF E → isSAF F

Proof of Theorem A

Using this, we can show the first key theorem (this time in Agda style)

Theorem A

$$\mathsf{TheoremA} : (n : \mathbb{N}_{\geq 1}) \to \mathsf{isConnected}\ 1\ X \to \mathsf{isSAF}\ X \to \mathsf{isSAF}\ (X\langle\, n \,\rangle)$$

Proof.

Show $X\langle n \rangle$ and $\|X\|_n$ are SAF mutually by induction on n. Easy using previous lemma + following two fibre sequences:

$$K(\pi_n(X), n) \to \|X\|_n \to \|X\|_{n-1}$$

$$X\langle n \rangle \to X \to \|X\|_n$$

□

Status check: Formalisation so far

The formalisation includes:

Status check: Formalisation so far

The formalisation includes:

- Definition + theory of (finite) CW complexes (postulated so far)

Status check: Formalisation so far

The formalisation includes:

- Definition + theory of (finite) CW complexes (postulated so far)
- Theory of (stably) n-finite/almost finite spaces
  - (A lot of) connectedness lemmas
  - Closure under cofibres
  - (Co)fibre sequences

Status check: Formalisation so far

The formalisation includes:

- Definition + theory of (finite) CW complexes (postulated so far)
- Theory of (stably) n-finite/almost finite spaces
  - (A lot of) connectedness lemmas
  - Closure under cofibres
  - (Co)fibre sequences
- Ganea construction

Status check: Formalisation so far

The formalisation includes:

- Definition + theory of (finite) CW complexes (postulated so far)
- Theory of (stably) n-finite/almost finite spaces
  - (A lot of) connectedness lemmas
  - Closure under cofibres
  - (Co)fibre sequences
- Ganea construction

All in all: 8000 lines of code (with some already in Cubical library)

Back to the main proof

We showed

Theorem

If X is 1-connected and SAF, then so is $X\langle n \rangle$ for any $n \geq 1$.

and know that $\pi_n(X) \cong \pi_n(X\langle n-1 \rangle)$ (*).

Back to the main proof

We showed

Theorem

If X is 1-connected and SAF, then so is $X\langle n\rangle$ for any $n \geq 1$.

and know that $\pi_n(X) \cong \pi_n(X\langle n-1\rangle)$ (*).

- In the end, we want:

Theorem

If X is SAF and 1-connected, then $\pi_n(X)$ is finitely presentable ($n \geq 2$).

Back to the main proof

We showed

Theorem

If X is 1-connected and SAF, then so is $X\langle n\rangle$ for any $n \geq 1$.

and know that $\pi_n(X) \cong \pi_n(X\langle n-1\rangle)$ (*).

- In the end, we want:

Theorem

If X is SAF and 1-connected, then $\pi_n(X)$ is finitely presentable ($n \geq 2$).

- With $(*)$, enough to show

Theorem B

If X is SAF and $(n-1)$-connected, then $\pi_n(X)$ is finitely presentable (≥ 2).

The weak Hurewicz theorem

- Let's inspect the premise of theorem further:
- X is $(n-1)$-connected $\implies$ $\pi_n(X) \cong \pi_{n+m}(\Sigma^m X)$ for any m (Freudenthal).

The weak Hurewicz theorem

- Let's inspect the premise of theorem further:
- X is $(n-1)$-connected $\implies$ $\pi_n(X) \cong \pi_{n+m}(\Sigma^m X)$ for any m (Freudenthal).
- X is SAF $\implies$ Can find m s.t. $\pi_{n+m}(\Sigma^m X) \cong \pi_{n+m}(C)$ for $(n+m-1)$-connected CW complex C.

