Polygraphs in homotopy type theory

Samuel Mimram

8-9 June 2023 / Métayer days

Polygraphs

The research of François

revolves around **polygraphs**.

A future for polygraphs

I will try to present an overview of recent results around

polygraphs in **homotopy type theory**

(which I recently started working on)

Some recent investigations around polygraphs

The general plan:

1. polygraphs for ω-categories are not right from a topological pov
2. we can define polygraphs for ∞-groupoids in HoTT
3. we can adapt traditional (rewriting) methods in this setting
4. we have new powerful methods to construct polygraphs

This is an overview: most of what I will present is not due to me (excepting errors).

We are currently investigating this with Émile Oleon.

Part I

Traditional polygraphs are not right

Polygraphs as free ω-categories

Polygraphs are (presentations of) free ω-categories, constructed from generators:

- 0-cells

$$x \qquad\qquad y \qquad\qquad z$$

- 1-cells

$$x \xrightarrow{f} x$$

- 2-cells

$$\begin{array}{ccccc} & & x & & \\ & \overset{f}{\nearrow} & \Downarrow \alpha & \overset{f}{\searrow} & \\ x & & \xrightarrow[f]{\qquad\qquad} & & x \end{array}$$

- etc.

This provides a good notion of presentation of category.

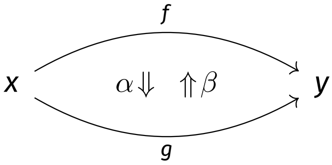
Equivalence between ω-categories

In an ω-category C, two cells $f, g : x \to y$ are **equivalent**, noted $f \sim g$, when there are cells

$$\alpha : f \Rightarrow g \qquad\qquad \beta : g \Rightarrow f$$

such that

$$\beta \circ \alpha \sim \mathrm{id}_f \qquad\qquad \alpha \circ \beta \sim \mathrm{id}_g$$



(this is a coinductive definition!)

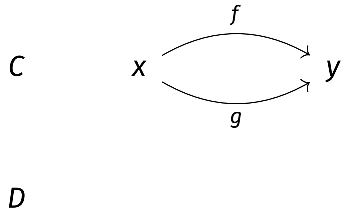
Equivalence between ω-categories

In an ω-category C, two cells $f, g : x \to y$ are **equivalent**, noted $f \sim g$, when there are cells

$$\alpha : f \Rightarrow g \qquad\qquad \beta : g \Rightarrow f$$

such that

$$\beta \circ \alpha \sim \mathrm{id}_f \qquad\qquad \alpha \circ \beta \sim \mathrm{id}_g$$



An *ω-functor* $F : C \to D$ is a **weak equivalence** when it is "surjective up to equivalence": given $f, g : x \to y$ and $\beta : Ff \Rightarrow Fg$ there is $\alpha : f \Rightarrow g$ such that $F\alpha = \beta$.

Equivalence between ω-categories

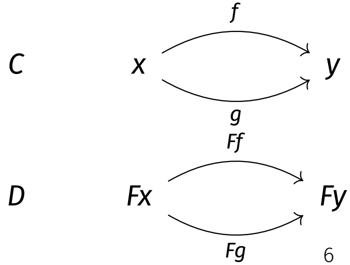
In an ω-category C, two cells $f, g : x \to y$ are **equivalent**, noted $f \sim g$, when there are cells

$$\alpha : f \Rightarrow g \qquad\qquad \beta : g \Rightarrow f$$

such that

$$\beta \circ \alpha \sim \mathrm{id}_f \qquad\qquad \alpha \circ \beta \sim \mathrm{id}_g$$

An *ω-functor* $F : C \to D$ is a **weak equivalence** when it is "surjective up to equivalence": given $f, g : x \to y$ and $\beta : Ff \Rightarrow Fg$ there is $\alpha : f \Rightarrow g$ such that $F\alpha = \beta$.



Equivalence between ω-categories

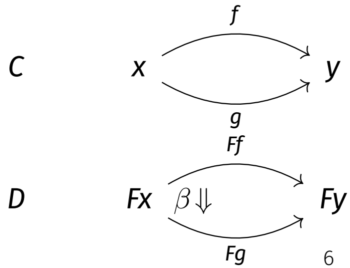
In an ω-category C, two cells $f, g : x \to y$ are **equivalent**, noted $f \sim g$, when there are cells

$$\alpha : f \Rightarrow g \qquad\qquad \beta : g \Rightarrow f$$

such that

$$\beta \circ \alpha \sim \mathrm{id}_f \qquad\qquad \alpha \circ \beta \sim \mathrm{id}_g$$

An ω-*functor* $F : C \to D$ is a **weak equivalence** when it is "surjective up to equivalence": given $f, g : x \to y$ and $\beta : Ff \Rightarrow Fg$ there is $\alpha : f \Rightarrow g$ such that $F\alpha = \beta$.



Equivalence between ω-categories

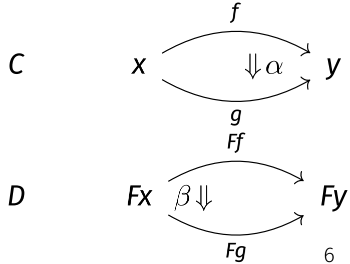
In an ω-category C, two cells $f, g : x \to y$ are **equivalent**, noted $f \sim g$, when there are cells

$$\alpha : f \Rightarrow g \qquad\qquad \beta : g \Rightarrow f$$

such that

$$\beta \circ \alpha \sim \mathrm{id}_f \qquad\qquad \alpha \circ \beta \sim \mathrm{id}_g$$

An ω*-functor* $F : C \to D$ is a **weak equivalence** when it is "surjective up to equivalence": given $f, g : x \to y$ and $\beta : Ff \Rightarrow Fg$ there is $\alpha : f \Rightarrow g$ such that $F\alpha = \beta$.



Equivalence between ω-categories

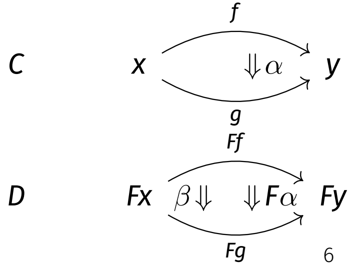
In an ω-category C, two cells $f, g : x \to y$ are **equivalent**, noted $f \sim g$, when there are cells

$$\alpha : f \Rightarrow g \qquad\qquad \beta : g \Rightarrow f$$

such that

$$\beta \circ \alpha \sim \mathrm{id}_f \qquad\qquad \alpha \circ \beta \sim \mathrm{id}_g$$

An *ω-functor* $F : C \to D$ is a **weak equivalence** when it is "surjective up to equivalence": given $f, g : x \to y$ and $\beta : Ff \Rightarrow Fg$ there is $\alpha : f \Rightarrow g$ such that $F\alpha = \beta$.



Equivalence between ω-categories

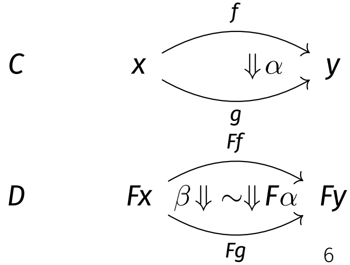
In an ω-category C, two cells $f, g : x \to y$ are **equivalent**, noted $f \sim g$, when there are cells

$$\alpha : f \Rightarrow g \qquad \beta : g \Rightarrow f$$

such that

$$\beta \circ \alpha \sim \mathrm{id}_f \qquad \alpha \circ \beta \sim \mathrm{id}_g$$

An *ω-functor* $F : C \to D$ is a **weak equivalence** when it is "surjective up to equivalence": given $f, g : x \to y$ and $\beta : Ff \Rightarrow Fg$ there is $\alpha : f \Rightarrow g$ such that $F\alpha = \beta$.



Resolutions of categories

In the 2003 article of François

Resolutions by polygraphs

it is shown that for every category C, there is a polygraph P such that

$$P \simeq C$$

Resolutions of categories

For instance, if we start from the category $B\mathbb{N}_2$

$$0 \circlearrowright \star \circlearrowleft 1$$

Resolutions of categories

For instance, if we start from the category $B\mathbb{N}_2$

$$0 \circlearrowright \star \circlearrowleft 1$$

we can start to construct P with a surjection $P \twoheadrightarrow C$ as

- one 0-cell $\star$
- one 1-cell $a : \star \to \star$

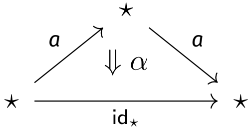
Resolutions of categories

For instance, if we start from the category $B\mathbb{N}_2$

$$0 \circlearrowleft \star \circlearrowright 1$$

we can start to construct P with a surjection $P \twoheadrightarrow C$ as

- one 0-cell $\star$
- one 1-cell $a : \star \to \star$
- one 2-cell



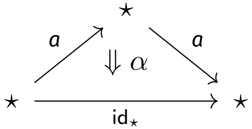
Resolutions of categories

For instance, if we start from the category $B\mathbb{N}_2$

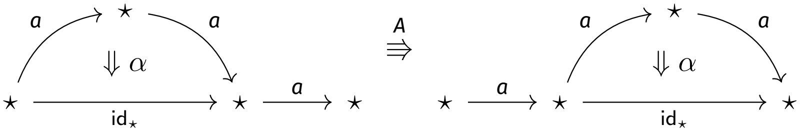
$$0 \circlearrowleft \star \circlearrowright 1$$

we can start to construct P with a surjection $P \twoheadrightarrow C$ as

- one 0-cell $\star$
- one 1-cell $a : \star \to \star$
- one 2-cell



- one 3-cell



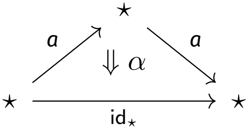
Resolutions of categories

For instance, if we start from the category $B\mathbb{N}_2$

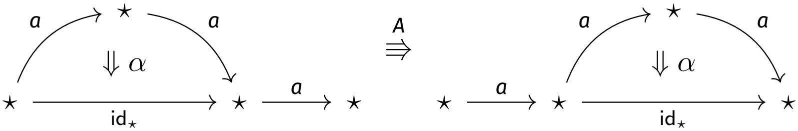
$$0 \circlearrowright \star \circlearrowleft 1$$

we can start to construct P with a surjection $P \twoheadrightarrow C$ as

- one 0-cell $\star$
- one 1-cell $a : \star \to \star$
- one 2-cell



- one 3-cell



- and so on...

