Constructing 1-Truncated Finitary Higher Inductive Types as Groupoid Quotients

Niels van der Weide

Radboud University, Nijmegen, The Netherlands

February 6, 2020

What are higher inductive types?

Higher inductive types: define types by describing constructors for the points, paths, 2-paths (paths between paths), ...
Examples (spaces):

$\texttt{Inductive}\ S^1 :=$
$|\ \mathbf{base}_{S^1} : S^1$
$|\ \mathbf{loop}_{S^1} : \mathbf{base}_{S^1} = \mathbf{base}_{S^1}$

$\texttt{Inductive}\ \mathcal{T}^2 :=$
$|\ \mathbf{base} : \mathcal{T}^2$
$|\ \mathbf{loop_l}, \mathbf{loop_r} : \mathbf{base} = \mathbf{base}$
$|\ \mathbf{surf} : \mathbf{loop_l} \bullet \mathbf{loop_r} = \mathbf{loop_r} \bullet \mathbf{loop_l}$

More Examples!

In general, one can have recursive constructors (both for the points and paths).

$\texttt{Inductive}\ \mathbb{Z}_2 :=$
$|\ \mathbf{Z} : \mathbb{Z}_2$
$|\ \mathbf{S} : \mathbb{Z}_2 \to \mathbb{Z}_2$
$|\ \mathbf{m} : \prod(x : \mathbb{Z}_2), \mathbf{S}(\mathbf{S}(x)) = x$
$|\ \mathbf{c} : \prod(x : \mathbb{Z}_2), \mathbf{m}(\mathbf{S}(x)) = \mathsf{ap}\ \mathbf{S}\ (\mathbf{m}(x))$

$\texttt{Inductive}\ ||A|| :=$
$|\ \mathbf{inc} : A \to ||A||$
$|\ \mathbf{trunc} : \prod(x, y : ||A||)(p, q : x = y), p = q$

The terminology from the title

- **Finitary**: each constructor only has a finite number of recursive arguments (arguments are described by a finitary polynomial).

The terminology from the title

- **Finitary**: each constructor only has a finite number of recursive arguments (arguments are described by a finitary polynomial). For example, H_1 is finitary while H_2 isn't.

  Inductive $H_1 :=$
  $\mid\ c_1 : H_1 \times H_1 \to H_1$
  $\mid\ p_1 : \prod(x, y : H_1), c_1(x, y) = c_1(y, x)$

  Inductive $H_2 :=$
  $\mid\ c_2 : (\mathbb{N} \to H_2) \to H_2$
  $\mid\ p_2 : \prod(f : \mathbb{N} \to H_2), c_2(f) = c_2(\lambda n, f(n + 1))$

The terminology from the title

- **Finitary**: each constructor only has a finite number of recursive arguments (arguments are described by a finitary polynomial). For example, H_1 is finitary while H_2 isn't.

  $$\begin{array}{l}\texttt{Inductive } H_1 := \\ \mid\ c_1 : H_1 \times H_1 \to H_1 \\ \mid\ p_1 : \prod(x, y : H_1), c_1(x, y) = c_1(y, x)\end{array}$$

  $$\begin{array}{l}\texttt{Inductive } H_2 := \\ \mid\ c_2 : (\mathbb{N} \to H_2) \to H_2 \\ \mid\ p_2 : \prod(f : \mathbb{N} \to H_2), c_2(f) = c_2(\lambda n, f(n+1))\end{array}$$

- **1-truncated**: a type X is 1-truncated if for all $x, y : X$, $p, q : x = y$, and $r, s : p = q$ we have $r = s$.

The terminology from the title

- **Finitary**: each constructor only has a finite number of recursive arguments (arguments are described by a finitary polynomial). For example, H_1 is finitary while H_2 isn't.

  $$\begin{aligned}&\texttt{Inductive}\ H_1 := \\ &\mid\ c_1 : H_1 \times H_1 \to H_1 \\ &\mid\ p_1 : \prod(x, y : H_1), c_1(x, y) = c_1(y, x)\end{aligned}$$

  $$\begin{aligned}&\texttt{Inductive}\ H_2 := \\ &\mid\ c_2 : (\mathbb{N} \to H_2) \to H_2 \\ &\mid\ p_2 : \prod(f : \mathbb{N} \to H_2), c_2(f) = c_2(\lambda n, f(n+1))\end{aligned}$$

- **1-truncated**: a type X is 1-truncated if for all $x, y : X$, $p, q : x = y$, and $r, s : p = q$ we have $r = s$.
- **Groupoid quotient**: a HIT that takes a groupoid and turns it into a 1-type (we will discuss it more formally later this talk)

Problem Statement and the Main Theorem

Goal: reduce finitary 1-truncated HITs to simpler principles.

Problem Statement and the Main Theorem

Goal: reduce finitary 1-truncated HITs to simpler principles.
More specifically, we

- define inside of type theory the notion of a signature for HITs (allows points, paths, and 2-path constructors)
- define the introduction, elimination, and computation rules for each signature

HIT in 1-types: a 1-type that satisfies all these rules.

Problem Statement and the Main Theorem

Goal: reduce finitary 1-truncated HITs to simpler principles.
More specifically, we

- define inside of type theory the notion of a signature for HITs (allows points, paths, and 2-path constructors)
- define the introduction, elimination, and computation rules for each signature

HIT in 1-types: a 1-type that satisfies all these rules.
Then we prove

Theorem
In a type theory with the groupoid quotient, each signature has a HIT in 1-types.