The weak Hurewicz theorem

- Let's inspect the premise of theorem further:
- X is $(n-1)$-connected $\implies$ $\pi_n(X) \cong \pi_{n+m}(\Sigma^m X)$ for any m (Freudenthal).
- X is SAF $\implies$ Can find m s.t. $\pi_{n+m}(\Sigma^m X) \cong \pi_{n+m}(C)$ for $(n+m-1)$-connected CW complex C.
- Together, implies we only need:

Theorem (Weak Hurewicz Theorem)

Given C, an $(n-1)$-connected CW complex, $\pi_n(C)$ is finitely presented

The weak Hurewicz theorem

- Of course, the full Hurewicz theorem (proved in HoTT by Christensen and Scoccola) implies the weak one. We don't take this route because:
  - we haven't formalised Christensen and Scoccola's proof
  - only using weak version gives a *completely homology free* proof of SFT
  - but mainly: it was already (almost by accident) about to be formalised by L. and Loïc Pujet!
- Let's talk a little bit about my project with Loïc (and CW complexes in general) and its relation to the SFT formalisation

Part II: CW complexes and the Hurewicz theorem

2023, in a universe parallel to Carnegie Mellon University...

- Anders Mörtberg gets a new postdoc: Loïc Pujet
  - Idea for a project: reduce complex Brunerie number computation to linear algebra by developing cellular (co)homology in HoTT

2023, in a universe parallel to Carnegie Mellon University...

- Anders Mörtberg gets a new postdoc: Loïc Pujet
  - Idea for a project: reduce complex Brunerie number computation to linear algebra by developing cellular (co)homology in HoTT
- Anders goes on parental leave.
  - Loïc and I decide to use this time to develop the key theory of CW complexes we'll need for our projects
  - Our goal: develop enough theory to be able to prove the Hurewicz theorem (as a kind of long-term goal/sanity check)

2023, in a universe parallel to Carnegie Mellon University...

- Anders Mörtberg gets a new postdoc: Loïc Pujet
  - Idea for a project: reduce complex Brunerie number computation to linear algebra by developing cellular (co)homology in HoTT
- Anders goes on parental leave.
  - Loïc and I decide to use this time to develop the key theory of CW complexes we'll need for our projects
  - Our goal: develop enough theory to be able to prove the Hurewicz theorem (as a kind of long-term goal/sanity check)
  - Swedish parental leave $\implies$ Loïc and I have time to get quite far and end up writing a paper on these developments

2023, in a universe parallel to Carnegie Mellon University...

- Anders Mörtberg gets a new postdoc: Loïc Pujet
  - Idea for a project: reduce complex Brunerie number computation to linear algebra by developing cellular (co)homology in HoTT
- Anders goes on parental leave.
  - Loïc and I decide to use this time to develop the key theory of CW complexes we'll need for our projects
  - Our goal: develop enough theory to be able to prove the Hurewicz theorem (as a kind of long-term goal/sanity check)
  - Swedish parental leave $\implies$ Loïc and I have time to get quite far and end up writing a paper on these developments
- Anders is back, gets me more involved with the SFT project. Reminds me that Reid and Owen need a version of the Hurewicz theorem.

2023, in a universe parallel to Carnegie Mellon University...

- Anders Mörtberg gets a new postdoc: Loïc Pujet
  - Idea for a project: reduce complex Brunerie number computation to linear algebra by developing cellular (co)homology in HoTT
- Anders goes on parental leave.
  - Loïc and I decide to use this time to develop the key theory of CW complexes we'll need for our projects
  - Our goal: develop enough theory to be able to prove the Hurewicz theorem (as a kind of long-term goal/sanity check)
  - Swedish parental leave $\implies$ Loïc and I have time to get quite far and end up writing a paper on these developments
- Anders is back, gets me more involved with the SFT project. Reminds me that Reid and Owen need a version of the Hurewicz theorem.
- I show Reid our paper and he spots the weak Hurewicz theorem – although it's only every mentioned implicitly in the middle of our Hurewicz theorem proof
  - He notices it's just strong enough to reproduce the SFT proof but without ever appealing to homology computations!

My paper with Loïc

- In addition to the above, my work with Loïc fills the gaps we've left open so far in the SFT proof. Key contributions:

My paper with Loïc

- In addition to the above, my work with Loïc fills the gaps we've left open so far in the SFT proof. Key contributions:
  - A workable definition of CW complexes

My paper with Loïc

- In addition to the above, my work with Loïc fills the gaps we've left open so far in the SFT proof. Key contributions:
  - A workable definition of CW complexes
  - Basic constructions. In particular: pushouts of CW complexes