Polygraphs as cofibrant replacements

In the 2010 article by Lafont, Métayer, Worytkiewicz,

A folk model structure on omega-cat

they construct a model structure on $\mathbf{Cat}_\omega$ where

- weak equivalences are weak equivalences
- cofibrant objects are polygraphs

In particular, every ω-category C has a cofibrant replacement

$$P \overset{\sim}{\twoheadrightarrow} C$$

Polygraphic homology

Every polygraph P induces a chain complex

$$\mathbb{Z}P_0 \xleftarrow{d_0} \mathbb{Z}P_1 \xleftarrow{d_1} \mathbb{Z}P_2 \xleftarrow{d_2} \mathbb{Z}P_3 \xleftarrow{d_3} \cdots$$

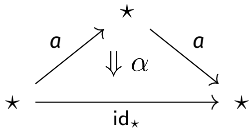
with our example

$$\mathbb{Z}\{\star\} \xleftarrow{d_0} \mathbb{Z}\{a\} \xleftarrow{d_1} \mathbb{Z}\{\alpha\} \xleftarrow{d_2} \mathbb{Z}\{A\} \xleftarrow{d_3} \cdots$$

with

$$d_1(\alpha) = -2a$$

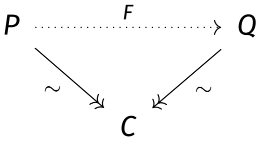
since



Polygraphic homology

We define the homology of an ω-category HC as the homology of the associated chain complex $H(\mathbb{Z}P)$ for some resolution P.

It does not depend on the choice of the resolution:



that's the point of using polygraphs!

However, the smaller the polygraph is, the simpler the calculations are!

Thomason equivalences

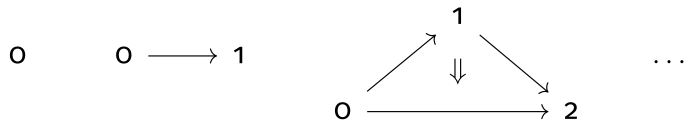
In 1987, Street has defined a functor

$$O : \Delta \to \mathbf{Cat}_\omega$$

reworked by Ara, Lafont and Métayer in 2023 in

Orientals as free algebras

The images of objects n of Δ can be pictured as



O_0 $\qquad O_1$ $\qquad O_2$ $\qquad \dots$

Thomason equivalences

In 1987, Street has defined a functor

$$O : \Delta \to \mathbf{Cat}_\omega$$

reworked by Ara, Lafont and Métayer in 2023 in

Orientals as free algebras

which induces a **nerve** functor

$$N : \mathbf{Cat}_\omega \to \hat{\Delta}$$

with

$$(NC)_n = \mathbf{Cat}_\omega(O_n, C)$$

The **Thomason equivalences** are induced on $\mathbf{Cat}_\omega$ by the one of $\hat{\Delta}$.

Ara's counter-example

Consider the polygraph P with

- one 0-generator $\star$
- one 2-generator $\alpha : \mathrm{id}_\star \Rightarrow \mathrm{id}_\star$

Its homology is the one of the chain complex

$$\mathbb{Z} \longleftarrow 0 \longleftarrow \mathbb{Z} \longleftarrow 0 \longleftarrow 0 \longleftarrow \cdots$$

In particular,

$$H_n(P) = 0$$

for $n > 2$.

Ara's counter-example

The ω-category C presented by P is trivial excepting in dimension 2 where

$$C_2 = \mathbb{N}$$

Ara's counter-example

The ω-category C presented by P is trivial excepting in dimension 2 where

$$C_2 = \mathbb{N}$$

It Thomason homology is the same as the one where we take the variant where

$$C_2 = \mathbb{Z}$$

Ara's counter-example

The ω-category C presented by P is trivial excepting in dimension 2 where

$$C_2 = \mathbb{N}$$

It Thomason homology is the same as the one where we take the variant where

$$C_2 = \mathbb{Z}$$

which is the homology of $K(\mathbb{Z}, 2) = \mathbb{CP}^\infty$ and we have

$$H_n^{\mathrm{Th}}(C) = \begin{cases} \mathbb{Z} & \text{for } n \text{ even} \\ 0 & \text{for } n \text{ odd} \end{cases}$$

Ara's counter-example

The ω-category C presented by P is trivial excepting in dimension 2 where

$$C_2 = \mathbb{N}$$

It Thomason homology is the same as the one where we take the variant where

$$C_2 = \mathbb{Z}$$

which is the homology of $K(\mathbb{Z}, 2) = \mathbb{CP}^\infty$ and we have

$$H_n^{\mathrm{Th}}(C) = \begin{cases} \mathbb{Z} & \text{for } n \text{ even} \\ 0 & \text{for } n \text{ odd} \end{cases}$$

In particular,

$$H_4^{\mathrm{Pol}}(C) = 0 \neq \mathbb{Z} = H_4^{\mathrm{Th}}(C)$$

Part II

Polygraphs in HoTT

Weak polygraphs

The morale is that polygraphs are not right from a topological point of view.

Weak polygraphs

The morale is that polygraphs are not right from a topological point of view.

Or rather working with *strict* ω-categories is a bad idea.

Weak polygraphs

The morale is that polygraphs are not right from a topological point of view.

Or rather working with *strict* ω-categories is a bad idea.

One way out consists in working with polygraphs adapted to *weak* ω*-categories* (Batanin) which is very technical.

Polygraphs for groupoids

Suppose that we are interested in ∞-**groupoids** instead of ∞-categories.

We get everything for “free” in **homotopy type theory**:

- every type is an ∞-groupoid
- polygraphs can be obtained as (some) higher inductive types

Homotopy type theory

Given a type A and two elements $x, y : A$, there is a type $x =^A y$ of **identities** between x and y.

We can think that

- A is a space
- $x, y : A$ are points in A
- $p : x =^A y$ is a path from x to y in A

Let's do a crash course in 2 slides.

Identity types

The identity types are characterized the fact that

- for every $x : A$, there is a constant path $\mathsf{refl}_x : x = x$
- given a predicate $P : \{y : A\} \to (x = y) \to \mathcal{U}$,
  if $P(\mathsf{refl}_x)$ then $P(p)$ for every $p : x = y$.

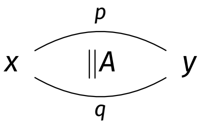
Identity types

The identity types are characterized the fact that

- for every $x : A$, there is a constant path $\mathsf{refl}_x : x = x$
- given a predicate $P : \{y : A\} \to (x = y) \to \mathcal{U}$,
  if $P(\mathsf{refl}_x)$ then $P(p)$ for every $p : x = y$.

Because of this types act as ∞-groupoids:

- 0-cells are points $x, y : A$
- 1-cells are paths $p : x =^A y$
- 2-cells are paths between paths $\alpha : p =^{x =^A y} q$
- etc.



Identity types

The identity types are characterized the fact that

- for every $x : A$, there is a constant path $\mathsf{refl}_x : x = x$
- given a predicate $P : \{y : A\} \to (x = y) \to \mathcal{U}$,
  if $P(\mathsf{refl}_x)$ then $P(p)$ for every $p : x = y$.

Lemma
Given a path $p : x = y$, there is an "inverse" path $\overline{p} : y = x$.

Proof.
In the case where p is refl_x (thus x and y are the same), we can take $\overline{p} := \mathsf{refl}_x$. □

Similarly, we can compose paths, composition is associative up to higher cells, etc.

Equivalences

Given $f, g : A \to B$, we write $f \sim g$ when they are **extensionally equivalent**:

$$(x \ : \ A) \to f(x) = g(x)$$

Equivalences

Given $f, g : A \to B$, we write $f \sim g$ when they are **extensionally equivalent**:

$$(x \ : \ A) \to f(x) = g(x)$$

A map $f : A \to B$ is an equivalence when there is $g : B \to A$ such that

- there are homotopies

$$\eta : g \circ f \sim \mathsf{id}_A \qquad\qquad \varepsilon : f \circ g \sim \mathsf{id}_B$$

- such that $(x \ : \ A) \to f\ (\eta\ x) = \varepsilon\ (g\ x)$

Equivalences

Given $f, g : A \to B$, we write $f \sim g$ when they are **extensionally equivalent**:

$$(x \ : \ A) \to f(x) = g(x)$$

A map $f : A \to B$ is an equivalence when there is $g : B \to A$ such that

- there are homotopies

$$\eta : g \circ f \sim \mathrm{id}_A \qquad\qquad \varepsilon : f \circ g \sim \mathrm{id}_B$$

- such that $(x \ : \ A) \to f\ (\eta\ x) = \varepsilon\ (g\ x)$

In homotopy type theory, **univalence** states that an identity $A = B$ is the same as an equivalence between A and B:

$$(A = B) \xrightarrow{\sim} (A \simeq B)$$

Resolutions in HoTT

Those equivalence play an analogous role as weak equivalence for ω-categories.

Resolutions in HoTT

Those equivalence play an analogous role as weak equivalence for ω-categories.

Given a type A of interest, our goal is to construct a **resolution**,
i.e. a polygraph P such that

$$P \simeq A$$

…for a decent notion of “polygraph”.

Inductive types

In good programming languages, we can define inductive types:

```
type Bool : Type =
  false : Bool
  true  : Bool
```

Inductive types

In good programming languages, we can define inductive types:

```
type Nat : Type =
  zero : Nat
  succ : Nat → Nat
```

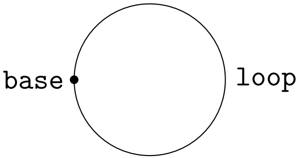
Inductive types

In good programming languages, we can define inductive types:

```
type Nat : Type =
  zero : Nat
  succ : Nat → Nat
```

In good type theories, we can define **higher inductive types**:

```
type S¹ : Type =
  base : S¹
  loop : base = base
```



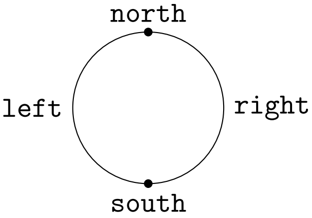
Inductive types

In good programming languages, we can define inductive types:

```
type Nat : Type =
  zero : Nat
  succ : Nat → Nat
```

In good type theories, we can define **higher inductive types**:

```
type S¹ : Type =
  north : S¹
  south : S¹
  left  : base = base'
  right : base = base'
```



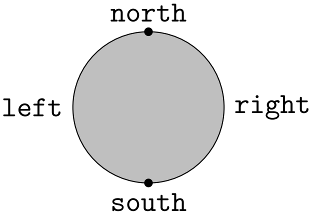
Inductive types

In good programming languages, we can define inductive types:

```
type Nat : Type =
  zero : Nat
  succ : Nat → Nat
```

In good type theories, we can define **higher inductive types**:

```
type S² : Type =
  north : S²
  south : S²
  left  : base = base'
  right : base = base'
  back  : left = right
  front : left = right
```



Inductive types

In good programming languages, we can define inductive types:

```
type Nat : Type =
  zero : Nat
  succ : Nat → Nat
```

In good type theories, we can define **higher inductive types**:

```
type id : A → A → Type =
  refl : (x : A) → id x x
```

Higher inductive types

Since HITs are obtained by successively attaching disks, they play an analogous role of to **polygraphs** or **cellular complexes**.

Higher inductive types

Since HITs are obtained by successively attaching disks, they play an analogous role of to **polygraphs** or **cellular complexes**.

