

Reducing higher inductive types to quotients

Floris van Doorn

Carnegie Mellon University

May 20, 2016

Goal: Reduce complicated higher inductive types to simpler ones.

Analogue: In Extensional Type Theory, we can reduce all inductive types to W -types and Σ -types.

Question: What makes a higher inductive type complicated?

- It is recursive (n -truncation, localization, spectrification)
- It has higher-dimensional path-constructors (torus, Eilenberg-MacLane spaces $K(G, 1)$)
- Higher inductive-inductive types (Cauchy reals)

Goal: Reduce complicated higher inductive types to simpler ones.

Analogue: In Extensional Type Theory, we can reduce all inductive types to W -types and Σ -types.

Question: What makes a higher inductive type complicated?

- It is recursive (n -truncation, localization, spectrification)
- It has higher-dimensional path-constructors (torus, Eilenberg-MacLane spaces $K(G, 1)$)
- Higher inductive-inductive types (Cauchy reals)

Question: What higher inductive types do we start with?

Given $A : \mathcal{U}$ and $R : A \rightarrow A \rightarrow \mathcal{U}$ the quotient is:

HIT $quotient_A(R) :=$

- $q : A \rightarrow quotient_A(R)$
- $\Pi(x, y : A), R(x, y) \rightarrow q(x) = q(y)$

This is the homotopy-coequalizer of the projections

$$\Sigma(x, y : A), R(x, y) \begin{array}{c} \xrightarrow{\pi_1} \\ \xrightarrow{\pi_2} \end{array} A$$

Using quotients we can define other simple HITs:

- The pushout of $C \xleftarrow{g} A \xrightarrow{f} B$ is the quotient of $B + C$ under the relation R , defined as a inductively

inductive $R : (B + C) \rightarrow (B + C) \rightarrow \mathcal{U} :=$

- ▶ $\Pi(a : A), R(\text{inl}(f(a)), \text{inr}(g(a)))$

Quotients

Using quotients we can define other simple HITs:

- The pushout of $C \xleftarrow{g} A \xrightarrow{f} B$ is the quotient of $B + C$ under the relation R , defined as a inductively

inductive $R : (B + C) \rightarrow (B + C) \rightarrow \mathcal{U} :=$

- ▶ $\Pi(a : A), R(\text{inl}(f(a)), \text{inr}(g(a)))$

- The colimit of $A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} \dots$ is the quotient of $\Sigma(n : \mathbb{N}), A(n)$ under the relation S , defined inductively

inductive $S : (\Sigma(n : \mathbb{N}), A(n)) \rightarrow (\Sigma(n : \mathbb{N}), A(n)) \rightarrow \mathcal{U} :=$

- ▶ $\Pi(n : \mathbb{N})(a : A_n), S((n, a), (n + 1, f_n(a)))$

Quotients

Using quotients we can define other simple HITs:

- The pushout of $C \xleftarrow{g} A \xrightarrow{f} B$ is the quotient of $B + C$ under the relation R , defined as a inductively

inductive $R : (B + C) \rightarrow (B + C) \rightarrow \mathcal{U} :=$

- ▶ $\prod (a : A), R(\text{inl}(f(a)), \text{inr}(g(a)))$

- The colimit of $A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} \dots$ is the quotient of $\Sigma(n : \mathbb{N}), A(n)$ under the relation S , defined inductively

inductive $S : (\Sigma(n : \mathbb{N}), A(n)) \rightarrow (\Sigma(n : \mathbb{N}), A(n)) \rightarrow \mathcal{U} :=$

- ▶ $\prod (n : \mathbb{N})(a : A_n), S((n, a), (n + 1, f_n(a)))$

- This also gives suspensions, spheres, wedge product, join, smash product, cofibers, ...

We can construct more HITs from quotients.

Today I will talk about the construction of

- The propositional truncation
- HITs with 2-constructors (torus, groupoid quotient, Eilenberg-MacLane spaces $K(G, 1)$, reduced suspension, reflexive quotient)

Work in progress:

- Define ω -compact localizations (which includes all n -truncations) using quotients (Egbert Rijke will talk about this in the afternoon).