Formalization

All results in this talk are formalized over the UniMath library.

https://github.com/nmvdw/GrpdHITs

The topics of this talk

- As a starter, we look at the theorem in the set truncated case.
- How to move from this case to the 1-truncated case?
- The 1-truncated case:
  - Signature for HITs
  - Bicategories of algebras (1-types and groupoids)
  - The groupoid quotient
  - Lifting the groupoid quotient to a biadjunction between algebras
- Conclusion and outlook

How to construct set-truncated HITs

Goal: construct set-truncated HITs as a quotient.
For this construction, we

- Define signatures for set-truncated HITs
- Define categories of algebras in sets and setoids
- Prove initial algebra semantics: initiality implies induction
- Lift the quotient to a adjunction between the category of algebras in sets and in setoids
- Construct the initial algebra in setoids

How to construct set-truncated HITs

Goal: construct set-truncated HITs as a quotient.
For this construction, we

- Define signatures for set-truncated HITs
- Define categories of algebras in sets and setoids
- Prove initial algebra semantics: initiality implies induction
- Lift the quotient to a adjunction between the category of algebras in sets and in setoids
- Construct the initial algebra in setoids

Scheme for set-truncated HITs

Our goal is to construct HITs of the following shape

$\texttt{Inductive}\ H :=$
$|\ \ c : P(H) \to H$
$|\ \ p : \prod(j : J)(x : Q_j(H)), l_j(x) = r_j(x)$
$|\ \ t : \prod(x, y : H)(p, q : x = y), p = q$

Scheme for set-truncated HITs

Our goal is to construct HITs of the following shape

`Inductive` $H :=$
$\mid\ c : P(H) \to H$
$\mid\ p : \prod(j : J)(x : Q_j(H)), l_j(x) = r_j(x)$
$\mid\ t : \prod(x, y : H)(p, q : x = y), p = q$

What are P, Q_j, l_j, and r_j?

Signatures for Set-HITs: the point constructors

Definition (Polynomials)

The type P of **finitary polynomials** is inductively generated by

$$\mathbf{C}(A) : \mathsf{P}, \quad \mathbf{I} : \mathsf{P}, \quad P_1 + P_2 : \mathsf{P}, \quad P_1 \times P_2 : \mathsf{P}$$

where A is a set and P_1 and P_2 are arbitrary polynomials.

Signatures for Set-HITs: the point constructors

Definition (Polynomials)

The type P of **finitary polynomials** is inductively generated by

$$\mathbf{C}(A) : \mathsf{P}, \quad \mathbf{I} : \mathsf{P}, \quad P_1 + P_2 : \mathsf{P}, \quad P_1 \times P_2 : \mathsf{P}$$

where A is a set and P_1 and P_2 are arbitrary polynomials.

A polynomial represents a functor $[\![P]\!]$ on sets.
Given a polynomial P and a set X, we get a set $P(X)$.

Signatures for Set-HITs: the path constructors

Definition (Path endpoints)

Let A, S, and T be polynomials The type $\mathsf{E}_A(S, T)$ of **path endpoints** with arguments A, source S, and target T is inductively generated by the constructors given on the next slide.

Signatures for Set-HITs: the path constructors

Definition (Path endpoints)

Let A, S, and T be polynomials The type $\mathsf{E}_A(S, T)$ of **path endpoints** with arguments A, source S, and target T is inductively generated by the constructors given on the next slide.

Given X with $c : A(X) \to X$, a path endpoint $e : \mathsf{E}_A(S, T)$ represents a function $S(X) \to T(X)$ which can make use of c.

Signatures for Set-HITs: some endpoints

$$\frac{P : \mathsf{P}}{\mathbf{id}_A : \mathsf{E}_A(P, P)}$$

$$\frac{P, Q, R : \mathsf{P} \qquad e_1 : \mathsf{E}_A(P, Q) \qquad e_2 : \mathsf{E}_A(Q, R)}{e_1 \cdot e_2 : \mathsf{E}_A(P, R)}$$

$$\mathbf{constr} : \mathsf{E}_A(A, \mathsf{I})$$

Signatures for Set-HITs: putting it together

Definition (HIT-signature (for set-truncated HITs))

A **HIT-signature** Σ consists of

- A polynomial A^{Σ}
- A type $\mathsf{J}^{\Sigma}_{\mathsf{P}}$ together with for each $j : \mathsf{J}^{\Sigma}_{\mathsf{P}}$ a polynomial S^{Σ}_{j} and endpoints $\mathsf{l}^{\Sigma}_{j}, \mathsf{r}^{\Sigma}_{j} : \mathsf{E}_{\mathsf{A}^{\Sigma}}(\mathsf{S}^{\Sigma}_{j}, \mathsf{I})$

Σ represents the following HIT

```
Inductive H :=
| c : A^Σ(H) → H
| p : ∏(j : J^Σ_P)(x : S^Σ_j(H)), l^Σ_j(x) = r^Σ_j(x)
| t : ∏(x, y : H)(p, q : x = y), p = q
```

Signatures for Set-HITs: putting it together

Definition (HIT-signature (for set-truncated HITs))

A **HIT-signature** Σ consists of

- A polynomial A^Σ
- A type $\mathsf{J}_\mathsf{P}^\Sigma$ together with for each $j : \mathsf{J}_\mathsf{P}^\Sigma$ a polynomial S_j^Σ and endpoints $\mathsf{l}_j^\Sigma, \mathsf{r}_j^\Sigma : \mathsf{E}_{\mathsf{A}^\Sigma}(\mathsf{S}_j^\Sigma, \mathsf{I})$

Σ represents the following HIT

```
Inductive H :=
| c : A^Σ(H) → H
| p : ∏(j : J_P^Σ)(x : S_j^Σ(H)), l_j^Σ(x) = r_j^Σ(x)
| t : ∏(x, y : H)(p, q : x = y), p = q
```

Signatures for Set-HITs: putting it together

Definition (HIT-signature (for set-truncated HITs))

A **HIT-signature** Σ consists of

- A polynomial A^{Σ}
- A type $\mathsf{J}_{\mathsf{P}}^{\Sigma}$ together with for each $j : \mathsf{J}_{\mathsf{P}}^{\Sigma}$ a polynomial S_j^{Σ} and endpoints $\mathsf{l}_j^{\Sigma}, \mathsf{r}_j^{\Sigma} : \mathsf{E}_{\mathsf{A}^{\Sigma}}(\mathsf{S}_j^{\Sigma}, \mathsf{I})$

Σ represents the following HIT

```
Inductive H :=
| c : A^Σ(H) → H
| p : ∏(j : J_P^Σ)(x : S_j^Σ(H)), l_j^Σ(x) = r_j^Σ(x)
| t : ∏(x, y : H)(p, q : x = y), p = q
```