At least some of them.

Homotopy levels

A type A can be

-2. contractible:

$$\mathsf{isContr}(A) := \Sigma(x : A).(y : A) \to x =^A y$$

Homotopy levels

A type A can be

-2. contractible:

$$\mathrm{isContr}(A) := \Sigma(x : A).(y : A) \to x =^A y$$

-1. a proposition:

$$\mathrm{isProp}(A) := (x\ y : A) \to \mathrm{isContr}(x =^A y)$$

Homotopy levels

A type A can be

-2. contractible:

$$\text{isContr}(A) := \Sigma(x : A).(y : A) \to x =^A y$$

-1. a proposition:

$$\text{isProp}(A) := (x\ y : A) \to \text{isContr}(x =^A y)$$

0. a set:

$$\text{isSet}(A) := (x\ y : A) \to \text{isProp}(x =^A y)$$

Homotopy levels

A type A can be

-2. contractible:

$$\mathsf{isContr}(A) :\equiv \Sigma(x : A).(y : A) \to x =^A y$$

-1. a proposition:

$$\mathsf{isProp}(A) :\equiv (x\ y : A) \to \mathsf{isContr}(x =^A y)$$

0. a set:

$$\mathsf{isSet}(A) :\equiv (x\ y : A) \to \mathsf{isProp}(x =^A y)$$

n. a n-type: for every $x, y : A$, $x =^A y$ is an $(n-1)$-type

Propositional truncation

The **propositional truncation** $\|A\|_{-1}$ turns a type A into a proposition in a universal way: for every *proposition* B,

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle t}\downarrow & \nearrow_{\tilde{f}} & \\ \|A\|_{-1} & & \end{array}$$

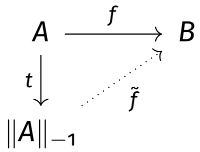
i.e. the map

$$- \circ t : (\|A\|_{-1} \to B) \to (A \to B)$$

is an equivalence.

Propositional truncation

The **propositional truncation** $\|A\|_{-1}$ turns a type A into a proposition in a universal way: for every *proposition* B,



i.e. the map

$$- \circ t : (\|A\|_{-1} \to B) \to (A \to B)$$

is an equivalence.

The ***n*-truncation** $\|A\|_n$ can be defined similarly for n-types.

Propositional truncation

Propositional truncation can be implemented as a HIT:

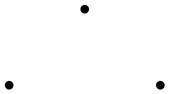
```
type ‖A‖₋₁ : 𝒰 =
  in   : A → ‖A‖₋₁
  path : (x y : ‖A‖₋₁) → x = y
```

Propositional truncation

Propositional truncation can be implemented as a HIT:

$$
\begin{aligned}
&\texttt{type}\ \|A\|_{-1} : \mathcal{U}\ \texttt{=} \\
&\quad \texttt{in}\quad\ \ :\ A \to \|A\|_{-1} \\
&\quad \texttt{path}\ :\ (x\,y\ :\ \|A\|_{-1}) \to x = y
\end{aligned}
$$

For instance,

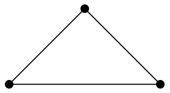


Propositional truncation

Propositional truncation can be implemented as a HIT:

$$
\begin{aligned}
&\texttt{type } \|A\|_{-1} : \mathcal{U} \texttt{ =} \\
&\quad \texttt{in} \quad : A \to \|A\|_{-1} \\
&\quad \texttt{path} : (x\, y : \|A\|_{-1}) \to x = y
\end{aligned}
$$

For instance,

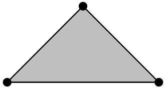


Propositional truncation

Propositional truncation can be implemented as a HIT:

$$\texttt{type}\ \|A\|_{-1} : \mathcal{U}\ \texttt{=}$$
$$\texttt{in}\ :\ A \to \|A\|_{-1}$$
$$\texttt{path}\ :\ (x\,y\ :\ \|A\|_{-1}) \to x = y$$

For instance,

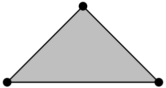


Propositional truncation

Propositional truncation can be implemented as a HIT:

```
type ‖A‖₋₁ : 𝒰 =
  in   : A → ‖A‖₋₁
  path : (x y : ‖A‖₋₁) → x = y
```

For instance,



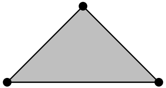
We can similarly define higher truncations $\|A\|_n$:
it fills in all spheres of dimension $k > n$.

Propositional truncation

Propositional truncation can be implemented as a HIT:

$$\begin{aligned}&\texttt{type}\ \|A\|_{-1} : \mathcal{U}\ \texttt{=}\\ &\quad \texttt{in}\quad : A \to \|A\|_{-1}\\ &\quad \texttt{path} : (x\,y : \|A\|_{-1}) \to x = y\end{aligned}$$

For instance,



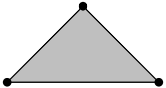
Problem solved?

Propositional truncation

Propositional truncation can be implemented as a HIT:

$$\begin{aligned}
&\texttt{type}\ \|A\|_{-1} : \mathcal{U}\ \texttt{=} \\
&\quad \texttt{in}\quad : A \to \|A\|_{-1} \\
&\quad \texttt{path} : (x\ y\ :\ \|A\|_{-1}) \to x = y
\end{aligned}$$

For instance,



This is a **recursive** HIT, we do not want this as a "polygraph".

Part III

The rewriting approach

Presenting $B\mathbb{Z}_2$

Consider the type $B\mathbb{Z}_2$. All we need to know is that

- it has one connected component
- its fundamental group is $\pi_1(B\mathbb{Z}_2) = \mathbb{Z}_2$
- its higher homotopy groups are trivial: $\pi_n(B\mathbb{Z}_2) = 1$

$$0 \circlearrowright \star \circlearrowleft 1$$

Suppose that we want to construct a presentation of this type by a polygraph.

Presenting $B\mathbb{Z}_2 = 0 \circlearrowleft \star \circlearrowright 1$

We thus define the following HIT:

$$\texttt{type } P^1 : \mathcal{U} \texttt{ =}$$

which looks like

Presenting $B\mathbb{Z}_2 = \; 0 \circlearrowright \; \star \; \circlearrowleft 1$

We thus define the following HIT:

$$\texttt{type}\ P^1 : \mathcal{U}\ \texttt{=}$$
$$\star : P^1$$

which looks like

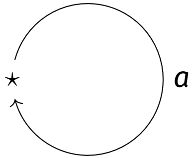
$\star$

Presenting $B\mathbb{Z}_2 = 0 \circlearrowright \star \circlearrowleft 1$

We thus define the following HIT:

$$\texttt{type}\ P^1 : \mathcal{U}\ =$$
$$\star : P^1$$
$$a : \star = \star$$

which looks like

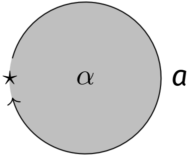


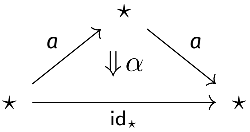
Presenting $B\mathbb{Z}_2 = 0 \circlearrowright \star \circlearrowleft 1$

We thus define the following HIT:

$$
\begin{aligned}
&\texttt{type}\ P^1 : \mathcal{U}\ \texttt{=} \\
&\quad \star : P^1 \\
&\quad a : \star = \star \\
&\quad \alpha : a \cdot a = \mathrm{refl}_\star
\end{aligned}
$$

which looks like



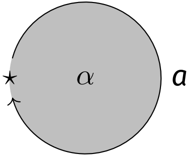


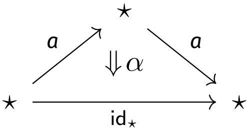
Presenting $B\mathbb{Z}_2 = 0 \circlearrowleft \star \circlearrowright 1$

We thus define the following HIT:

$$\begin{aligned}&\texttt{type } P^1 : \mathcal{U} \texttt{ =}\\ &\quad \star : P^1\\ &\quad a : \star = \star\\ &\quad \alpha : a \cdot a = \mathsf{refl}_\star\end{aligned}$$

which looks like





P^1 is a "good approximation" of $B\mathbb{Z}_2$ in the sense that it has one connected component and the right fundamental group, but higher groups are not trivial!

Killing higher groups

One way to obtain a faithful description of $B\mathbb{Z}_2$ consists in considering $\|P^1\|_1$, which amounts to change the definition to

$$
\begin{aligned}
&\texttt{type } Q^1 : \mathcal{U} \texttt{ =} \\
&\quad \star : Q^1 \\
&\quad a : \star = \star \\
&\quad \alpha : a \cdot a = \mathrm{refl}_\star \\
&\quad \texttt{trunc} : (x, y : Q^1)\ (p, q : x = y)\ (\alpha, \beta : p = q) \to \alpha = \beta
\end{aligned}
$$

Killing higher groups

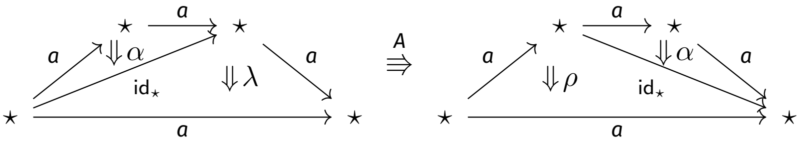
One way to obtain a faithful description of $B\mathbb{Z}_2$ consists in considering $\|P^1\|_1$, which amounts to change the definition to

$$
\begin{aligned}
&\texttt{type } Q^1 : \mathcal{U} \texttt{ =}\\
&\quad \star : Q^1\\
&\quad a : \star = \star\\
&\quad \alpha : a \cdot a = \mathrm{refl}_\star\\
&\quad \texttt{trunc} : (x, y : Q^1)\ (p, q : x = y)\ (\alpha, \beta : p = q) \to \alpha = \beta
\end{aligned}
$$

But this is a recursive definition!

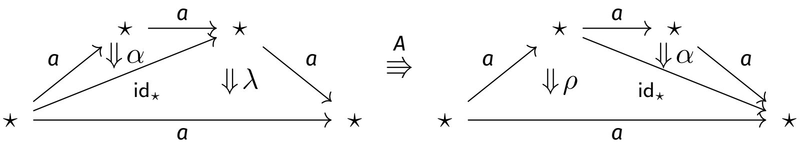
Killing $\pi_2(P^1)$

A non-trivial element of $\pi_2(P^1)$ is



Killing $\pi_2(P^1)$

A non-trivial element of $\pi_2(P^1)$ is

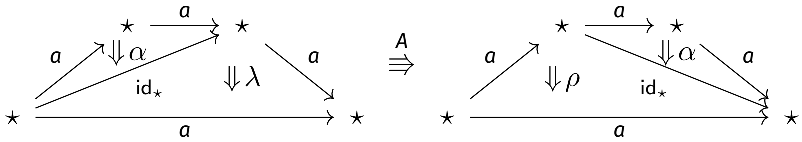


This suggests extending the previous HIT as

$$
\begin{aligned}
&\texttt{type } P^2 : \mathcal{U} \texttt{ =} \\
&\quad \star : P^2 \\
&\quad a : \star = \star \\
&\quad \alpha : a \cdot a = \mathrm{refl}_\star \\
&\quad A : (\alpha \otimes a) \cdot \lambda = (a \otimes \alpha) \cdot \rho
\end{aligned}
$$

Killing $\pi_2(P^1)$

A non-trivial element of $\pi_2(P^1)$ is



This suggests extending the previous HIT as

```
type P² : U =
```

$\star : P^2$

$a : \star = \star$

$\alpha : a \cdot a = \mathsf{refl}_\star$

$A : (\alpha \otimes a) \cdot \lambda = (a \otimes \alpha) \cdot \rho$

This is enough to have $\pi_2(P^2) = 1$, but how do we show this?
We need to have an induction principle for paths!