Propositional Truncation

The Propositional Truncation $\|-\|$ as a HIT:

HIT $\|A\| :=$

- $|-| : A \rightarrow \|A\|$
- $\varepsilon : \Pi(x, y : \|A\|), x = y$

Propositional Truncation

The Propositional Truncation $\|-\|$ as a HIT:

HIT $\|A\| :=$

- $|-| : A \rightarrow \|A\|$
- $\varepsilon : \Pi(x, y : \|A\|), x = y$

ε is a recursive: the domain of the recursor is the type $\|A\|$ being constructed.

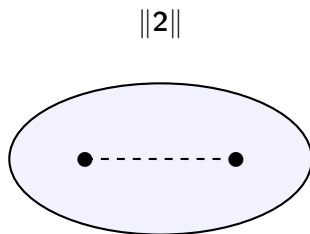
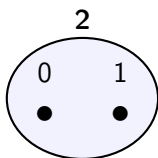
Propositional Truncation

The Propositional Truncation $\|- \|$ as a HIT:

HIT $\|A\| :=$

- $|-| : A \rightarrow \|A\|$
- $\varepsilon : \Pi(x, y : \|A\|), x = y$

ε is a recursive: the domain of the recursor is the type $\|A\|$ being constructed.



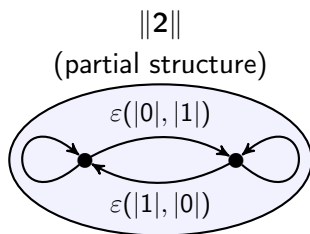
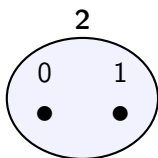
Propositional Truncation

The Propositional Truncation $\|- \|$ as a HIT:

HIT $\|A\| :=$

- $|-| : A \rightarrow \|A\|$
- $\varepsilon : \Pi(x, y : \|A\|), x = y$

ε is a recursive: the domain of the recursor is the type $\|A\|$ being constructed.



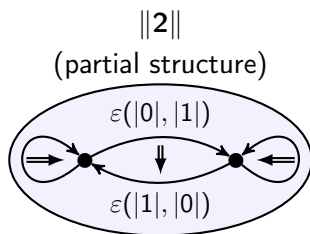
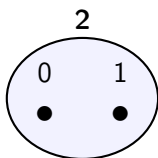
Propositional Truncation

The Propositional Truncation $\|-$ as a HIT:

HIT $\|A\| :=$

- $|- : A \rightarrow \|A\|$
- $\varepsilon : \Pi(x, y : \|A\|), x = y$

ε is a recursive: the domain of the recursor is the type $\|A\|$ being constructed.



Proof that $\varepsilon(|0|, |1|) = \varepsilon(|1|, |0|)^{-1}$.

Propositional Truncation

Proof that $\varepsilon(|0|, |1|) = \varepsilon(|1|, |0|)^{-1}$.

We prove the following statement:

$$\prod(x : ||2||)(p : |0| = x), p = \varepsilon(|0|, |0|)^{-1} \cdot \varepsilon(|0|, x).$$

Proof that $\varepsilon(|0|, |1|) = \varepsilon(|1|, |0|)^{-1}$.

We prove the following statement:

$$\Pi(x : ||2||)(p : |0| = x), p = \varepsilon(|0|, |0|)^{-1} \cdot \varepsilon(|0|, x).$$

We can prove this by path induction on p . Then we need to prove

$$\text{refl}_{|0|} = \varepsilon(|0|, |0|)^{-1} \cdot \varepsilon(|0|, |0|),$$

which is an ∞ -groupoid law.

Propositional Truncation

Proof that $\varepsilon(|0|, |1|) = \varepsilon(|1|, |0|)^{-1}$.