Signatures for Set-HITs: putting it together

Definition (HIT-signature (for set-truncated HITs))

A **HIT-signature** Σ consists of

- A polynomial A^{Σ}
- A type $\mathsf{J}_{\mathsf{P}}^{\Sigma}$ together with for each $j : \mathsf{J}_{\mathsf{P}}^{\Sigma}$ a polynomial S_j^{Σ} and endpoints $\mathsf{l}_j^{\Sigma}, \mathsf{r}_j^{\Sigma} : \mathsf{E}_{\mathsf{A}^{\Sigma}}(\mathsf{S}_j^{\Sigma}, \mathsf{I})$

Σ represents the following HIT

```
Inductive H :=
| c : A^Σ(H) → H
| p : ∏(j : J_P^Σ)(x : S_j^Σ(H)), l_j^Σ(x) = r_j^Σ(x)
| t : ∏(x, y : H)(p, q : x = y), p = q
```

Signatures for Set-HITs: putting it together

Definition (HIT-signature (for set-truncated HITs))

A **HIT-signature** Σ consists of

- A polynomial A^{Σ}
- A type $\mathsf{J}_{\mathsf{P}}^{\Sigma}$ together with for each $j : \mathsf{J}_{\mathsf{P}}^{\Sigma}$ a polynomial S_j^{Σ} and endpoints $\mathsf{l}_j^{\Sigma}, \mathsf{r}_j^{\Sigma} : \mathsf{E}_{\mathsf{A}^{\Sigma}}(\mathsf{S}_j^{\Sigma}, \mathsf{I})$

Σ represents the following HIT

$\texttt{Inductive}\ H :=$
$|\ \ c : \mathsf{A}^{\Sigma}(H) \to H$
$|\ \ p : \prod(j : \mathsf{J}_{\mathsf{P}}^{\Sigma})(x : \mathsf{S}_j^{\Sigma}(H)), \mathsf{l}_j^{\Sigma}(x) = \mathsf{r}_j^{\Sigma}(x)$
$|\ \ t : \prod(x, y : H)(p, q : x = y), p = q$

Let us recall the structure of the argument

- ~~Define signatures for set-truncated HITs~~
- **Define categories of algebras in sets and setoids**
- Prove initial algebra semantics: initiality implies induction
- Lift the quotient to a adjunction between the category of algebras in sets and in setoids
- Construct the initial algebra in setoids

Algebras for HITs

An algebra X for Σ that describes

$\texttt{Inductive}\ H :=$
$|\ \ c : \mathsf{A}^{\Sigma}(H) \rightarrow H$
$|\ \ p : \prod(j : \mathsf{J}_{\mathsf{P}}^{\Sigma})(x : \mathsf{S}_{j}^{\Sigma}(H)), \mathsf{l}_{j}^{\Sigma}(x) = \mathsf{r}_{j}^{\Sigma}(x)$
$|\ \ t : \prod(x, y : H)(p, q : x = y), p = q$

consists of

- A set X
- An operation $\mathsf{c}^{\mathsf{X}} : \mathsf{A}^{\Sigma}(X) \rightarrow X$
- For each $j : \mathsf{J}_{\mathsf{P}}^{\Sigma}$ and $x : \mathsf{S}_{j}^{\Sigma}(H)$, a path $\mathsf{p}_{j}^{\mathsf{X}} : \mathsf{l}_{j}^{\Sigma}(x) = \mathsf{r}_{j}^{\Sigma}(x)$

Goal: define a category of algebras.

Algebras for HITs

An algebra X for Σ that describes

$\texttt{Inductive}\ H :=$
$|\ c : \mathsf{A}^{\Sigma}(H) \to H$
$|\ p : \prod(j : \mathsf{J}^{\Sigma}_{\mathsf{P}})(x : \mathsf{S}^{\Sigma}_{j}(H)), \mathsf{l}^{\Sigma}_{j}(x) = \mathsf{r}^{\Sigma}_{j}(x)$
$|\ t : \prod(x, y : H)(p, q : x = y), p = q$

consists of

- A set X
- An operation $\mathsf{c}^{\mathsf{X}} : \mathsf{A}^{\Sigma}(X) \to X$
- For each $j : \mathsf{J}^{\Sigma}_{\mathsf{P}}$ and $x : \mathsf{S}^{\Sigma}_{j}(H)$, a path $\mathsf{p}^{\mathsf{X}}_{j} : \mathsf{l}^{\Sigma}_{j}(x) = \mathsf{r}^{\Sigma}_{j}(x)$

Goal: define a category of algebras.

Algebras for HITs

An algebra X for Σ that describes

$\texttt{Inductive}\ H :=$
$\mid\ c : \mathsf{A}^{\Sigma}(H) \rightarrow H$
$\mid\ p : \prod(j : \mathsf{J}^{\Sigma}_{\mathsf{P}})(x : \mathsf{S}^{\Sigma}_{j}(H)), \mathsf{l}^{\Sigma}_{j}(x) = \mathsf{r}^{\Sigma}_{j}(x)$
$\mid\ t : \prod(x, y : H)(p, q : x = y), p = q$

consists of

- A set X
- An operation $\mathsf{c}^{\mathsf{X}} : \mathsf{A}^{\Sigma}(X) \rightarrow X$
- For each $j : \mathsf{J}^{\Sigma}_{\mathsf{P}}$ and $x : \mathsf{S}^{\Sigma}_{j}(H)$, a path $\mathsf{p}^{\mathsf{X}}_{j} : \mathsf{l}^{\Sigma}_{j}(x) = \mathsf{r}^{\Sigma}_{j}(x)$

Goal: define a category of algebras.

Algebras for HITs

An algebra X for Σ that describes

$\texttt{Inductive}\ H :=$
$|\ c : \mathsf{A}^{\Sigma}(H) \to H$
$|\ p : \prod(j : \mathsf{J}^{\Sigma}_{\mathsf{P}})(x : \mathsf{S}^{\Sigma}_{j}(H)), \mathsf{l}^{\Sigma}_{j}(x) = \mathsf{r}^{\Sigma}_{j}(x)$
$|\ t : \prod(x, y : H)(p, q : x = y), p = q$

consists of

- A set X
- An operation $\mathsf{c}^{\mathsf{X}} : \mathsf{A}^{\Sigma}(X) \to X$
- For each $j : \mathsf{J}^{\Sigma}_{\mathsf{P}}$ and $x : \mathsf{S}^{\Sigma}_{j}(H)$, a path $\mathsf{p}^{\mathsf{X}}_{j} : \mathsf{l}^{\Sigma}_{j}(x) = \mathsf{r}^{\Sigma}_{j}(x)$

Goal: define a category of algebras.