Killing $\pi_2(P^1)$

Our aim is to show that

$$\pi_2(P^2) = 1$$

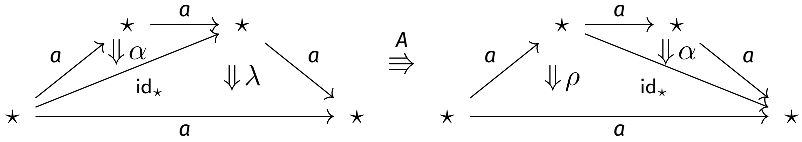
This is equivalent to showing that

- for every paths $p, q : \star = \star$
- for every paths $\alpha, \beta : p = q$

we have that there merely exists a path

$$A : \alpha = \beta$$

as in



Paths in homotopy quotients

Suppose given a type A with a relation $R : A \times A \to \mathcal{U}$, i.e. a **graph**.

Paths in homotopy quotients

Suppose given a type A with a relation $R : A \times A \to \mathcal{U}$, i.e. a **graph**.

The **homotopy quotient** A/R is

$$
\begin{aligned}
&\texttt{type}\ A/R : \mathcal{U}\ \texttt{=} \\
&\quad \iota : A \to A/R \\
&\quad \texttt{path} : (x\ y\ :\ A) \to R\ x\ y \to x = y
\end{aligned}
$$

Paths in homotopy quotients

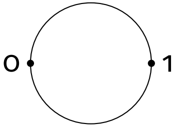
Suppose given a type A with a relation $R : A \times A \to \mathcal{U}$, i.e. a **graph**.

The **homotopy quotient** A/R is

$$
\begin{aligned}
&\texttt{type } A/R : \mathcal{U} \ \texttt{=} \\
&\quad \iota : A \to A/R \\
&\quad \texttt{path} : (x\, y\ :\ A) \to R\, x\, y \to x = y
\end{aligned}
$$

For instance, with $A = \{0, 1\}$ and $R\, x\, y := (x \neq y)$, we have

$$A/R = S^1 =$$



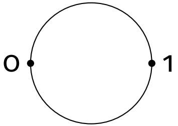
Paths in homotopy quotients

Suppose given a type A with a relation $R : A \times A \to \mathcal{U}$, i.e. a **graph**.

The **homotopy quotient** A/R is

$$
\begin{aligned}
&\texttt{type}\ A/R : \mathcal{U}\ \texttt{=} \\
&\quad \iota : A \to A/R \\
&\quad \texttt{path} : (x\ y\ :\ A) \to R\ x\ y \to x = y
\end{aligned}
$$

For instance, with $A = \{0, 1\}$ and $R\ x\ y := (x \neq y)$, we have

$A/R = S^1 =$ 

Given $x, y : A$, we want to have a description of the type $\iota\ x = \iota\ y$ in A/R.

Paths in homotopy quotients

Given A and $R : A \times A \to \mathcal{U}$, we define the **free groupoid** type

```
type FG : A → A → U =
```

Altenkirch, Kraus, von Raummer have shown

Theorem
Given $x, y : A$, *we have* $(\iota\, x = \iota\, y) = FG\; x\; y$.

Paths in homotopy quotients

Given A and $R : A \times A \to \mathcal{U}$, we define the **free groupoid** type

$$\texttt{type}\ FG\ :\ A \to A \to \mathcal{U}\ \texttt{=}$$
$$\texttt{[]} : (x : A) \to FG\ x\ x$$
$$\_\texttt{::}\_ : (f\ :\ R\ x\ y) \to (p\ :\ FG\ y\ z) \to FG\ x\ z$$

Altenkirch, Kraus, von Raummer have shown

Theorem
Given $x, y : A$, we have $(\iota\ x = \iota\ y) = FG\ x\ y$.

Paths in homotopy quotients

Given A and $R : A \times A \to \mathcal{U}$, we define the **free groupoid** type

```
type FG : A → A → 𝒰 =
   [] : (x : A) → FG x x
   _::_ : (f : R x y) → (p : FG y z) → FG x z
   _::'_ : (f : R y x) → (p : FG y z) → FG x z
```

Altenkirch, Kraus, von Raummer have shown

Theorem
Given $x, y : A$*, we have* $(\iota\, x = \iota\, y) = FG\ x\ y$.

Paths in homotopy quotients

Given A and $R : A \times A \to \mathcal{U}$, we define the **free groupoid** type

$$
\begin{aligned}
&\texttt{type}\ FG\ :\ A \to A \to \mathcal{U}\ \texttt{=}\\
&\quad \texttt{[]} : (x : A) \to FG\ x\ x\\
&\quad \_\texttt{::}\_ : (f\ :\ R\ x\ y) \to (p\ :\ FG\ y\ z) \to FG\ x\ z\\
&\quad \_\texttt{::'}\_ : (f\ :\ R\ y\ x) \to (p\ :\ FG\ y\ z) \to FG\ x\ z\\
&\quad \lambda : (f\ :\ R\ x\ y) \to (p\ :\ FG\ y\ z) \to f\texttt{::'}f\texttt{::}p = p\\
&\quad \rho : (f\ :\ R\ x\ y) \to (p\ :\ FG\ y\ z) \to f\texttt{::'}f\texttt{::}p = p
\end{aligned}
$$

Altenkirch, Kraus, von Raummer have shown

Theorem
Given $x, y : A$, we have $(\iota\, x = \iota\, y) = FG\ x\ y$.

Paths in homotopy quotients

Given A and $R : A \times A \to \mathcal{U}$, we define the **free groupoid** type

$$
\begin{aligned}
&\texttt{type}\ \ FG\ :\ A \to A \to \mathcal{U}\ \ \texttt{=}\\
&\quad \texttt{[]} : (x : A) \to FG\ x\ x\\
&\quad \_\texttt{::}\_ : (f\ :\ R\ x\ y) \to (p\ :\ FG\ y\ z) \to FG\ x\ z\\
&\quad \_\texttt{::'}\_ : (f\ :\ R\ y\ x) \to (p\ :\ FG\ y\ z) \to FG\ x\ z\\
&\quad \lambda : (f\ :\ R\ x\ y) \to (p\ :\ FG\ y\ z) \to f\texttt{::'}f\texttt{::}p = p\\
&\quad \rho : (f\ :\ R\ x\ y) \to (p\ :\ FG\ y\ z) \to f\texttt{::'}f\texttt{::}p = p\\
&\quad \texttt{coh} : \ldots
\end{aligned}
$$

Altenkirch, Kraus, von Raummer have shown

Theorem
Given $x, y : A$, we have $(\iota\ x = \iota\ y) = FG\ x\ y$.

Paths in homotopy quotients

If we want to show a *property* on paths, for instance

$$\textit{for every paths } p, q : x = y \textit{ there merely exists an equality } p = q$$

it is enough to reason on the low-dimensional structure, i.e. zig-zags:

$$\begin{aligned}
&\texttt{type}\ FG\ :\ A \to A \to \mathcal{U}\ \texttt{=} \\
&\quad \texttt{[]} : (x : A) \to FG\ x\ x \\
&\quad _\texttt{::}_ : (f\ :\ R\ x\ y) \to (p\ :\ FG\ y\ z) \to FG\ x\ z \\
&\quad _\texttt{::'}_ : (f\ :\ R\ y\ x) \to (p\ :\ FG\ y\ z) \to FG\ x\ z
\end{aligned}$$

and we can reason by induction.

Paths in homotopy quotients

When reasoning with “lists” (or “zig-zags”) in the type

$$\texttt{type}\ FG : A \to A \to \mathcal{U}\ \texttt{=}$$
$$\texttt{[]} : (x : A) \to FG\ x\ x$$
$$\_::\_ : (f : R\ x\ y) \to (p : FG\ y\ z) \to FG\ x\ z$$
$$\_::'\_ : (f : R\ y\ x) \to (p : FG\ y\ z) \to FG\ x\ z$$

there is one problem with the base case: the elements of $FG\ x\ x$ of length 0 is equivalent to $x = x$, i.e. there can be more than simply [].

Things work out if we suppose that A is a set.

A coherent presentation

If we go back to the type

$$
\begin{aligned}
&\texttt{type } P^2 : \mathcal{U} \texttt{ =} \\
&\quad \star : P^2 \\
&\quad a : \star = \star \\
&\quad \alpha : a \cdot a = \mathsf{refl}_\star \\
&\quad A : (\alpha \otimes a) \cdot \lambda = (a \otimes \alpha) \cdot \rho
\end{aligned}
$$

This implies that

- any path $\star = \star$ has a representative as a list over $\{a, \overline{a}\}$

A coherent presentation

If we go back to the type

$$
\begin{aligned}
&\texttt{type } P^2 : \mathcal{U} \texttt{ =} \\
&\quad \star : P^2 \\
&\quad a : \star = \star \\
&\quad \alpha : a \cdot a = \mathrm{refl}_\star \\
&\quad A : (\alpha \otimes a) \cdot \lambda = (a \otimes \alpha) \cdot \rho
\end{aligned}
$$

This implies that

- any path $\star = \star$ has a representative as a list over $\{a, \overline{a}\}$
- identities between two such lists are generated by

$$laal' = ll' \qquad l\overline{a}al' = ll' \qquad la\overline{a}l' = ll'$$

A coherent presentation

If we go back to the type

$$
\begin{aligned}
&\texttt{type } P^2 : \mathcal{U} \texttt{ =}\\
&\quad \star : P^2\\
&\quad a : \star = \star\\
&\quad \alpha : a \cdot a = \mathrm{refl}_\star\\
&\quad A : (\alpha \otimes a) \cdot \lambda = (a \otimes \alpha) \cdot \rho
\end{aligned}
$$

This implies that

- any path $\star = \star$ has a representative as a list over $\{a, \overline{a}\}$
- identities between two such lists are generated by

$$
la a l' = l l' \qquad l\overline{a} a l' = l l' \qquad l a \overline{a} l' = l l'
$$

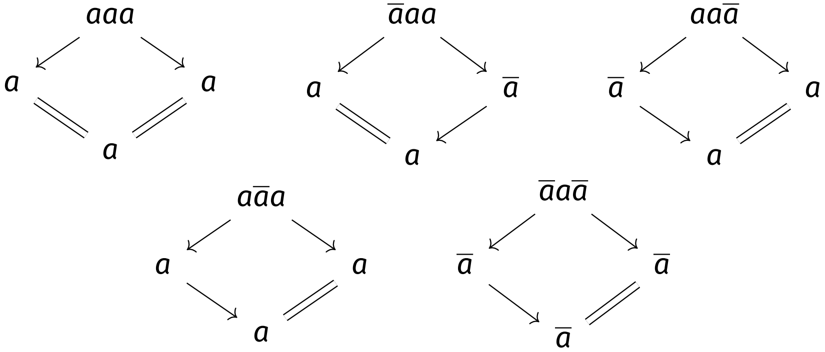
- identities between identities are generated by A and `coh`

A coherent presentation

It can be noted that the string rewriting system over $\{a, \overline{a}\}$ with rules

$$aa \to 1 \qquad \overline{a}a \to 1 \qquad a\overline{a} \to 1 \qquad \overline{a} \to a$$

is convergent



and those coherence correspond to identities between identities.