My paper with Loïc

- In addition to the above, my work with Loïc fills the gaps we've left open so far in the SFT proof. Key contributions:
  - A workable definition of CW complexes
  - Basic constructions. In particular: pushouts of CW complexes
  - The 'Hurewicz approximation theorem' (a characterisation of low dimensional skeleta of n-connected CW complexes)
    - ⋆ This + computation of homotopy groups of a certain class of pushouts = weak Hurewicz Theorem

My paper with Loïc

- In addition to the above, my work with Loïc fills the gaps we've left open so far in the SFT proof. Key contributions:
  - A workable definition of CW complexes
  - Basic constructions. In particular: pushouts of CW complexes
  - The 'Hurewicz approximation theorem' (a characterisation of low dimensional skeleta of n-connected CW complexes)
    - ⋆ This + computation of homotopy groups of a certain class of pushouts = weak Hurewicz Theorem
- Also, a bunch of stuff about cellular homology

My paper with Loïc

- In addition to the above, my work with Loïc fills the gaps we've left open so far in the SFT proof. Key contributions:
  - A workable definition of CW complexes
  - Basic constructions. In particular: pushouts of CW complexes
  - The 'Hurewicz approximation theorem' (a characterisation of low dimensional skeleta of n-connected CW complexes)
    - ⋆ This + computation of homotopy groups of a certain class of pushouts = weak Hurewicz Theorem
- Also, a bunch of stuff about cellular homology
- Let's take a look at what we did!

Loïc paper: CW complexes

- CW complexes are defined as follows. Very implementation sensitive!

Definition (CW structures)

```
record CWStr : Type₁ where
  field
    -- Unerlying sequence of types
    Skel : (n : ℕ₋₁) → Type
    ι : (n : ℕ₋₁) → Skel n → Skel (succ n)

    -- Number of cells in each dimension
    CellSize : ℕ → ℕ

    -- attaching maps
    α : (n : ℕ) → S (n -1) × Fin (CellSize n) → Skel (n -1)

    -- axioms
    Skel₋₁ : ¬ (Skel -1)
    Skel₊Pushout : (n : ℕ) → Skel n ≃ Pushout (α n) snd
```

Loïc paper: CW complexes

- CW complexes are defined as follows. Very implementation sensitive!

Definition (CW structures)

```
record CWStr : Type₁ where
  field
    -- Unerlying sequence of types
    Skel : (n : ℕ₋₁) → Type
    ι : (n : ℕ₋₁) → Skel n → Skel (succ n)

    -- Number of cells in each dimension
    CellSize : ℕ → ℕ

    -- attaching maps
    α : (n : ℕ) → S (n -1) × Fin (CellSize n) → Skel (n -1)

    -- axioms
    Skel₋₁ : ¬ (Skel -1)
    Skel₊Pushout : (n : ℕ) → Skel n ≃ Pushout (α n) snd
```

$$\begin{array}{ccc} S^{n-1} \times \mathrm{Fin}(\mathrm{CellSize}\ n) & \xrightarrow{\mathrm{snd}} & \mathrm{Fin}(\mathrm{CellSize}\ n) \\ {\scriptstyle \alpha_n}\downarrow & & \downarrow \\ \mathrm{Skel}_{n-1} & \xrightarrow{\ulcorner} & \mathrm{Skel}_n \end{array}$$

Loïc paper: CW complexes

Definition (Finite CW structures)

```
-- Finite CW structures of a given dimension
isFinCWStr_ofDim_ : CWStr → ℕ₋₁ → hProp
isFinCWStr X ofDim n = (m : ℕ₋₁) → n ≤ m → isEquiv (ι X n)

-- Finite CW structures
isFinCWStr : CWStr → hProp
isFinCWStr X = ∃[ n ∈ ℕ₋₁ ] (isFinCWStr X ofDim n)
```

Definition (Finite CW complexes)

```
-- Property of (a type) being a finite CW complex
isFinCW : Type → hProp
isFinCW X = ∃[ S ∈ CWStr ] (isFinCWStr S × colim S ≃ X)

-- Universe of finite CW complexes
FinCW : Type₁
FinCW = Σ[ A ∈ Type ] (isFinCW A)
```

Loïc paper: Pushouts

- We can, of course, remove the finiteness assumption to get a more general class CW.