We prove the following statement:

$$\Pi(x : ||2||)(p : |0| = x), p = \varepsilon(|0|, |0|)^{-1} \cdot \varepsilon(|0|, x).$$

We can prove this by path induction on p . Then we need to prove

$$\text{refl}_{|0|} = \varepsilon(|0|, |0|)^{-1} \cdot \varepsilon(|0|, |0|),$$

which is an ∞ -groupoid law.

Now $\varepsilon(|0|, |1|)$ and $\varepsilon(|1|, |0|)^{-1}$ are both equal to the same path, hence equal to each other.

One-step truncation

We define the *one-step truncation* $\{A\}$, which is the following HIT.

HIT $\{A\} :=$

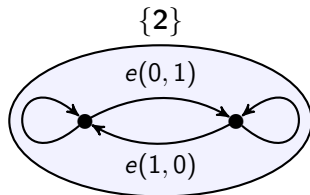
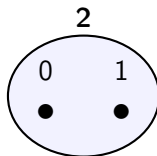
- $f : A \rightarrow \{A\}$
- $e : \Pi(x, y : A), f(x) = f(y)$

One-step truncation

We define the *one-step truncation* $\{A\}$, which is the following HIT.

HIT $\{A\} :=$

- $f : A \rightarrow \{A\}$
- $e : \prod(x, y : A), f(x) = f(y)$

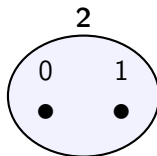


One-step truncation

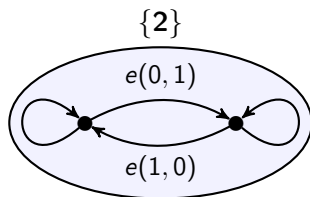
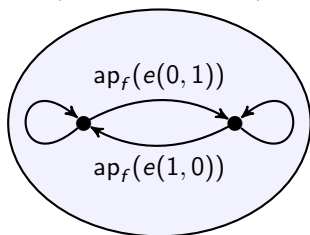
We define the *one-step truncation* $\{A\}$, which is the following HIT.

HIT $\{A\} :=$

- $f : A \rightarrow \{A\}$
- $e : \Pi(x, y : A), f(x) = f(y)$



$\{\{2\}\}$
(partial structure)

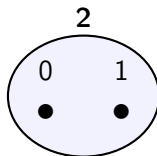


One-step truncation

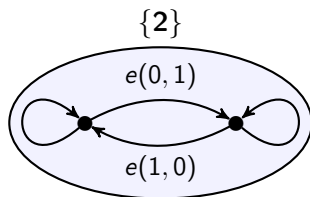
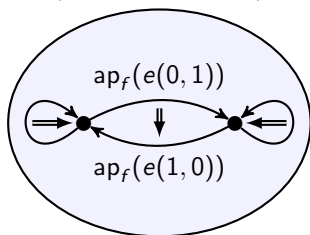
We define the *one-step truncation* $\{A\}$, which is the following HIT.

HIT $\{A\} :=$

- $f : A \rightarrow \{A\}$
- $e : \Pi(x, y : A), f(x) = f(y)$



$\{\{2\}\}$
(partial structure)



One-step truncation

Proof that $\text{ap}_f(e(0, 1)) = (\text{ap}_f(e(1, 0)))^{-1}$.

[This equality lives in type $f(f(0)) =_{\{\{2\}\}} f(f(1))$]

One-step truncation

Proof that $\text{ap}_f(e(0, 1)) = (\text{ap}_f(e(1, 0)))^{-1}$.

[This equality lives in type $f(f(0)) =_{\{\{2\}\}} f(f(1))$]

Definition. $g : A \rightarrow B$ is *weakly constant* if $\prod(x, y : A), g(x) = g(y)$

One-step truncation

Proof that $\text{ap}_f(e(0, 1)) = (\text{ap}_f(e(1, 0)))^{-1}$.

[This equality lives in type $f(f(0)) =_{\{\{2\}\}} f(f(1))$]

Definition. $g : A \rightarrow B$ is *weakly constant* if $\prod(x, y : A), g(x) = g(y)$

Lemma. If $g : A \rightarrow B$ is weakly constant, then for any $x, y : A$ the function $\text{ap}_g : x = y \rightarrow g(x) = g(y)$ is weakly constant.