Algebras for HITs: Categorical Constructions

Defining the category of algebras is done in 2 steps.

Definition

Given a category C, an endofunctor F on C, we have a category $\mathrm{Falg}(F)$ whose objects are pairs $X : \mathrm{C}$ and $f : F(X) \to X$.

Algebras for HITs: Categorical Constructions

Defining the category of algebras is done in 2 steps.

Definition

Given a category C, an endofunctor F on C, we have a category $\mathsf{Falg}(F)$ whose objects are pairs $X : \mathsf{C}$ and $f : F(X) \to X$.

Definition

Given a category C and a predicate P on the objects of C, we have a category $\mathsf{FSub}(\mathsf{C}, P)$ whose objects are X with a proof of $P(X)$.

Algebras for HITs: Categorical Constructions

Defining the category of algebras is done in 2 steps.

Definition

Given a category C, an endofunctor F on C, we have a category $\mathsf{Falg}(F)$ whose objects are pairs $X : \mathsf{C}$ and $f : F(X) \rightarrow X$.

Definition

Given a category C and a predicate P on the objects of C, we have a category $\mathsf{FSub}(\mathsf{C}, P)$ whose objects are X with a proof of $P(X)$.

- $\mathsf{Falg}(F)$ adds the point constructor
- $\mathsf{FSub}(\mathsf{C}, P)$ adds the path constructor

Algebras for HITs: the point constructor

Definition
Given a polynomial P, we get a functor $[\![P]\!] : \mathrm{Sets} \to \mathrm{Sets}$.

Definition
Given a signature Σ, define the category $\mathrm{PreAlg}(\Sigma)$ of **prealgebras** of Σ to be $\mathrm{Falg}([\![\mathsf{A}^{\Sigma}]\!])$.

Write U for the forgetful functor from $\mathrm{PreAlg}(\Sigma)$ to Sets.

Algebras for HITs: the path constructor

Definition

Suppose, we have an endpoint $e : \mathsf{E}_{\mathsf{A}^\Sigma}(S, T)$. Note that we have

$$\mathsf{PreAlg}(\Sigma) \xrightarrow{U} \mathsf{Sets} \underset{[\![T]\!]}{\overset{[\![S]\!]}{\rightrightarrows}} \mathsf{Sets}$$

Then we get a natural transformation $[\![e]\!] : [\![S]\!] \circ U \Rightarrow [\![T]\!] \circ U$.

Algebras for HITs: the path constructor

Definition

Suppose, we have an endpoint $e : \mathsf{E}_{\mathsf{A}^\Sigma}(S, T)$. Note that we have

$$\mathrm{PreAlg}(\Sigma) \xrightarrow{U} \mathrm{Sets} \underset{[\![T]\!]}{\overset{[\![S]\!]}{\rightrightarrows}} \mathrm{Sets}$$

Then we get a natural transformation $[\![e]\!] : [\![S]\!] \circ U \Rightarrow [\![T]\!] \circ U$.

Definition

The category $\mathrm{Alg}(\Sigma)$ of **algebras** on Σ is define to be the full subcategory of $\mathrm{PreAlg}(\Sigma)$ such that each object (X, c) satisfies:

$$\text{for all } j : \mathsf{J}_{\mathsf{P}}^{\Sigma} \text{ and } x : \mathsf{S}_j^{\Sigma}(X) \text{ we have } [\![\mathsf{l}_j^{\Sigma}]\!]x = [\![\mathsf{r}_j^{\Sigma}]\!]x$$

Algebras for HITs: in setoids

Similarly, we define

- $\mathsf{PreAlg}_{\mathsf{Setoid}}(\Sigma)$: prealgebras in setoids
- $\mathsf{Alg}_{\mathsf{Setoid}}(\Sigma)$: algebras in setoids

Let us recall the structure of the argument

- ~~Define signatures for set-truncated HITs~~
- ~~Define categories of algebras in sets and setoids~~
- ~~Prove initial algebra semantics: initiality implies induction~~
- **Lift the quotient to a adjunction between the category of algebras in sets and in setoids**
- Construct the initial algebra in setoids

Recall the quotient type

Given a set X and an equivalence relation R on X, we define X/R as the following HIT.

```
Inductive X/R :=
| class : X → X/R
| eqclass : ∏(x, y : X), R(x, y) → class(x) = class(y)
| trunc : ∏(x, y : H)(p, q : x = y), p = q
```

The quotient gives an adjunction

Write

- Sets for the category of sets with functions
- Setoid for the category of setoids with functions that preserve the relation

Then

- We have a functor $\mathsf{Quot} : \mathsf{Setoid} \to \mathsf{Sets}$
- We have a functor $\mathsf{PathSetoid} : \mathsf{Sets} \to \mathsf{Setoid}$
- We have an adjunction $\mathsf{Quot} \dashv \mathsf{PathSetoid}$

(Rijke, Spitters, 2015)

Lifting the quotient

Proposition (Hermida and Jacobs, 1998, Theorem 2.14)

We have

- *a functor* $\mathsf{Quot}_{\mathsf{PreAlg}} : \mathsf{PreAlg}_{\mathsf{Setoid}}(\Sigma) \to \mathsf{PreAlg}(\Sigma)$
- *a functor* $\mathsf{PathSetoid}_{\mathsf{PreAlg}} : \mathsf{PreAlg}(\Sigma) \to \mathsf{PreAlg}_{\mathsf{Setoid}}(\Sigma)$
- *an adjunction* $\mathsf{Quot}_{\mathsf{PreAlg}} \dashv \mathsf{PathSetoid}_{\mathsf{PreAlg}}$