A coherent presentation

By the **Squier theorem** (with polygraphs implemented in type theory!),

any two zig-zags can thus be filled by identities between identities and we deduce (Kraus, von Raummer) that in the type

$$
\begin{aligned}
&\texttt{type } P^2 : \mathcal{U} \texttt{ =}\\
&\quad \star : P^2\\
&\quad a : \star = \star\\
&\quad \alpha : a \cdot a = \mathrm{refl}_\star\\
&\quad \beta : \overline{a} = a\\
&\quad A : (\alpha \otimes a) \cdot \lambda = (a \otimes \alpha) \cdot \rho\\
&\quad B : \ldots\\
&\quad C : \ldots
\end{aligned}
$$

we have $\pi_2(P^2) = 1$.

A coherent presentation

By the **Squier theorem** (with polygraphs implemented in type theory!),

any two zig-zags can thus be filled by identities between identities and we deduce (Kraus, von Raummer) that in the type

$$
\begin{aligned}
&\texttt{type } P^2 : \mathcal{U} \ \texttt{=} \\
&\quad \star : P^2 \\
&\quad a : \star = \star \\
&\quad \alpha : a \cdot a = \mathsf{refl}_\star \\
\\
&\quad A : (\alpha \otimes a) \cdot \lambda = (a \otimes \alpha) \cdot \rho \\
&\quad B : \ldots
\end{aligned}
$$

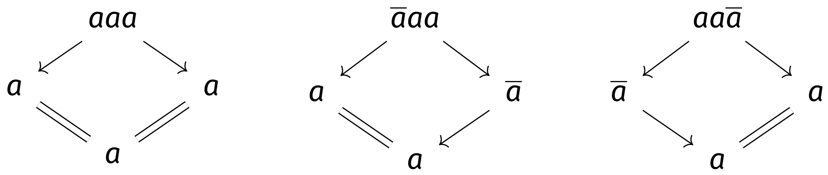
we have $\pi_2(P^2) = 1$.

The Squier theorem

$$aa \to 1 \qquad \bar{a}a \to 1 \qquad a\bar{a} \to 1 \qquad \bar{a} \to a$$

We can show the following:

1. the local confluence diagrams commute modulo equality

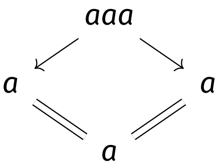


The Squier theorem

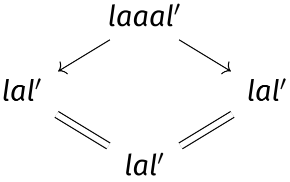
$$aa \to 1 \qquad \overline{a}a \to 1 \qquad a\overline{a} \to 1 \qquad \overline{a} \to a$$

We can show the following:

1. the local confluence diagrams commute modulo equality
2. the local confluence diagrams can be extended under context



implies



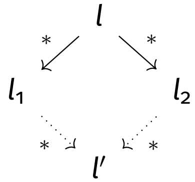
The Squier theorem

$$aa \to 1 \qquad \bar{a}a \to 1 \qquad a\bar{a} \to 1 \qquad \bar{a} \to a$$

We can show the following:

1. the local confluence diagrams commute modulo equality
2. the local confluence diagrams can be extended under context
3. the rewriting system is terminating, and thus we have confluence

(modulo equality!)

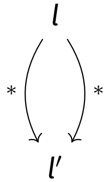


The Squier theorem

$$aa \to 1 \qquad \bar{a}a \to 1 \qquad a\bar{a} \to 1 \qquad \bar{a} \to a$$

We can show the following:

1. the local confluence diagrams commute modulo equality
2. the local confluence diagrams can be extended under context
3. the rewriting system is terminating, and thus we have confluence
4. any two parallel zig-zags are equal



The Squier theorem

$$aa \to 1 \qquad \overline{a}a \to 1 \qquad a\overline{a} \to 1 \qquad \overline{a} \to a$$

We can show the following:

1. the local confluence diagrams commute modulo equality
2. the local confluence diagrams can be extended under context
3. the rewriting system is terminating, and thus we have confluence
4. any two parallel zig-zags are equal

We thus have

$$\pi_2(P^2) = 1$$

What we have so far

We have constructed a type P^2 which has the same π_0, π_1 and π_2 as $B\mathbb{Z}_2$.

What we have so far

We have constructed a type P^2 which has the same π_0, π_1 and π_2 as $B\mathbb{Z}_2$.

Extending P^2 into P^3 which also has the same π_3 could certainly be done, but this would require tremendous amounts of work!

What we have so far

We have constructed a type P^2 which has the same π_0, π_1 and π_2 as $B\mathbb{Z}_2$.

Extending P^2 into P^3 which also has the same π_3 could certainly be done, but this would require tremendous amounts of work!

Not to mention P^4, P^n, or P^∞...

What we have so far

We have constructed a type P^2 which has the same π_0, π_1 and π_2 as $B\mathbb{Z}_2$.

Extending P^2 into P^3 which also has the same π_3 could certainly be done, but this would require tremendous amounts of work!

Not to mention P^4, P^n, or P^∞...

Also, because of the previous remark, we are forced to reason with *set-theoretic* polygraphs, where we have *sets* of cells.

Part IV

The Milnor construction

Some other methods allow us to construct P^∞ such that

$$P^\infty = B\mathbb{Z}_2$$

Projective spaces

In algebraic topology, there is a well-known model of $B\mathbb{Z}_2$, the **real projective space** $\mathbb{RP}^\infty$.

Projective spaces

In algebraic topology, there is a well-known model of $B\mathbb{Z}_2$,
the **real projective space** $\mathbb{RP}^\infty$.

We know that it has a CW structure with one cell in every dimension.

There should be a corresponding HIT. How can we construct it?

Projective spaces

In algebraic topology, there is a well-known model of $B\mathbb{Z}_2$, the **real projective space** $\mathbb{RP}^\infty$.

We know that it has a CW structure with one cell in every dimension.

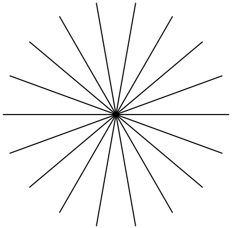
There should be a corresponding HIT. How can we construct it?

There is a wonderful construction based on the **join** construction, due to Milnor, Rijke, Finster, Joyal, …

…and the construction of orientals by Ara, Lafont and Métayer in *Orientals as free algebras* is closely related to the join construction.

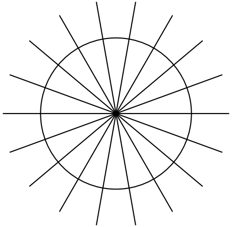
Projective spaces

The **projective space P^n** is the space of lines in $\mathbb{R}^{n+1}$:



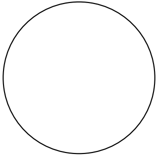
Projective spaces

The **projective space** P^n is the space of lines in $\mathbb{R}^{n+1}$:



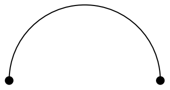
Projective spaces

The **projective space** P^n is the quotient of S^n under the antipodal action:



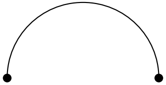
Projective spaces

The **projective space** P^n is a disk D^n with antipodal points identified in ∂D^n:



Projective spaces

The **projective space** P^n is a disk D^n with antipodal points identified in ∂D^n:

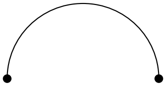


We thus have a pushout

$$\begin{array}{ccc} S^0 & \longrightarrow & D^1 \\ {\scriptstyle p^0}\downarrow & & \downarrow \\ P^0 & \longrightarrow & P^1 \end{array}$$

Projective spaces

The **projective space** P^n is a disk D^n with antipodal points identified in ∂D^n:

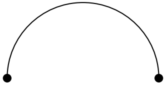


We thus have a pushout

$$\begin{array}{ccc} S^n & \longrightarrow & D^{n+1} \\ {\scriptstyle p^0}\downarrow & & \downarrow \\ P^n & \longrightarrow & P^{n+1} \end{array}$$

Projective spaces

The **projective space** P^n is a disk D^n with antipodal points identified in ∂D^n:

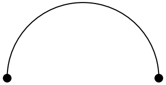


We thus have a pushout

$$\begin{array}{ccc} S^n & \longrightarrow & 1 \\ {\scriptstyle p^n}\big\downarrow & \ulcorner & \big\downarrow \\ P^n & \longrightarrow & P^{n+1} \end{array}$$

Projective spaces

The **projective space** P^n is a disk D^n with antipodal points identified in ∂D^n:



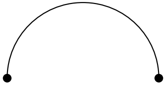
We thus have a pushout

$$\begin{array}{ccc} S^n & \longrightarrow & 1 \\ {\scriptstyle p^n}\downarrow & & \downarrow \\ P^n & \longrightarrow & P^{n+1} \end{array}$$

and P^n is build from exactly one cell in each dimension $i \leq n$.

Projective spaces

The **projective space** P^n is a disk D^n with antipodal points identified in ∂D^n:



We thus have a pushout

$$\begin{array}{ccc} S^n & \longrightarrow & 1 \\ {\scriptstyle p^n}\downarrow & & \downarrow \\ P^n & \longrightarrow & P^{n+1} \end{array}$$

and P^n is build from exactly one cell in each dimension $i \leq n$.

Moreover, there are exactly two points y such that $p^n(y) = x$, i.e. $\mathrm{fib}_{p^n}(x) = S^0$.