Theorem

Given a span of CW complexes $B \xleftarrow{f} A \xrightarrow{g} C$, where A is finite, its pushout is again a CW complex (finite if B and C are).

Proof.

Needs a (constructive version) of the cellular approximation theorem + a bunch of technical path algebra. □

- This was crucial in the SFT proof and only postulated until now!

Definition (Hurewicz CW structures)

```
record hasHurewiczStr (X : CWStr) (n : ℕ₋₁) : Type₁ where
  open CWStr X
  field
    lowSkelVanish : (m : ℕ) → m ≤ n → isContr (Skel m)

    A : FinSet
    B : FinSet
    -- f : ⋁_A S^(n+1) → ⋁_B S^(n+1)
    f : ∨ A (S (succ n)) →• ∨ B (S (succ n))

    -- X_(n+1) ≃ ⋁_A S^(n+1)
    bottomLevel : Skel (succ n) ≃ ∨ A (S (succ n))
    -- X_(n+2) ≃ cofib (f : ⋁_A S^(n+1) → ⋁_B S^(n+1))
    bottomLevel+1 : Skel (succ (succ n)) ≃ cofib f
```

Definition (Hurewicz connectedness)

```
isHurewiczConnected : (n : ℕ₋₁) → Type → hProp
isHurewiczConnected n X = ∃[ X' ∈ CWStr ] (hasHurewiczStr X' n) × (colim X' ≃ X)
```

Loïc paper: Hurewicz connectedness

- Significance: if X is Hurewicz n-connected, we (merely) have some $\mathsf{f} : \bigvee_{\mathsf{A}} S^{n+1} \to_\star \bigvee_{\mathsf{B}} S^{n+1}$ and

$$\pi_{n+1}(X) \cong \pi_{n+1}(X_{n+2}) \cong \pi_{n+1}(\text{cofib } \mathsf{f})$$

- We can compute the 'degree' of such a map f.

$$\deg(\mathsf{f}) : \mathrm{Hom}(\mathbb{Z}[\mathsf{A}], \mathbb{Z}[\mathsf{B}])$$

Theorem

$\pi_{n+1}^{\mathrm{ab}}(\text{cofib } \mathsf{f}) \cong \mathbb{Z}[\mathsf{B}]/\mathrm{im}(\deg(\mathsf{f}))$

Proof.

Direct computation when $n = 0$ and Blakers–Massey when $n > 0$. □

Loïc paper: Hurewicz connectedness

- Significance: if X is Hurewicz n-connected, we (merely) have some $\mathsf{f} : \bigvee_{\mathsf{A}} S^{n+1} \to_\star \bigvee_{\mathsf{B}} S^{n+1}$ and

$$\pi_{n+1}(X) \cong \pi_{n+1}(X_{n+2}) \cong \pi_{n+1}(\text{cofib } \mathsf{f})$$

- We can compute the 'degree' of such a map f.

$$\deg(\mathsf{f}) : \mathrm{Hom}(\mathbb{Z}[\mathsf{A}], \mathbb{Z}[\mathsf{B}])$$

Theorem

$\pi_{n+1}^{\mathrm{ab}}(\text{cofib } \mathsf{f}) \cong \mathbb{Z}[\mathsf{B}]/\mathrm{im}(\deg(\mathsf{f}))$ ← *this is what it means to be finitely presented*

Proof.

Direct computation when $n = 0$ and Blakers–Massey when $n > 0$. □

Loïc paper: the weak Hurewicz theorem

Theorem (Weak(er) Hurewicz theorem)

*Given X, an $(n-1)$-***Hurewicz** *connected CW complex, $\pi_n(X)$ is finitely presented*

Loïc paper: the weak Hurewicz theorem

Theorem (Weak(er) Hurewicz theorem)

*Given X, an $(n-1)$-**Hurewicz** connected CW complex, $\pi_n(X)$ is finitely presented*

Theorem (Hurewicz approximation theorem)

A CW complex X is Hurewicz n-connected iff it is n-connected

Loïc paper: the weak Hurewicz theorem

Theorem (Weak~~(er)~~ Hurewicz theorem)

Given X, an $(n-1)$-~~Hurewicz~~ connected CW complex, $\pi_n(X)$ is finitely presented

Theorem (Hurewicz approximation theorem)