Use: polynomial functors commutes with quotients.
Needs: P is finitary! (Chapman, Uustalu, Veltri)

Lifting the quotient

Proposition (Hermida and Jacobs, 1998, Theorem 2.14)

We have

- *a functor* $\mathsf{Quot}_{\mathsf{PreAlg}} : \mathsf{PreAlg}_{\mathsf{Setoid}}(\Sigma) \to \mathsf{PreAlg}(\Sigma)$
- *a functor* $\mathsf{PathSetoid}_{\mathsf{PreAlg}} : \mathsf{PreAlg}(\Sigma) \to \mathsf{PreAlg}_{\mathsf{Setoid}}(\Sigma)$
- *an adjunction* $\mathsf{Quot}_{\mathsf{PreAlg}} \dashv \mathsf{PathSetoid}_{\mathsf{PreAlg}}$

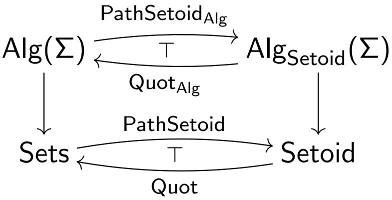
Use: polynomial functors commutes with quotients.
Needs: P is finitary! (Chapman, Uustalu, Veltri)

Proposition

We have

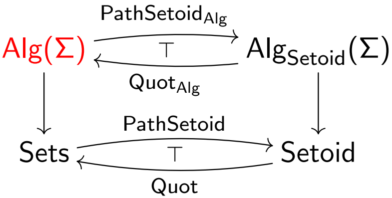
- *a functor* $\mathsf{Quot}_{\mathsf{Alg}} : \mathsf{Alg}_{\mathsf{Setoid}}(\Sigma) \to \mathsf{Alg}(\Sigma)$
- *a functor* $\mathsf{PathSetoid}_{\mathsf{Alg}} : \mathsf{Alg}(\Sigma) \to \mathsf{Alg}_{\mathsf{Setoid}}(\Sigma)$
- *an adjunction* $\mathsf{Quot}_{\mathsf{Alg}} \dashv \mathsf{PathSetoid}_{\mathsf{Alg}}$

Concluding the set truncated case



To construct a HIT on Σ, we do

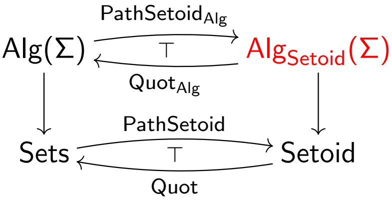
Concluding the set truncated case



To construct a HIT on Σ, we do

- By initial algebra semantics, find initial object in $\mathsf{Alg}(\Sigma)$

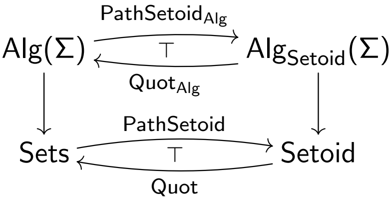
Concluding the set truncated case



To construct a HIT on Σ, we do

- By initial algebra semantics, find initial object in $\mathsf{Alg}(\Sigma)$
- By adjunction, find initial object in $\mathsf{Alg}_{\mathsf{Setoid}}(\Sigma)$

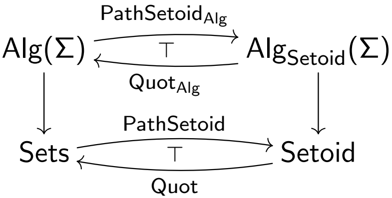
Concluding the set truncated case



To construct a HIT on Σ, we do

- By initial algebra semantics, find initial object in Alg(Σ)
- By adjunction, find initial object in $\mathsf{Alg}_{\mathsf{Setoid}}(\Sigma)$
- Technical, see formalization and (Moeneclaey, internship report)

Concluding the set truncated case



To construct a HIT on Σ, we do

- By initial algebra semantics, find initial object in $\mathsf{Alg}(\Sigma)$
- By adjunction, find initial object in $\mathsf{Alg}_{\mathsf{Setoid}}(\Sigma)$
- Technical, see formalization and (Moeneclaey, internship report)

Hence:

Each signature Σ has a HIT in sets.

Now let's do this for 1-types

Recall our main theorem:

Theorem

In a type theory with the groupoid quotient, each signature has a HIT in 1-types.

We use a similar approach.

Sets versus 1-Types

To translate our argument for sets to 1-types, we use the following table

Sets	1-types
Setoids	Groupoids
Quotient	Groupoid quotient
Category	Bicategory
Initial object	Biinitial object
Functor	Pseudofunctor
Natural transformation	Pseudotransformation
Adjunction	Biadjunction

Sets versus 1-Types

To translate our argument for sets to 1-types, we use the following table

Sets	1-types
Setoids	Groupoids
Quotient	Groupoid quotient
Category	Bicategory
Initial object	Biinitial object
Functor	Pseudofunctor
Natural transformation	Pseudotransformation
Adjunction	Biadjunction

Recall:

- the bicategory 1-Type: 1-types with functions and paths between functions
- the bicategory Grpd of groupoids

The approach for set-truncated HITs

This construction is done as follows:

- Define signatures for set-truncated HITs
- Define categories of algebras in sets and setoids
- Prove initial algebra semantics: initiality implies induction
- Lift the quotient to a adjunction between the category of algebras in sets and in setoids
- Construct the initial algebra in setoids

The approach for **1-truncated** HITs

This construction is done as follows:

- Define signatures for **1-truncated** HITs
- Define **bicategories** of algebras in **1-types** and **groupoids**
- Prove **biinitial** algebra semantics: **biinitiality** implies induction
- Lift the **groupoid quotient** to a **biadjunction** between the **bicategory** of algebras in **1-types** and in **groupoids**
- Construct the **biinitial** algebra in **groupoids**

The approach for 1-truncated HITs

This construction is done as follows:

- Define signatures for **1-truncated** HITs
- Define **bicategories** of algebras in **1-types** and **groupoids**
- Prove **biinitial** algebra semantics: **biinitiality** implies induction
- Lift the **groupoid quotient** to a **biadjunction** between the **bicategory** of algebras in **1-types** and in **groupoids**
- Construct the **biinitial** algebra in **groupoids**

The HITs we consider

Similar to Dybjer, Moeneclaey, 2018.