One-step truncation

Proof that $\text{ap}_f(e(0, 1)) = (\text{ap}_f(e(1, 0)))^{-1}$.

[This equality lives in type $f(f(0)) =_{\{\{2\}\}} f(f(1))$]

Definition. $g : A \rightarrow B$ is *weakly constant* if $\Pi(x, y : A), g(x) = g(y)$

Lemma. If $g : A \rightarrow B$ is weakly constant, then for any $x, y : A$ the function $\text{ap}_g : x = y \rightarrow g(x) = g(y)$ is weakly constant.

Proof. Fix $x : A$ and let q be the proof that g is weakly constant. We first prove:

$$\Pi(z : A)(p : x = z), \text{ap}_g(p) = q(x, x)^{-1} \cdot q(x, z).$$

One-step truncation

Proof that $\text{ap}_f(e(0, 1)) = (\text{ap}_f(e(1, 0)))^{-1}$.

[This equality lives in type $f(f(0)) =_{\{\{2\}\}} f(f(1))$]

Definition. $g : A \rightarrow B$ is *weakly constant* if $\prod(x, y : A), g(x) = g(y)$

Lemma. If $g : A \rightarrow B$ is weakly constant, then for any $x, y : A$ the function $\text{ap}_g : x = y \rightarrow g(x) = g(y)$ is weakly constant.

Proof. Fix $x : A$ and let q be the proof that g is weakly constant. We first prove:

$$\prod(z : A)(p : x = z), \text{ap}_g(p) = q(x, x)^{-1} \cdot q(x, z).$$

This again follows by path induction.

Now for any $p, q : x = y$ both $\text{ap}_g(p)$ and $\text{ap}_g(q)$ are both equal to $q(x, x)^{-1} \cdot q(x, y)$. □

One-step truncation

Proof that $\text{ap}_f(e(0, 1)) = (\text{ap}_f(e(1, 0)))^{-1}$.

[This equality lives in type $f(f(0)) =_{\{\{2\}\}} f(f(1))$]

Definition. $g : A \rightarrow B$ is *weakly constant* if $\prod(x, y : A), g(x) = g(y)$

Lemma. If $g : A \rightarrow B$ is weakly constant, then for any $x, y : A$ the function $\text{ap}_g : x = y \rightarrow g(x) = g(y)$ is weakly constant.

Proof. Fix $x : A$ and let q be the proof that g is weakly constant. We first prove:

$$\prod(z : A)(p : x = z), \text{ap}_g(p) = q(x, x)^{-1} \cdot q(x, z).$$

This again follows by path induction.

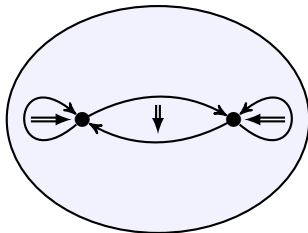
Now for any $p, q : x = y$ both $\text{ap}_g(p)$ and $\text{ap}_g(q)$ are both equal to $q(x, x)^{-1} \cdot q(x, y)$. □

Since e proves that f is weakly-constant, we have

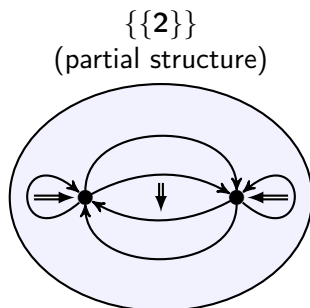
$$\text{ap}_f(e(0, 1)) = \text{ap}_f(e(1, 0)^{-1}) = (\text{ap}_f(e(1, 0)))^{-1}.$$

One-step truncation

$\{\{2\}\}$
(partial structure)



One-step truncation



However, we have $e(f(0), f(1)) \neq ap_f(e(0, 1))$.

Lemma. If $p : a =_A b$, then the paths $ap_f(p)$ and $e(a, b)$ in $\{A\}$ are provably different, i.e. $ap_f(p) \neq e(a, b)$.