A CW complex X is Hurewicz n-connected iff it is n-connected

Summarising the proof of Serre finiteness

So, let's put everything together:

Summarising the proof of Serre finiteness

So, let's put everything together:

- Let X be 1-connected and SAF and fix $n \geq 2$

Summarising the proof of Serre finiteness

So, let's put everything together:

- Let X be 1-connected and SAF and fix $n \geq 2$
- $\pi_n(X) \cong \pi_n(X\langle n-1 \rangle)$

Summarising the proof of Serre finiteness

So, let's put everything together:

- Let X be 1-connected and SAF and fix $n \geq 2$
- $\pi_n(X) \cong \pi_n(X\langle n-1\rangle)$
- X is SAF $\implies$ $X\langle n-1\rangle$ is SAF and $(n-1)$-connected
  $\implies$ $\pi_n(X\langle n-1\rangle) \cong \pi_n(C)$ for $(n-1)$-connected CW complex C.

Summarising the proof of Serre finiteness

So, let's put everything together:

- Let X be 1-connected and SAF and fix $n \geq 2$
- $\pi_n(X) \cong \pi_n(X\langle n-1\rangle)$
- X is SAF $\implies$ $X\langle n-1\rangle$ is SAF and $(n-1)$-connected
  $\implies$ $\pi_n(X\langle n-1\rangle) \cong \pi_n(C)$ for $(n-1)$-connected CW complex C.
- So $\pi_n(C)$ is FP by the weak Hurewicz theorem. Hence we have the main theorem:

Summarising the proof of Serre finiteness

So, let's put everything together:

- Let X be 1-connected and SAF and fix $n \geq 2$
- $\pi_n(X) \cong \pi_n(X\langle n-1\rangle)$
- X is SAF $\implies$ $X\langle n-1\rangle$ is SAF and $(n-1)$-connected
  $\implies$ $\pi_n(X\langle n-1\rangle) \cong \pi_n(C)$ for $(n-1)$-connected CW complex C.
- So $\pi_n(C)$ is FP by the weak Hurewicz theorem. Hence we have the main theorem:

Theorem (Barton-Campion)

Let X be a 1-connected pointed type. If X is SAF, then $\pi_n(X)$ is finitely presented for $n \geq 2$.

Part III: On the formalisation and some lessons I've learnt

Lessons learnt: Finicking about with Fin

Lesson 1

Implementation of Fin *matters when formalising things like finite subsequences (e.g. finite subcomplexes of a CW complex)*

To avoid transport hell, we want equations like the following to hold definitionally:

$$\begin{array}{ccc} \mathrm{Fin}(\mathrm{suc}\ n) & \xrightarrow{+1} & \mathrm{Fin}(\mathrm{suc}\ (\mathrm{suc}\ n)) \\ \uparrow & & \uparrow \\ \mathrm{Fin}(n) & \xrightarrow[+1]{} & \mathrm{Fin}(\mathrm{suc}\ n) \end{array}$$

Lessons learnt: Finicking about with Fin

Lesson 1

Implementation of Fin *matters when formalising things like finite subsequences (e.g. finite subcomplexes of a CW complex)*

To avoid transport hell, we want equations like the following to hold definitionally:

$$\begin{array}{ccc} \mathrm{Fin}(\mathrm{suc}\ n) & \xrightarrow{+1} & \mathrm{Fin}(\mathrm{suc}\ (\mathrm{suc}\ n)) \\ \uparrow & & \uparrow \\ \mathrm{Fin}(n) & \xrightarrow[+1]{} & \mathrm{Fin}(\mathrm{suc}\ n) \end{array}$$

- Indexed-inductive version: reasonable but not supported in Cubical Agda

Lessons learnt: Finicking about with Fin

Lesson 1

Implementation of Fin *matters when formalising things like finite subsequences (e.g. finite subcomplexes of a CW complex)*

To avoid transport hell, we want equations like the following to hold definitionally:

$$\begin{array}{ccc} \mathrm{Fin}(\mathrm{suc}\ n) & \xrightarrow{+1} & \mathrm{Fin}(\mathrm{suc}\ (\mathrm{suc}\ n)) \\ \uparrow & & \uparrow \\ \mathrm{Fin}(n) & \xrightarrow[+1]{} & \mathrm{Fin}(\mathrm{suc}\ n) \end{array}$$