`Inductive` $H :=$

$| \ c : \mathsf{A}(H) \to H$

$| \ p : \prod (j : \mathsf{J}_\mathsf{P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$

$| \ s : \prod (j : \mathsf{J}_\mathsf{H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)),$

$\quad \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$

$| \ t : \prod (x, y : H)(p, q : x = y)(r, s : p = q), r = s$

The HITs we consider

Similar to Dybjer, Moeneclaey, 2018.

```
Inductive H :=
```
$|\ c : \mathsf{A}(H) \to H$

$|\ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$

$|\ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)),$
$\quad \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$

$|\ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

Note: a part is similar to set-truncated HITs

The HITs we consider

Similar to Dybjer, Moeneclaey, 2018.

`Inductive` $H :=$
$\mid \; c : \mathsf{A}(H) \to H$
$\mid \; p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$
$\mid \; s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)),$
$\quad \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$
$\mid \; t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

Note: a part is similar to set-truncated HITs
What are a_j, b_j, p_j, and q_j?

The path argument

In general, a 2-path constructor could have any finite number of path arguments

```
Inductive ||A|| :=
| inc : A → ||A||
| trunc : ∏(x, y : ||A||)(p, q : x = y), p = q
```

The path argument

In general, a 2-path constructor could have any finite number of path arguments

`Inductive` $||A|| :=$
$|$ **inc** : $A \to ||A||$
$|$ **trunc** : $\prod(x, y : ||A||)(p, q : x = y), p = q$

However, it suffices to assume there is 1 path argument.

`Inductive` $||A|| :=$
$|$ **inc** : $A \to ||A||$
$|$ **trunc** : $\prod(x, y : ||A||)(p : (x, x) = (y, y)), \mathsf{ap}\ \pi_1\ p = \mathsf{ap}\ \pi_2\ q$

The path argument

In general, a 2-path constructor could have any finite number of path arguments

```
Inductive ||A|| :=
| inc : A → ||A||
| trunc : ∏(x, y : ||A||)(p, q : x = y), p = q
```

However, it suffices to assume there is 1 path argument.

```
Inductive ||A|| :=
| inc : A → ||A||
| trunc : ∏(x, y : ||A||)(p : (x, x) = (y, y)), ap π1 p = ap π2 q
```

So: we represent the path argument by two path endpoints.

Signatures for 1-Truncated HITs : Homotopy Endpoints

$\texttt{Inductive}\ H :=$
$\mid\ c : \mathsf{A}(H) \to H$
$\mid\ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$
$\mid\ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)),$
$\quad \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$
$\mid\ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

The type of homotopy endpoints depends on

- a polynomial A

Signatures for 1-Truncated HITs : Homotopy Endpoints

$\texttt{Inductive}\ H :=$
$|\ c : \mathsf{A}(H) \to H$
$|\ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$
$|\ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)),$
$\quad \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$
$|\ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

The type of homotopy endpoints depends on

- a polynomial A
- a type $\mathsf{J_P}$

Signatures for 1-Truncated HITs : Homotopy Endpoints

$\texttt{Inductive}\ H :=$
$|\ c : \mathsf{A}(H) \to H$
$|\ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$
$|\ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)),$
$\quad \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$
$|\ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

The type of homotopy endpoints depends on

- a polynomial A
- a type $\mathsf{J_P}$
- for each $j : \mathsf{J_P}$ a polynomial S_j and path endpoints l_j and r_j

Signatures for 1-Truncated HITs : Homotopy Endpoints

Inductive $H :=$
$\mid\ c : \mathsf{A}(H) \to H$
$\mid\ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$
$\mid\ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)),$
$\mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$
$\mid\ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

The type of homotopy endpoints depends on

- a polynomial A
- a type $\mathsf{J_P}$
- for each $j : \mathsf{J_P}$ a polynomial S_j and path endpoints l_j and r_j
- a polynomial R

Signatures for 1-Truncated HITs : Homotopy Endpoints

$\texttt{Inductive}\ H :=$
$|\ \ c : \mathsf{A}(H) \to H$
$|\ \ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$
$|\ \ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)),$
$\quad \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$
$|\ \ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

The type of homotopy endpoints depends on

- a polynomial A
- a type $\mathsf{J_P}$
- for each $j : \mathsf{J_P}$ a polynomial S_j and path endpoints l_j and r_j
- a polynomial R
- a polynomial T and path endpoints $\mathsf{a}, \mathsf{b} : \mathsf{E_A}(\mathsf{R}, T)$

Signatures for 1-Truncated HITs : Homotopy Endpoints

`Inductive` $H :=$
$\mid\ c : \mathsf{A}(H) \to H$
$\mid\ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$
$\mid\ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)),$
$\quad \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$
$\mid\ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

The type of homotopy endpoints depends on

- a polynomial A
- a type $\mathsf{J_P}$
- for each $j : \mathsf{J_P}$ a polynomial S_j and path endpoints l_j and r_j
- a polynomial R
- a polynomial T and path endpoints $\mathsf{a}, \mathsf{b} : \mathsf{E_A}(\mathsf{R}, T)$
- path endpoints $\mathsf{s}, \mathsf{t} : \mathsf{E_A}(\mathsf{R}, \mathsf{I})$

Signatures for 1-Truncated HITs : Homotopy Endpoints

Given

- a polynomial A
- a type $\mathsf{J_P}$
- for each $j : \mathsf{J_P}$ a polynomial S_j and path endpoints l_j and r_j
- a polynomial R
- a polynomial T and path endpoints $\mathsf{a}, \mathsf{b} : \mathsf{E_A}(\mathsf{R}, T)$
- path endpoints $\mathsf{s}, \mathsf{t} : \mathsf{E_A}(\mathsf{R}, \mathsf{I})$

we define the type $\mathsf{H}_{\mathsf{l}_j, \mathsf{r}_j, \mathsf{a}, \mathsf{b}}(\mathsf{s}, \mathsf{t})$ inductively.