Proof sketch. We can define a map $\{A\} \rightarrow S^1$ sending all point constructors to base and all path constructors $e(a, b)$ to loop.

Construction of the Propositional truncation

We define $\{A\}_\infty$ as the colimit of this diagram:

$$A \xrightarrow{f} \{A\} \xrightarrow{f} \{\{A\}\} \xrightarrow{f} \{\{\{A\}\}\} \xrightarrow{f} \dots \quad (1)$$

Theorem

$\{A\}_\infty$ is the propositional truncation of A .

Corollary

A function in $\|A\| \rightarrow B$ is the same as a cocone over (1), for any type B .

2-HITs

We want to construct HITs with 2-path constructors.

Examples:

HIT $T^2 :=$

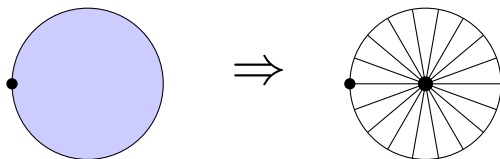
- $b : T^2$
- $l_1, l_2 : b = b$
- $s : l_1 \cdot l_2 = l_2 \cdot l_1$

HIT $K(G, 1) :=$

- $\star : K(G, 1)$
- $p : G \rightarrow \star = \star$
- $m : \Pi(g, h : G), p(gh) = p(g) \cdot p(h)$
- $K(G, 1)$ is 1-truncated

Hubs and spokes

In the book, 2-HITs are reduced to 1-HITs using the hubs and spokes method.



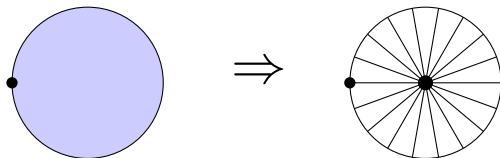
The torus becomes:

HIT $T^2 :=$

- $b : T^2$
- $\ell_1, \ell_2 : b = b$
- $h : T^2$
- $s : \Pi(x : S^1),$ $\underbrace{\hspace{10em}}_{S^1 \rightarrow T^2} x = h$

Hubs and spokes

In the book, 2-HITs are reduced to 1-HITs using the hubs and spokes method.



The torus becomes:

HIT $T^2 :=$

- $b : T^2$
- $l_1, l_2 : b = b$
- $h : T^2$
- $s : \Pi(x : S^1), \underbrace{\text{circle.rec } b (l_1 \cdot l_2 \cdot l_1^{-1} \cdot l_2^{-1})}_{S^1 \rightarrow T^2} x = h$

Hubs and spokes

Why does hubs and spokes work?

For $a_0 : A$ and $p : a_0 = a_0$ we have

$$(\Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h) \simeq (p = 1).$$

Proof. Computing with equivalences:

Hubs and spokes

Why does hubs and spokes work?

For $a_0 : A$ and $p : a_0 = a_0$ we have

$$(\Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h) \simeq (p = 1).$$

Proof. Computing with equivalences:

$$\Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h$$

Hubs and spokes

Why does hubs and spokes work?

For $a_0 : A$ and $p : a_0 = a_0$ we have

$$(\Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h) \simeq (p = 1).$$

Proof. Computing with equivalences:

$$\begin{aligned} & \Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h \\ \simeq & \Sigma(h : A), \Sigma(q : a_0 = h), q =_{\text{loop}}^{\lambda x. \text{circle.rec } a_0 \ p \ x = h} q \end{aligned}$$

Hubs and spokes

Why does hubs and spokes work?

For $a_0 : A$ and $p : a_0 = a_0$ we have

$$(\Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h) \simeq (p = 1).$$

Proof. Computing with equivalences:

$$\begin{aligned} & \Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h \\ \simeq & \Sigma(h : A), \Sigma(q : a_0 = h), q =_{\text{loop}}^{\lambda x. \text{circle.rec } a_0 \ p \ x = h} q \\ \simeq & 1 =_{\text{loop}}^{\lambda x. \text{circle.rec } a_0 \ p \ x = a_0} 1 \end{aligned}$$

Hubs and spokes

Why does hubs and spokes work?