- Indexed-inductive version: reasonable but not supported in Cubical Agda
- 'Standard' definition: $\mathrm{Fin}(n) := \Sigma_{m:\mathbb{N}}(m < n)$ only behaves nicely if $<$ is appropriately defined:

Lessons learnt: Finicking about with Fin

Lesson 1

Implementation of Fin *matters when formalising things like finite subsequences (e.g. finite subcomplexes of a CW complex)*

To avoid transport hell, we want equations like the following to hold definitionally:

$$\begin{array}{ccc} \mathrm{Fin}(\mathrm{suc}\ n) & \xrightarrow{+1} & \mathrm{Fin}(\mathrm{suc}\ (\mathrm{suc}\ n)) \\ \uparrow & & \uparrow \\ \mathrm{Fin}(n) & \xrightarrow[+1]{} & \mathrm{Fin}(\mathrm{suc}\ n) \end{array}$$

- Indexed-inductive version: reasonable but not supported in Cubical Agda
- 'Standard' definition: $\mathrm{Fin}(n) := \Sigma_{m:\mathbb{N}}(m < n)$ only behaves nicely if $<$ is appropriately defined:
  - Bad: $m < n = \Sigma[\ x \in \mathbb{N}\ ]\ ((\mathrm{suc}\ x) + m \equiv n)$

Lessons learnt: Finicking about with Fin

Lesson 1

Implementation of Fin *matters when formalising things like finite subsequences (e.g. finite subcomplexes of a CW complex)*

To avoid transport hell, we want equations like the following to hold definitionally:

$$\begin{array}{ccc} \mathrm{Fin}(\mathrm{suc}\ n) & \xrightarrow{+1} & \mathrm{Fin}(\mathrm{suc}\ (\mathrm{suc}\ n)) \\ \uparrow & & \uparrow \\ \mathrm{Fin}(n) & \xrightarrow[+1]{} & \mathrm{Fin}(\mathrm{suc}\ n) \end{array}$$

- Indexed-inductive version: reasonable but not supported in Cubical Agda
- 'Standard' definition: $\mathrm{Fin}(n) := \Sigma_{m:\mathbb{N}}(m < n)$ only behaves nicely if $<$ is appropriately defined:
  - Bad: $m < n = \Sigma[\ x \in \mathbb{N}\ ]\ ((\mathrm{suc}\ x) + m \equiv n)$
  - Good:

$$\begin{aligned} &\_<\_ : \mathbb{N} \to \mathbb{N} \to \mathrm{hProp} \\ &a < \mathrm{zero} = \bot \\ &\mathrm{zero} < \mathrm{suc}\ b = \top \\ &\mathrm{suc}\ a < \mathrm{suc}\ b = a < b \end{aligned}$$

Lessons learnt: universe polymorphism

Lesson 2

Be universe polymorphic **from the start**

- Just do it...

Lessons learnt: universe polymorphism

Lesson 2

Be universe polymorphic **from the start**

- Just do it...
- ...even if it forces you do define things like this:

$$
\begin{aligned}
&\ell\text{-maxList} : \text{List Level} \to \text{Level} \\
&\ell\text{-maxList } [] = \ell\text{-zero} \\
&\ell\text{-maxList } (\ell :: \ell s) = \ell\text{-max } \ell \ (\ell\text{-maxList } \ell s)
\end{aligned}
$$

Lessons learnt: generality vs strictness

Lesson 3

Sacrificing generality in a definition can often lead to better computational behaviour... But don't exclude the option of having both!

Example: Def of finite CW complexes

Lessons learnt: generality vs strictness

Lesson 3

Sacrificing generality in a definition can often lead to better computational behaviour... But don't exclude the option of having both!

Example: Def of finite CW complexes

- Pros of recursive definition:
  - pushout equation holds definitionally

Lessons learnt: generality vs strictness

Lesson 3

Sacrificing generality in a definition can often lead to better computational behaviour... But don't exclude the option of having both!