Signatures for 1-Truncated HITs : Homotopy Endpoints

Given

- a polynomial A
- a type $\mathsf{J_P}$
- for each $j : \mathsf{J_P}$ a polynomial S_j and path endpoints l_j and r_j
- a polynomial R
- a polynomial T and path endpoints $\mathsf{a}, \mathsf{b} : \mathsf{E_A}(\mathsf{R}, T)$
- path endpoints $\mathsf{s}, \mathsf{t} : \mathsf{E_A}(\mathsf{R}, \mathsf{I})$

we define the type $\mathsf{H}_{\mathsf{l}_j,\mathsf{r}_j,\mathsf{a},\mathsf{b}}(\mathsf{s}, \mathsf{t})$ inductively.
Given $x : \mathsf{R}(X)$ and $w : \mathsf{a}(x) = \mathsf{b}(x)$, a homotopy endpoint $h : \mathsf{H}_{\mathsf{l}_j,\mathsf{r}_j,\mathsf{a},\mathsf{b}}(\mathsf{s}, \mathsf{t})$ represents a path

$$[\![h]\!](x, w) : \mathsf{s}(x) = \mathsf{t}(x)$$

Defining Bicategories of Algebras

- Type of algebras is an iterated $\sum$-type
- We want to construct the biadjunction on each component separately
- **Displayed bicategories** allow us to do that

Displayed Bicategories

Definition

Let B be a bicategory.
A **displayed bicategory** D over B consists of

- For each $x : \mathsf{B}$ a type $\mathsf{D}(x)$ of **objects over** x;
- For each $f : x \to y$, $\overline{x} : \mathsf{D}(x)$ and $\overline{y} : \mathsf{D}(y)$, a type $\overline{x} \xrightarrow{f} \overline{y}$ of **1-cells over** f;
- For each $\theta : f \Rightarrow g$, $\overline{f} : \overline{x} \xrightarrow{f} \overline{y}$, and $\overline{g} : \overline{x} \xrightarrow{g} \overline{y}$, a set $\overline{f} \overset{\theta}{\Rightarrow} \overline{g}$ of **2-cells over** θ.

Details: Ahrens, Frumin, Maggesi, Veltri, Van der Weide
For the categorical case: see Ahrens and Lumsdaine

Total Bicategory

Definition

Let D be a displayed bicategory on B.
Define the **total bicategory** $\int D$

- Objects are pairs $(x, \overline{x})$ with $x : \mathrm{B}$ and $\overline{x} : D(x)$
- 1-cells are pairs $(f, \overline{f})$ with $f : x \to y$ and $\overline{f} : \overline{x} \xrightarrow{f} \overline{y}$
- 2-cells are pairs $(\theta, \overline{\theta})$ with $\theta : f \Rightarrow g$ and $\overline{\theta} : \overline{f} \overset{\theta}{\Rightarrow} \overline{g}$

Note: there is a **projection** pseudofunctor $\pi_D : \int D \to B$.

Examples of Displayed Bicategory

Recall the HITs we consider

$\texttt{Inductive}\ H :=$
$|\ \ c : \mathsf{A}(H) \to H$
$|\ \ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$
$|\ \ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)), \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$
$|\ \ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

We define three displayed bicategories:

Examples of Displayed Bicategory

Recall the HITs we consider

`Inductive` $H :=$
$\mid\ c : \mathsf{A}(H) \to H$
$\mid\ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$
$\mid\ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)), \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$
$\mid\ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

We define three displayed bicategories:

- One that adds structure for the point constructor

Examples of Displayed Bicategory

Recall the HITs we consider

$\texttt{Inductive}\ H :=$
$|\ c : \mathsf{A}(H) \to H$
$|\ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$
$|\ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)), \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$
$|\ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

We define three displayed bicategories:

- One that adds structure for the point constructor
- One that adds structure for the path constructor

Examples of Displayed Bicategory

Recall the HITs we consider

```
Inductive H :=
```
$|\ c : \mathsf{A}(H) \to H$

$|\ p : \prod(j : \mathsf{J_P})(x : \mathsf{S}_j(H)), \mathsf{l}_j(x) = \mathsf{r}_j(x)$

$|\ s : \prod(j : \mathsf{J_H})(x : \mathsf{R}_j(H))(r : \mathsf{a}_j(x) = \mathsf{b}_j(x)), \mathsf{p}_j(x, r) = \mathsf{q}_j(x, r)$

$|\ t : \prod(x, y : H)(p, q : x = y)(r, s : p = q), r = s$

We define three displayed bicategories:

- One that adds structure for the point constructor
- One that adds structure for the path constructor
- One that adds structure for the 2-path constructor

Algebras on a Pseudofunctor

Note: a polynomial P gives a pseudofunctor

$$[\![P]\!] : \text{1-Type} \to \text{1-Type}.$$

Definition

Let B be a bicategory and let $F : \mathsf{B} \to \mathsf{B}$ be a pseudofunctor. Define a displayed bicategory $\mathsf{DFalg}(F)$ over B such that

- objects over $x : \mathsf{B}$ are 1-cells $h_x : F(x) \to x$;

Algebras on a Pseudofunctor

Note: a polynomial P gives a pseudofunctor

$$[\![P]\!] : \text{1-Type} \to \text{1-Type}.$$

Definition

Let B be a bicategory and let $F : \mathsf{B} \to \mathsf{B}$ be a pseudofunctor. Define a displayed bicategory DFalg(F) over B such that

- objects over $x : \mathsf{B}$ are 1-cells $h_x : F(x) \to x$;
- 1-cells over $f : x \to y$ from h_x to h_y are invertible 2-cells $\tau_f : h_x \cdot f \Rightarrow F(f) \cdot h_y$;
- 2-cells over $\theta : f \Rightarrow g$ from τ_f to τ_g are equalities

$$h_x \lhd \theta \bullet \tau_g = \tau_f \bullet F(\theta) \rhd h_y.$$

A Brief Intermezzo: endpoints

Note that we have

$$\int \mathsf{DFalg}(\llbracket A \rrbracket) \xrightarrow{\pi_{\mathsf{DFalg}(\llbracket A \rrbracket)}} \text{1-Type} \underset{\llbracket T \rrbracket}{\overset{\llbracket S \rrbracket}{\rightrightarrows}} \text{1-Type}$$