The join construction

Given two types A and B, their **coproduct** $A \sqcup B$ is

$$\begin{array}{ccc} & & B \\ & & \downarrow \iota_2 \\ A & \xrightarrow{\iota_1} & A \sqcup B \end{array}$$

In type theory this can be defined as

$$\begin{aligned} &\texttt{type } A * B : \mathcal{U} \ \texttt{=} \\ &\quad \iota_1 : A \to A \sqcup B \\ &\quad \iota_2 : B \to A \sqcup B \end{aligned}$$

The join construction

Given two types A and B, their **join** $A * B$ is the homotopy pushout

$$\begin{array}{ccc} A \times B & \xrightarrow{\pi_2} & B \\ {\scriptstyle \pi_1}\downarrow & & \vdots\,{\scriptstyle \iota_2} \\ A & \underset{\iota_1}{\cdots\cdots\rightarrow} & A * B \end{array}$$

In type theory this can be defined as

$$\begin{aligned} &\texttt{type}\ A * B : \mathcal{U}\ \texttt{=} \\ &\quad \iota_1 : A \to A * B \\ &\quad \iota_2 : B \to A * B \\ &\quad \pi : (a\ :\ A) \to (b\ :\ B) \to a = b \end{aligned}$$

The join construction

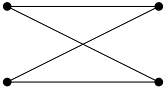
For instance, consider $A = \mathbf{2}$. We have that A is

The join construction

For instance, consider $A = \mathbf{2}$. We have that $A \sqcup A$ is

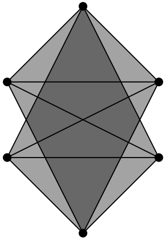
The join construction

For instance, consider $A = \mathbf{2}$. We have that $A * A$ is



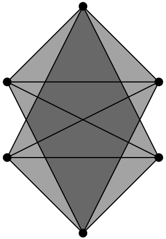
The join construction

For instance, consider $A = 2$. We have that $A * A * A$ is



The join construction

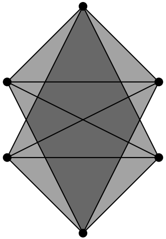
For instance, consider $A = 2$. We have that $A * A * A$ is



If compute the iterated joins A^{*n} the spaces get more and more connected.

The join construction

For instance, consider $A = \mathbf{2}$. We have that $A * A * A$ is

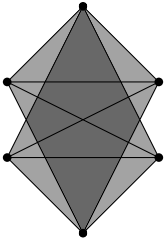


If compute the iterated joins A^{*n} the spaces get more and more connected.

Taking the inductive limit, we obtain $A^{*\infty} =$.

The join construction

For instance, consider $A = \mathbf{2}$. We have that $A * A * A$ is



If compute the iterated joins A^{*n} the spaces get more and more connected.

Taking the inductive limit, we obtain $A^{*\infty} = \mathbf{1}$.

The join construction

Previous example should not be surprising: we have

$$S^0 = 2$$

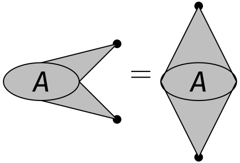
The join construction

Previous example should not be surprising: we have

$$S^0 = 2$$

and

$$A * S^0 = \Sigma A$$



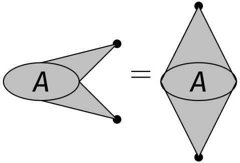
The join construction

Previous example should not be surprising: we have

$$S^0 = 2$$

and

$$A * S^0 = \Sigma A$$



so that

$$(S^0)^{*n} = \Sigma^n S^0 = S^n$$

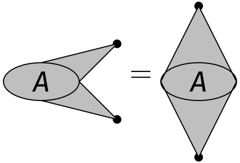
The join construction

Previous example should not be surprising: we have

$$S^0 = 2$$

and

$$A * S^0 = \Sigma A$$



so that

$$(S^0)^{*n} = \Sigma^n S^0 = S^n$$

The inductive limit is

$$(S^0)^{*\infty} = S^\infty$$

which is known to be contractible.

Connecteness

A type A is n-**connected** when $\|A\|_n = 1$.

Proposition
If A is m-connected and B is n-connected then $A * B$ is $(m+n+1)$-connected.

The join construction

We have seen that given a type A, we have

$$A^{*\infty} = 1$$

…excepting

The join construction

We have seen that given a type A, we have

$$A^{*\infty} = 1$$

…excepting when $A = 0$!

The join construction

We have seen that given a type A, we have

$$A^{*\infty} = \mathbf{1}$$

…excepting when $A = \mathbf{0}$!

In fact, it can be shown that

$$A^{*\infty} = \|A\|_{-1}$$

The join construction

We have seen that given a type A, we have

$$A^{*\infty} = 1$$

…excepting when $A = 0$!

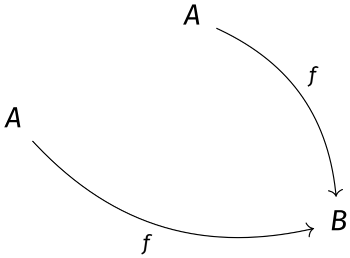
In fact, it can be shown that

$$A^{*\infty} = \|A\|_{-1}$$

This construction was known as the **Milnor construction** (1956).

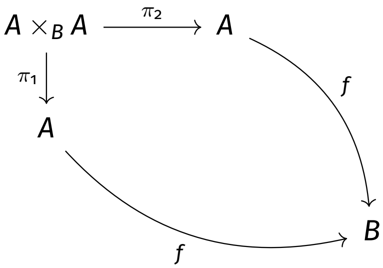
The join of maps

Given a map $f : A \to B$, consider the following construction



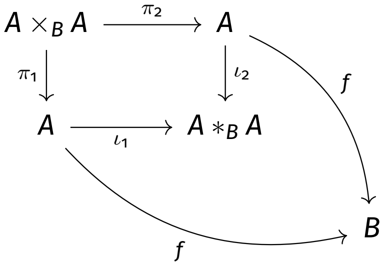
The join of maps

Given a map $f : A \to B$, consider the following construction



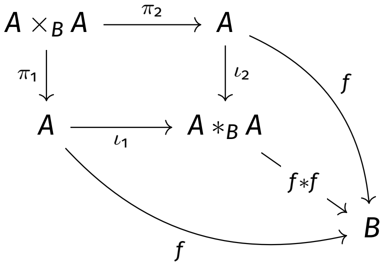
The join of maps

Given a map $f : A \to B$, consider the following construction



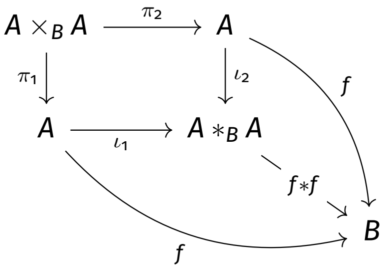
The join of maps

Given a map $f : A \to B$, consider the following construction



The join of maps

Given a map $f : A \to B$, consider the following construction



For $b : B$, we have

$$\mathrm{fib}_{f*f}(b) = \mathrm{fib}_f(b) * \mathrm{fib}_f(b)$$

with $\mathrm{fib}_f(b) = \Sigma(a : A).f\ a = b$.

The join of maps

If we compute iterated joins $f^{*n} = f * f * \ldots$, the fibers get more and more connected and at the colimit we have

$$\mathrm{fib}_{f^{*\infty}}(b) = (\mathrm{fib}_f(b))^{*\infty} = \mathbf{1}$$

excepting when $\mathrm{fib}_f(b) = \mathbf{0}$ where we get $\mathbf{0}$.

The join of maps

If we compute iterated joins $f^{*n} = f * f * \ldots$, the fibers get more and more connected and at the colimit we have

$$\mathrm{fib}_{f^{*\infty}}(b) = (\mathrm{fib}_f(b))^{*\infty} = 1$$

excepting when $\mathrm{fib}_f(b) = 0$ where we get 0.

In other words, $f^{*\infty}$ is the canonical inclusion

$$\mathrm{im}\, f := \Sigma(b : B).\| \mathrm{fib}_f(b) \|_{-1} \hookrightarrow B$$

Plan for the construction of $B\mathbb{Z}_2$

Rijke's general recipe for constructing a resolution of $B\mathbb{Z}_2$ is the following:

1. consider the map $f : \mathbf{1} \to B\mathbb{Z}_2$

Plan for the construction of $B\mathbb{Z}_2$

Rijke's general recipe for constructing a resolution of $B\mathbb{Z}_2$ is the following:

1. consider the map $f : \mathbf{1} \to B\mathbb{Z}_2$
2. compute the iterated joins $f^{*n} : P^n \to B\mathbb{Z}_2$

Plan for the construction of $B\mathbb{Z}_2$

Rijke's general recipe for constructing a resolution of $B\mathbb{Z}_2$ is the following:

1. consider the map $f : 1 \to B\mathbb{Z}_2$
2. compute the iterated joins $f^{*n} : P^n \to B\mathbb{Z}_2$
3. take the inductive limit $f^{*\infty} : P^\infty \to B\mathbb{Z}_2$

Plan for the construction of $B\mathbb{Z}_2$

Rijke's general recipe for constructing a resolution of $B\mathbb{Z}_2$ is the following:

1. consider the map $f : \mathbf{1} \to B\mathbb{Z}_2$
2. compute the iterated joins $f^{*n} : P^n \to B\mathbb{Z}_2$
3. take the inductive limit $f^{*\infty} : P^\infty \to B\mathbb{Z}_2$
4. conclude that $P^\infty = \mathrm{im} f = B\mathbb{Z}_2$

Plan for the construction of $B\mathbb{Z}_2$

Rijke's general recipe for constructing a resolution of $B\mathbb{Z}_2$ is the following:

1. consider the map $f : 1 \to B\mathbb{Z}_2$
2. compute the iterated joins $f^{*n} : P^n \to B\mathbb{Z}_2$
3. take the inductive limit $f^{*\infty} : P^\infty \to B\mathbb{Z}_2$
4. conclude that $P^\infty = \operatorname{im} f = B\mathbb{Z}_2$
5. observe that P^∞ is a polygraph / CW-complex

Plan for the construction of $B\mathbb{Z}_2$

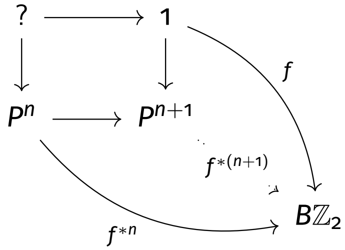
Rijke's general recipe for constructing a resolution of $B\mathbb{Z}_2$ is the following:

1. consider the map $f : 1 \to B\mathbb{Z}_2$
2. compute the iterated joins $f^{*n} : P^n \to B\mathbb{Z}_2$
3. take the inductive limit $f^{*\infty} : P^\infty \to B\mathbb{Z}_2$
4. conclude that $P^\infty = \operatorname{im} f = B\mathbb{Z}_2$
5. observe that P^∞ is a polygraph / CW-complex

This recipe works for any BG (excepting the last point)!