For $a_0 : A$ and $p : a_0 = a_0$ we have

$$(\Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h) \simeq (p = 1).$$

Proof. Computing with equivalences:

$$\begin{aligned} & \Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h \\ \simeq & \Sigma(h : A), \Sigma(q : a_0 = h), q =_{\text{loop}}^{\lambda x. \text{circle.rec } a_0 \ p \ x = h} q \\ \simeq & 1 =_{\text{loop}}^{\lambda x. \text{circle.rec } a_0 \ p \ x = a_0} 1 \\ \simeq & 1 \cdot 1 = \text{ap}_{\text{circle.rec } a_0 \ p}(\text{loop}) \cdot 1 \end{aligned}$$

Hubs and spokes

Why does hubs and spokes work?

For $a_0 : A$ and $p : a_0 = a_0$ we have

$$(\Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h) \simeq (p = 1).$$

Proof. Computing with equivalences:

$$\begin{aligned} & \Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h \\ \simeq & \Sigma(h : A), \Sigma(q : a_0 = h), q =_{\text{loop}}^{\lambda x. \text{circle.rec } a_0 \ p \ x = h} q \\ \simeq & 1 =_{\text{loop}}^{\lambda x. \text{circle.rec } a_0 \ p \ x = a_0} 1 \\ \simeq & 1 \cdot 1 = \text{ap}_{\text{circle.rec } a_0 \ p}(\text{loop}) \cdot 1 \\ \simeq & p = 1 \end{aligned}$$

Torus from quotients

Problem. The hubs-and-spokes-torus is *not* a quotient. The last path constructor refers to previous path constructors, and cannot be written down before you have ℓ_1 and ℓ_2 as paths in the torus.

$$s : \prod(x : S^1), \text{circle.rec } b (\ell_1 \cdot \ell_2 \cdot \ell_1^{-1} \cdot \ell_2^{-1}) x = h$$

Torus from quotients

Problem. The hubs-and-spokes-torus is *not* a quotient. The last path constructor refers to previous path constructors, and cannot be written down before you have ℓ_1 and ℓ_2 as paths in the torus.

$$s : \Pi(x : S^1), \text{circle.rec } b (\ell_1 \cdot \ell_2 \cdot \ell_1^{-1} \cdot \ell_2^{-1}) x = h$$

Solution. Do the constructions in two stages. First define a HIT

HIT pretorus :=

- $\tilde{b} : \text{pretorus}$
- $\tilde{\ell}_1, \tilde{\ell}_2 : b = b$
- $\tilde{h} : \text{pretorus}$

This is a quotient. Now we can define $f : S^1 \rightarrow \text{pretorus}$ by

$$\begin{aligned} f(\text{base}) &::= \tilde{b} \\ \text{ap}_f(\text{loop}) &::= \tilde{\ell}_1 \cdot \tilde{\ell}_2 \cdot \tilde{\ell}_1^{-1} \cdot \tilde{\ell}_2^{-1}. \end{aligned}$$

Torus from quotients

We then define the torus:

HIT $T^2 :=$

- $i : \text{pretorus} \rightarrow T^2$
- $\sigma : \Pi(x : S^1), i(f(x)) = i(h)$

The constructors for T^2 are defined as:

$$\begin{array}{ll} b \equiv i(\tilde{b}) & : T^2 \\ l_i \equiv \text{ap}_i(\tilde{\ell}_i) & : b = b \\ s \equiv ?? & : l_1 \cdot l_2 = l_2 \cdot l_1 \end{array}$$

Torus from quotients

We want to apply the equivalence

$$(\Sigma(h : A), \Pi(x : S^1), \text{circle.rec } a_0 \ p \ x = h) \simeq (p = 1).$$

Let $\tilde{p} = \tilde{l}_1 \cdot \tilde{l}_2 \cdot \tilde{l}_1^{-1} \cdot \tilde{l}_2^{-1}$.