Example: Def of finite CW complexes

- Pros of recursive definition:
  - pushout equation holds definitionally
- Pros of our definition:
  - flexibility; for instance, it is easier to make alterations to CW structures (e.g. swapping out the n-skeleton for something else without affecting any higher skeleta).
  - works for infinite complexes too

Lessons learnt: generality vs strictness

Lesson 3

Sacrificing generality in a definition can often lead to better computational behaviour... But don't exclude the option of having both!

Example: Def of finite CW complexes

- Pros of recursive definition:
  - pushout equation holds definitionally
- Pros of our definition:
  - flexibility; for instance, it is easier to make alterations to CW structures (e.g. swapping out the n-skeleton for something else without affecting any higher skeleta).
  - works for infinite complexes too
- Strict pushout equation is very tempting but we still wish to go with our definition. Solution: define a strictification functor (if possible)

$$\mathsf{strictify} : \mathsf{CW} \to \mathsf{CW}$$

s.t. $X \simeq \mathsf{strictify}\ X$ but the latter type enjoys the desired strict equalities.

Lessons learnt: encode-decode is your friend

Lesson 4

Encode-decode works for π_1^{ab} too

- Example: needed to compute $\pi_1^{\mathsf{ab}}(\bigvee_A S^1)$
- Can use Seifert–Van Kampen but that requires a bunch of set-up (and some algebra)
- Can be computed completely 'homotopy-theoretically' (and very directly) using encode-decode

Lessons learnt: encode-decode is your friend

Lesson 4

Encode-decode works for π_1^{ab} too

- Example: needed to compute $\pi_1^{\mathsf{ab}}(\bigvee_A S^1)$
- Can use Seifert–Van Kampen but that requires a bunch of set-up (and some algebra)
- Can be computed completely 'homotopy-theoretically' (and very directly) using encode-decode
- To make it work for π_1^{ab} instead of π_1, just swap the identity type for the following

```
data _≡ᵃᵇ_ (x y : A) : Type
  where
  [_] : x ≡ y → x ≡ᵃᵇ y
  com : (p q r : x ≡ y) → [ p · q ⁻¹ · r ] ≡ [ r · q ⁻¹ · p ]
```

and carry out the encode-decode they way you're used to.

Formalisation

- The formalisation is available at `https://github.com/CMU-HoTT/serre-finiteness`
- Just under 10k LOC here but this excludes most of the work on CW complexes and many other results from the Cubical library
- Plan: get it merged into the Cubical Agda library.
- Material on CW complexes by Loïc and me: available in the library already. See `https://github.com/agda/cubical/tree/master/Cubical/CW` in particular

So what about computations?

- So now that this has been formalised, this means that we can plug any fixed $\pi_n(S^m)$ into Agda and compute a list of integers $(r_0, r_1, \ldots r_k)$ s.t.

$$\pi_n(S^m) \cong \mathbb{Z}^{r_0} \times \Pi_{1 \leq i \leq k}(\mathbb{Z}/m_{r_i}\mathbb{Z})$$

So what about computations?

- So now that this has been formalised, this means that we can plug any fixed $\pi_n(S^m)$ into Agda and compute a list of integers $(r_0, r_1, \ldots r_k)$ s.t.

$$\pi_n(S^m) \cong \mathbb{Z}^{r_0} \times \Pi_{1 \leq i \leq k}(\mathbb{Z}/m_{r_i}\mathbb{Z})$$

- *Insert meme with Guillaume Brunerie*

So what about computations?

- So now that this has been formalised, this means that we can plug any fixed $\pi_n(S^m)$ into Agda and compute a list of integers $(r_0, r_1, \ldots r_k)$ s.t.

$$\pi_n(S^m) \cong \mathbb{Z}^{r_0} \times \Pi_{1 \leq i \leq k}(\mathbb{Z}/m_{r_i}\mathbb{Z})$$

- *Insert meme with Guillaume Brunerie*
- Haven't even been able to get the number of generators of $\pi_2(S^2)$

Future work: improving the result

- Technically, we have only proved the mere existence of the list of integers and isomorphism mentioned on the previous page
- We can get out of the propositional truncation by proving an appropriate uniqueness result – this needs a proof of the uniqueness of the Smith normal form
- Has been done before, but not in Cubical Agda. Not crucial, but would be cool to have!

Thanks!