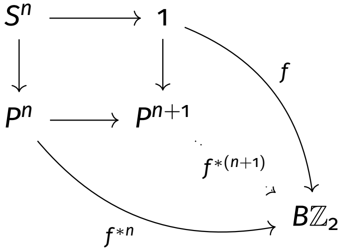
Iterated joins

We have $f^{*(n+1)} = f^{*n} * f$:

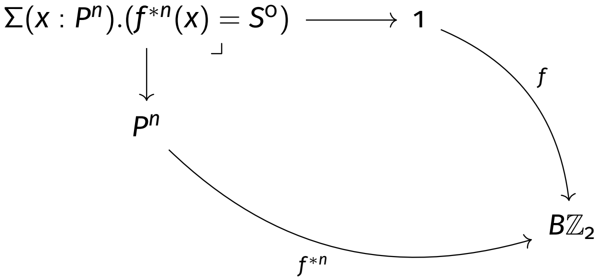


Iterated joins

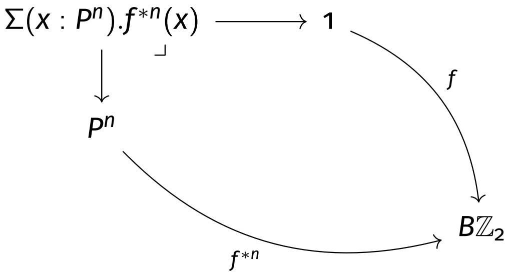
We have $f^{*(n+1)} = f^{*n} * f$:



Iterated joins: inductive case



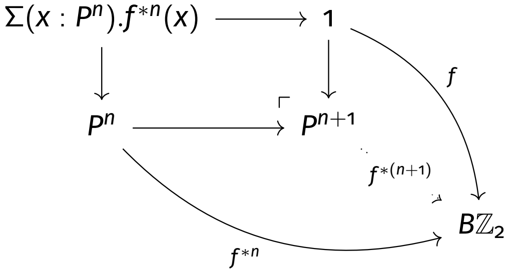
Iterated joins: inductive case



Iterated joins: inductive case

$$
\begin{array}{ccc}
\Sigma(x : P^n).f^{*n}(x) & \longrightarrow & 1 \\
\downarrow & & \downarrow \\
P^n & \longrightarrow & \ulcorner\, P^{n+1}
\end{array}
$$

Iterated joins: inductive case



Iterated joins: inductive case

$$
\begin{array}{ccc}
\Sigma(z : \Sigma(x : P^n).f^{*n}(x)).f(\star) & \longrightarrow & \Sigma(x : 1).f(\star) \\
\downarrow & & \downarrow \\
\Sigma(x : P^n).f^{*n} & \longrightarrow & \ulcorner\ \Sigma(x : P^{n+1})
\end{array}
$$

Iterated joins: inductive case

$$
\begin{array}{ccc}
(\Sigma(x : P^n).f^{*n}(x)) \times S^0 & \longrightarrow & S^0 \\
\big\downarrow & & \big\downarrow \\
\Sigma(x : P^n).f^{*n}(x) & \longrightarrow & \ulcorner\, \Sigma(x : P^{n+1}).f^{*(n+1)}
\end{array}
$$

Iterated joins: inductive case

$$
\begin{array}{ccc}
(\Sigma(x : P^n).f^{*n}(x)) \times S^0 & \longrightarrow & S^0 \\
\downarrow & & \downarrow \\
\Sigma(x : P^n).f^{*n}(x) & \longrightarrow & \ulcorner\ \Sigma(x : P^{n+1}).f^{*(n+1)}
\end{array}
$$

And thus

$$\Sigma(x : P^{n+1}).f^{*(n+1)} = (\Sigma(x : P^n).f^{*n}(x)) * S^0$$

Iterated joins: inductive case

$$\begin{array}{ccc} \Sigma(x : P^n).f^{*n}(x) & \longrightarrow & 1 \\ \big\downarrow & & \big\downarrow \\ P^n & \longrightarrow & \ulcorner\, P^{n+1} \end{array}$$

And thus

$$\Sigma(x : P^{n+1}).f^{*(n+1)} = (\Sigma(x : P^n).f^{*n}(x)) * S^0$$

A concrete implementation of $B\mathbb{Z}_2$

In order to perform computations, we need a concrete implementation of $B\mathbb{Z}_2$.

A concrete implementation of $B\mathbb{Z}_2$

In order to perform computations, we need a concrete implementation of $B\mathbb{Z}_2$.

Consider the type $\mathbb{B} := \{0, 1\}$ of booleans.

A concrete implementation of $B\mathbb{Z}_2$

In order to perform computations, we need a concrete implementation of $B\mathbb{Z}_2$.

Consider the type $\mathbb{B} := \{0, 1\}$ of booleans.

There are two isomorphisms $\mathbb{B} \to \mathbb{B}$: the identity and the swap.

A concrete implementation of $B\mathbb{Z}_2$

In order to perform computations, we need a concrete implementation of $B\mathbb{Z}_2$.

Consider the type $\mathbb{B} := \{0, 1\}$ of booleans.

There are two isomorphisms $\mathbb{B} \to \mathbb{B}$: the identity and the swap.

By univalence, we thus have $(\mathbb{B} = \mathbb{B}) = \mathbb{Z}_2$.

$$\mathrm{id} \circlearrowright \mathbb{B} \circlearrowleft \mathrm{swap}$$

A concrete implementation of $B\mathbb{Z}_2$

In order to perform computations, we need a concrete implementation of $B\mathbb{Z}_2$.

Consider the type $\mathbb{B} := \{0, 1\}$ of booleans.

There are two isomorphisms $\mathbb{B} \to \mathbb{B}$: the identity and the swap.

By univalence, we thus have $(\mathbb{B} = \mathbb{B}) = \mathbb{Z}_2$.

$$\text{id} \circlearrowright \mathbb{B} \circlearrowleft \text{swap}$$

We can therefore define $B\mathbb{Z}_2$ as the connected component of $\mathbb{B}$ in the universe:

$$B\mathbb{Z}_2 = \Sigma(X : \mathcal{U}).\|X = \mathbb{B}\|_{-1}$$

which satisfies

$$\Omega B\mathbb{Z}_2 = \mathbb{Z}_2$$

A concrete implementation of $B\mathbb{Z}_2$

In order to perform computations, we need a concrete implementation of $B\mathbb{Z}_2$.

Consider the type $\mathbb{B} := \{0, 1\}$ of booleans.

There are two isomorphisms $\mathbb{B} \to \mathbb{B}$: the identity and the swap.

By univalence, we thus have $(\mathbb{B} = \mathbb{B}) = \mathbb{Z}_2$.

$$\text{id} \circlearrowright \mathbb{B} \circlearrowleft \text{swap}$$

More generally, we have, for $A : B\mathbb{Z}_2$

$$(\mathbb{B} = A) \quad = \quad A$$

and thus, for $A : X \to B\mathbb{Z}_2$,

$$\Sigma(x : X).(\mathbb{B} = A(x)) \quad = \quad \Sigma(x : X).A(x)$$

A fiber sequence

Writing

$$p : S^n \to P^n$$

it can be shown that for $x : P^n$, we have

$$\mathrm{fib}_p(x) = \Sigma(y : S^n).(p(y) = x) = S^0$$

i.e. we have a **fiber sequence**

$$S^0 \hookrightarrow S^n \xrightarrow{p} P^n$$

A long exact sequence

By general arguments, such a fiber sequence induces a long exact sequence of homotopy groups

$$\cdots \longrightarrow \pi_3(S^0) \longrightarrow \pi_3(S^n) \longrightarrow \pi_3(P^n)$$

$$\longrightarrow \pi_2(S^0) \longrightarrow \pi_2(S^n) \longrightarrow \pi_2(P^n)$$

$$\longrightarrow \pi_1(S^0) \longrightarrow \pi_1(S^n) \longrightarrow \pi_1(P^n)$$

A long exact sequence

By general arguments, such a fiber sequence induces a long exact sequence of homotopy groups

$$\cdots \longrightarrow 0 \longrightarrow \pi_3(S^n) \longrightarrow \pi_3(P^n) \longrightarrow 0 \longrightarrow \pi_2(S^n) \longrightarrow \pi_2(P^n) \longrightarrow 0 \longrightarrow \pi_1(S^n) \longrightarrow \pi_1(P^n)$$

A long exact sequence

By general arguments, such a fiber sequence induces a long exact sequence of homotopy groups

$$\cdots \longrightarrow 0 \longrightarrow \pi_3(S^n) \longrightarrow \pi_3(P^n) \longrightarrow 0 \longrightarrow \pi_2(S^n) \longrightarrow \pi_2(P^n) \longrightarrow 0 \longrightarrow \pi_1(S^n) \longrightarrow \pi_1(P^n)$$

The P^n we have defined thus has the right homotopy groups!

Part V

Lens spaces

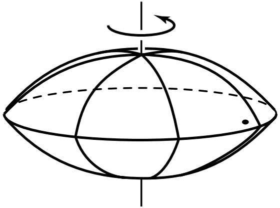
This generalizes to cyclic groups $\mathbb{Z}_m$!

(which requires a bit more than replacing 2 by m)

(work in progress with Émile Oleon)

Lens spaces

There is a geometric construction for $B\mathbb{Z}_m$ called **infinite lens spaces**



We can implement it in HoTT.

We first need to define $B\mathbb{Z}_m$.

Equality between endomorphisms

We write

$$\mathsf{Fin}\, m = \{0, 1, \ldots, m-1\}$$

and $s : \mathsf{Fin}\, m \to \mathsf{Fin}\, m$ for the successor (modulo m).

Equality between endomorphisms

We write

$$\mathsf{Fin}\, m = \{0, 1, \ldots, m-1\}$$

and $s : \mathsf{Fin}\, m \to \mathsf{Fin}\, m$ for the successor (modulo m).

An equality $s = s$ is the same as an isomorphism $f : \mathsf{Fin}\, m \to \mathsf{Fin}\, m$ such that

$$\begin{array}{ccc} \mathsf{Fin}\, m & \xrightarrow{f} & \mathsf{Fin}\, m \\ {\scriptstyle s}\downarrow & & \downarrow{\scriptstyle s} \\ \mathsf{Fin}\, m & \xrightarrow[f]{} & \mathsf{Fin}\, m \end{array}$$

Equality between endomorphisms

We write

$$\mathrm{Fin}\, m = \{0, 1, \ldots, m-1\}$$

and $s : \mathrm{Fin}\, m \to \mathrm{Fin}\, m$ for the successor (modulo m).