For $x : S^1$ we have

$$\text{circle.rec } (i \ a_0) \ (\text{ap}_i(\tilde{p})) \ x = i(\text{circle.rec } a_0 \ \tilde{p} \ x) \stackrel{\sigma}{=} i(\tilde{h}).$$

From the equivalence we get

$$\text{ap}_i(\tilde{p}) = 1.$$

This gives

$$l_1 \cdot l_2 \cdot l_1^{-1} \cdot l_2^{-1} \equiv \text{ap}_i(\tilde{l}_1) \cdot \text{ap}_i(\tilde{l}_2) \cdot \text{ap}_i(\tilde{l}_1)^{-1} \cdot \text{ap}_i(\tilde{l}_2)^{-1} = 1.$$

which gives

$$s : l_1 \cdot l_2 = l_2 \cdot l_1.$$

- This is just the definition of the constructors! We still need to define the induction principle, and computation rules.
- It should be a little bit simpler in a cubical type theory. However, most steps do not become definitional even in cubical type theory.
- We don't prove the computation rule for the induction principle on 2-paths. However, this is not needed to characterize T^2 up to equivalence.
- We don't just define the torus, but a wide class of 2-HITs. The 2-path constructor must be an equality between
 - ▶ 1-path constructors;
 - ▶ A point constructor $f : A \rightarrow X$ applied to a path in A ;
 - ▶ reflexivity;
 - ▶ concatenations/inverses of such paths.

More 2-HITs

Given $A : \mathcal{U}$ and $R : A \rightarrow A \rightarrow \mathcal{U}$ we define *words* in R to be inductive $\bar{R} : A \rightarrow A \rightarrow \mathcal{U} :=$

- $[-] : \prod(a, a' : A), R(a, a') \rightarrow \bar{R}(a, a')$
- $\rho : \prod(a, a' : A), a = a' \rightarrow \bar{R}(a, a')$
- $-^{-1} : \prod(a, a' : A), \bar{R}(a, a') \rightarrow \bar{R}(a', a)$
- $- \cdot - : \prod(a, a', a'' : A), \bar{R}(a, a') \rightarrow \bar{R}(a', a'') \rightarrow \bar{R}(a, a'')$

Then, if we have a map

$$p : \prod(a, a' : A), R(a, a') \rightarrow i(a) = i(a')$$

we can extend it to a map

$$\bar{p} : \prod(a, a' : A), \bar{R}(a, a') \rightarrow i(a) = i(a')$$

If we are also given a “relation over \bar{R} ,” i.e. a family

$$Q : \prod(a, a' : A), \bar{R}(a, a') \rightarrow \bar{R}(a, a') \rightarrow \mathcal{U}$$

Then we define the following 2-HIT:

HIT two-quotient(A, R, Q) : $\mathcal{U} :=$

- $i : A \rightarrow \text{two-quotient}(A, R, Q)$
- $p : \prod(a, a' : A), R(a, a') \rightarrow i(a) = i(a')$
- $r : \prod(a, a' : A)(r, r' : \bar{R}(a, a')), Q(r, r') \rightarrow \bar{p}(r) = \bar{p}(r')$

More 2-HITs

Example: $K(G, 1) := \|\text{two-quotient}(A, R, Q)\|_1$ with

$$A := 1$$

$$R(\star, \star) := G$$

and Q is an inductive family with 1 constructor, namely:

$$q : \Pi(g_1, g_2 : G), Q([g_1 * g_2], [g_1] \cdot [g_2])$$

HIT $K(G, 1) :=$

- $b : K(G, 1)$
- $p : G \rightarrow b = b$
- $m : \Pi(g, h : G), p(gh) = p(g) \cdot p(h)$
- $K(G, 1)$ is 1-truncated

Conclusions

- We can reduce a wide class of HITs to quotients.
- Are there HITs which we cannot reduce to quotients?
 - ▶ I don't know
 - ▶ There are certainly HITs where I have no idea *how* to reduce them. (e.g. arbitrary localizations)

Thank you