An equality $s = s$ is the same as an isomorphism $f : \mathrm{Fin}\, m \to \mathrm{Fin}\, m$ such that

$$\begin{array}{ccc} \mathrm{Fin}\, m & \xrightarrow{f} & \mathrm{Fin}\, m \\ {\scriptstyle s}\downarrow & & \downarrow{\scriptstyle s} \\ \mathrm{Fin}\, m & \xrightarrow[f]{} & \mathrm{Fin}\, m \end{array}$$

Such an isomorphism f is determined by the image of 0 since

$$f(k) = f(s^k(0)) = s^k(f(0))$$

Equality between endomorphisms

We write

$$\mathsf{Fin}\, m = \{0, 1, \dots, m-1\}$$

and $s : \mathsf{Fin}\, m \to \mathsf{Fin}\, m$ for the successor (modulo m).

An equality $s = s$ is the same as an isomorphism $f : \mathsf{Fin}\, m \to \mathsf{Fin}\, m$ such that

$$\begin{array}{ccc} \mathsf{Fin}\, m & \xrightarrow{f} & \mathsf{Fin}\, m \\ {\scriptstyle s}\downarrow & & \downarrow{\scriptstyle s} \\ \mathsf{Fin}\, m & \xrightarrow[f]{} & \mathsf{Fin}\, m \end{array}$$

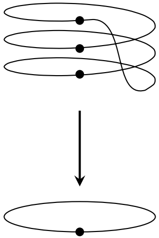
Such an isomorphism f is determined by the image of 0 since

$$f(k) = f(s^k(0)) = s^k(f(0))$$

i.e. $(s = s) = \mathsf{Fin}\, m$.

Equality between endomorphisms

The picture to have in mind is



Cyclic groups

We define the type of **endomorphisms**

$$\mathcal{U}^{\circlearrowleft} = \Sigma(A : \mathcal{U}).(A \to A)$$

Cyclic groups

We define the type of **endomorphisms**

$$\mathcal{U}^{\circlearrowleft} = \Sigma(A : \mathcal{U}).(A \to A)$$

We consider the object

$$\sigma = (\mathsf{Fin}\, m, s)$$

Cyclic groups

We define the type of **endomorphisms**

$$\mathcal{U}^{\circlearrowleft} = \Sigma(A : \mathcal{U}).(A \to A)$$

We consider the object

$$\sigma = (\mathsf{Fin}\, m, s)$$

We define

$$B\mathbb{Z}_m = \mathcal{U}^{\circlearrowleft}_{S} = \Sigma(X : \mathcal{U}^{\circlearrowleft}).\|X = \sigma\|_{-1}$$

Lens spaces

We have a map

$$f : 1 \to B\mathbb{Z}_m$$

given by S and we can define

$$B\mathbb{Z}_m = \operatorname{im} f = \partial^-(f^{*\infty})$$

Lens spaces

We have a map

$$f : \mathbf{1} \to B\mathbb{Z}_m$$

given by S and we can define

$$B\mathbb{Z}_m = \operatorname{im} f = \partial^-(f^{*\infty})$$

We also have a map

$$f : S^1 \to B\mathbb{Z}_m$$

and we can define

$$B\mathbb{Z}_m = \operatorname{im} f = \partial^-(f^{*\infty})$$

Lens spaces

To be more precise there are multiple maps

$$f^k : S^1 \to B\mathbb{Z}_m$$

$$\star \mapsto \sigma$$

$$\texttt{loop} \mapsto s^k : \sigma \simeq \sigma$$

Given $k_1, \ldots, k_n$ all relatively prime to m, we can define

$$L(k_1, \ldots, k_n) = \partial^-(f^{k_1} * \ldots * f^{k_n})$$

which correspond to the well-known **lens spaces**.

Lens spaces

To be more precise there are multiple maps

$$f^k : S^1 \to B\mathbb{Z}_m$$

$$\star \mapsto \sigma$$

$$\texttt{loop} \mapsto s^k : \sigma \simeq \sigma$$

Given $k_1, \ldots, k_n$ all relatively prime to m, we can define

$$L(k_1, \ldots, k_n) = \partial^-(f^{k_1} * \ldots * f^{k_n})$$

which correspond to the well-known **lens spaces**.

By default,

$$L^n = L(1, \ldots, 1)$$

and

$$L^\infty = \text{colim}_n\, L^n$$

Lens spaces

It can be shown that we have a pushout

$$\begin{array}{ccc} S^{2n+1} & \longrightarrow & 1 \\ \downarrow & & \downarrow \\ L^n & \longrightarrow & L^{n+1} \end{array}$$

from which we can deduce that we have a cellular decomposition
with one cell in each dimension:
at each step we are adding a cell in dimension $2n$ and one in dimension $2n+1$.

Lens spaces: traditional definition

We can see S^{2n-1} as a subset of $\mathcal{C}^n$:

$$S^{2n-1} = \{(z_1, \ldots, z_n) \mid |z_1|^2 + \ldots + |z_n|^2 = 1\}$$

Lens spaces: traditional definition

We can see S^{2n-1} as a subset of $\mathcal{C}^n$:

$$S^{2n-1} = \{(z_1, \ldots, z_n) \mid |z_1|^2 + \ldots + |z_n|^2 = 1\}$$

There is a free action ζ of $\mathbb{Z}_m$ on S^{2n-1} given by

$$1 \cdot (z_1, \ldots, z_n) = (e^{2i\pi/m} z_1, \ldots, e^{2i\pi/m} z_n)$$

Lens spaces: traditional definition

We can see S^{2n-1} as a subset of $\mathcal{C}^n$:

$$S^{2n-1} = \{(z_1, \ldots, z_n) \mid |z_1|^2 + \ldots + |z_n|^2 = 1\}$$

There is a free action ζ of $\mathbb{Z}_m$ on S^{2n-1} given by

$$1 \cdot (z_1, \ldots, z_n) = (\mathrm{e}^{2i\pi/m} z_1, \ldots, \mathrm{e}^{2i\pi/m} z_n)$$

and we define

$$L^n = S^{2n-1}/\zeta$$

Lens spaces: traditional definition

We can see S^{2n-1} as a subset of $\mathcal{C}^n$:

$$S^{2n-1} = \{(z_1, \ldots, z_n) \mid |z_1|^2 + \ldots + |z_n|^2 = 1\}$$

There is a free action ζ of $\mathbb{Z}_m$ on S^{2n-1} given by

$$1 \cdot (z_1, \ldots, z_n) = (\mathrm{e}^{2i\pi/m} z_1, \ldots, \mathrm{e}^{2i\pi/m} z_n)$$

and we define

$$L^n = S^{2n-1}/\zeta$$

we thus get a covering $S^{2n-1} \twoheadrightarrow L^n$ with Z_m as fiber, i.e. a fiber sequence

$$\mathbb{Z}_m \hookrightarrow S^{2n-1} \twoheadrightarrow L^n$$

Lens spaces: traditional definition

We can see S^{2n-1} as a subset of $\mathcal{C}^n$:

$$S^{2n-1} = \{(z_1, \ldots, z_n) \mid |z_1|^2 + \ldots + |z_n|^2 = 1\}$$

There is a free action ζ of $\mathbb{Z}_m$ on S^{2n-1} given by

$$1 \cdot (z_1, \ldots, z_n) = (\mathrm{e}^{2i\pi/m} z_1, \ldots, \mathrm{e}^{2i\pi/m} z_n)$$

and we define

$$L^n = S^{2n-1}/\zeta$$

we thus get a covering $S^{2n-1} \twoheadrightarrow L^n$ with Z_m as fiber, i.e. a fiber sequence

$$\mathbb{Z}_m \hookrightarrow S^{2n-1} \twoheadrightarrow L^n$$

from which we deduce $\pi_1(L^n) = \mathbb{Z}_m$ and $\pi_k(L^n) = 0$ for $1 < k < 2n - 1$.

Lens spaces: traditional definition

We can see S^{2n-1} as a subset of $\mathcal{C}^n$:

$$S^{2n-1} = (S^1)^{*n}$$

There is a free action ζ of $\mathbb{Z}_m$ on S^{2n-1} given by

$$1 \cdot (z_1, \ldots, z_n) = (e^{2i\pi/m} z_1, \ldots, e^{2i\pi/m} z_n)$$

and we define

$$L^n = S^{2n-1}/\zeta$$

we thus get a covering $S^{2n-1} \twoheadrightarrow L^n$ with Z_m as fiber, i.e. a fiber sequence

$$\mathbb{Z}_m \hookrightarrow S^{2n-1} \twoheadrightarrow L^n$$

from which we deduce $\pi_1(L^n) = \mathbb{Z}_m$ and $\pi_k(L^n) = 0$ for $1 < k < 2n - 1$.

Lens spaces

From

$$f : S^1 \to B\mathbb{Z}_m$$

we obtained by joining n times a map

$$f^{*n} : L^n \to B\mathbb{Z}_m$$

Lens spaces

From

$$f : S^1 \to B\mathbb{Z}_m$$

we obtained by joining n times a map

$$f^{*n} : L^n \to B\mathbb{Z}_m$$

Via the family/fibration correspondence it corresponds to a map

$$g : B\mathbb{Z}_m \to \mathcal{U} \qquad g(x) = \mathrm{fib}_{f^{*n}}(x) \qquad g(\star) = S^{2n-1}$$

i.e. an action of $B\mathbb{Z}_m$ on S^{2n-1}

Lens spaces

From

$$f : S^1 \to B\mathbb{Z}_m$$

we obtained by joining n times a map

$$f^{*n} : L^n \to B\mathbb{Z}_m$$

Via the family/fibration correspondence it corresponds to a map

$$g : B\mathbb{Z}_m \to \mathcal{U} \qquad g(x) = \mathrm{fib}_{f^{*n}}(x) \qquad g(\star) = S^{2n-1}$$

i.e. an action of $B\mathbb{Z}_m$ on S^{2n-1} and we have

$$S^{2n-1}/B\mathbb{Z}_m = \Sigma(x : B\mathbb{Z}_m).g(x) = \Sigma(x : B\mathbb{Z}_m).\,\mathrm{fib}_{f^{*n}}(x) = L^n$$

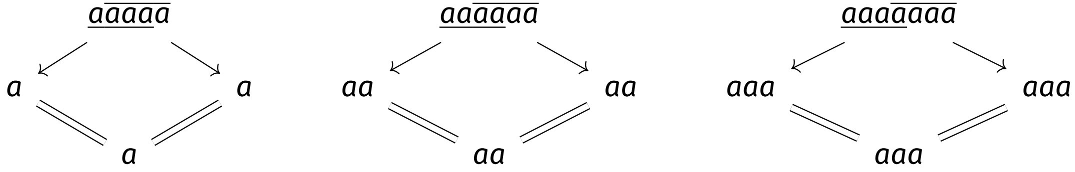
which corresponds to the usual definition of lens spaces!

Compared to rewriting

We have a presentation

$$\mathbb{Z}_m = \langle a \mid a^m = 1 \rangle$$

If we compute the critical branchings for $\mathbb{Z}_4$, we get 3 of them:



which would inevitably lead to a larger presentation…

Lots remains to be done!

Closing words

Thank you François!