An endpoint $e : \mathsf{E}_A(S, T)$ gives rise to a pseudotransformation

$$\llbracket e \rrbracket : \llbracket S \rrbracket \circ \pi_{\mathsf{DFalg}(\llbracket A \rrbracket)} \Rightarrow \llbracket T \rrbracket \circ \pi_{\mathsf{DFalg}(\llbracket A \rrbracket)}.$$

Adding 2-cells to the structure

Definition

Let D be a displayed bicategory over B.
Suppose that we have pseudofunctors $S, T : \mathsf{B} \to \mathsf{B}$ and pseudotransformations $l, r : S \circ \pi_D \Rightarrow T \circ \pi_D$.
Define a displayed bicategory $\mathsf{DFcell}(l, r)$ over $\int D$ such that

- the objects over x are 2-cells $\gamma_x : l(x) \Rightarrow r(x)$;

Adding 2-cells to the structure

Definition

Let D be a displayed bicategory over B.
Suppose that we have pseudofunctors $S, T : \mathrm{B} \to \mathrm{B}$ and pseudotransformations $l, r : S \circ \pi_D \Rightarrow T \circ \pi_D$.
Define a displayed bicategory $\mathrm{DFcell}(l, r)$ over $\int D$ such that

- the objects over x are 2-cells $\gamma_x : l(x) \Rightarrow r(x)$;
- the 1-cells over $f : x \to y$ from γ_x to γ_y are equalities

$$(\gamma_x \triangleright T(\pi_D(f))) \bullet r(f) = l(f) \bullet (S(\pi_D(f)) \triangleleft \gamma_y);$$

- the 2-cells over $\theta : f \Rightarrow g$ are inhabitants of the unit type.

The full subbicategory

Definition

Let B be a bicategory and let P be a family of propositions on the objects of B. Define a displayed bicategory FSub(P) over B

- objects over x are proofs of $P(x)$
- the displayed 1-cells and 2-cells are inhabitants of the unit type

The total bicategory $\int$FSub(P) is the **full subbicategory** of B whose objects satisfy P.

Putting it together (1-types)

For a signature Σ, we get the following bicategories

- $\mathsf{PreAlg}(\Sigma)$ (via DFalg). Objects (**prealgebras**) consist of
  - a 1-type X
  - a function $\mathsf{c}^X : \mathsf{A}(X) \to X$

Putting it together (1-types)

For a signature Σ, we get the following bicategories

- $\mathsf{PreAlg}(\Sigma)$ (via DFalg). Objects (**prealgebras**) consist of
  - a 1-type X
  - a function $\mathsf{c}^X : \mathsf{A}(X) \to X$
- $\mathsf{PathAlg}(\Sigma)$ (via DFcell). Objects (**path algebras**) consist of
  - a 1-type X
  - a function $\mathsf{c}^X : \mathsf{A}(X) \to X$
  - for each $j : \mathsf{J}_\mathsf{P}$ and $x : \mathsf{S}_j(X)$ a path $\mathsf{p}_j^X(x) : [\![\mathsf{l}_j]\!](x) = [\![\mathsf{r}_j]\!](x)$;

Putting it together (1-types)

For a signature Σ, we get the following bicategories

- $\mathsf{PreAlg}(\Sigma)$ (via DFalg). Objects (**prealgebras**) consist of
  - a 1-type X
  - a function $\mathsf{c}^X : \mathsf{A}(X) \to X$
- $\mathsf{PathAlg}(\Sigma)$ (via DFcell). Objects (**path algebras**) consist of
  - a 1-type X
  - a function $\mathsf{c}^X : \mathsf{A}(X) \to X$
  - for each $j : \mathsf{J}_\mathsf{P}$ and $x : \mathsf{S}_j(X)$ a path $\mathsf{p}_j^X(x) : [\![\mathsf{l}_j]\!](x) = [\![\mathsf{r}_j]\!](x)$;
- $\mathsf{Alg}(\Sigma)$ (via FSub). Objects (**algebras**) consist of
  - a 1-type X
  - a function $\mathsf{c}^X : \mathsf{A}(X) \to X$
  - for each $j : \mathsf{J}_\mathsf{P}$ and $x : \mathsf{S}_j(X)$ a path $\mathsf{p}_j^X(x) : [\![\mathsf{l}_j]\!](x) = [\![\mathsf{r}_j]\!](x)$;
  - for each $j : \mathsf{J}_\mathsf{H}$, $x : \mathsf{R}(X)$ and $w : [\![a_l]\!](x) = [\![a_r]\!](x)$, a 2-path

$$\mathsf{h}_j^X : [\![\mathsf{p}]\!](x, w) = [\![\mathsf{q}]\!](x, w)$$

Putting it together (groupoids)

For a signature Σ, we get the following bicategories

- $\mathsf{PreAlg}_{\mathsf{Grpd}}(\Sigma)$ (via DFalg).
- $\mathsf{PathAlg}_{\mathsf{Grpd}}(\Sigma)$ (via DFcell).
- $\mathsf{Alg}_{\mathsf{Grpd}}(\Sigma)$ (via FSub).

The Groupoid Quotient

The groupoid quotient is the following HIT:

$\texttt{Inductive}$ GQuot (G : Grpd) :=
| **gcl** : $G \to \mathrm{GQuot}(G)$
| **gcleq** : $\prod(x, y : G)(f : G(x, y)), \mathbf{gcl}(x) = \mathbf{gcl}(y)$
| **ge** : $\prod(x : G), \mathbf{gcleq}(\mathrm{id}(x)) = \mathbf{idpath}(x)$
| **gconcat** : $\prod(x, y, z : G)(f : G(x, y))(g : G(y, z)),$
$\mathbf{gcleq}(f \cdot g) = \mathbf{gcleq}(f) \bullet \mathbf{gcleq}(g)$
| **gtrunc** : $\prod(x, y : \mathrm{GQuot}(G))(p, q : x = y)(r, s : p = q),$
$r = s$

The Groupoid Quotient is a Biadjunction

The groupoid quotient gives to a pseudofunctor

$$\mathsf{GQuot} : \mathsf{Grpd} \to \text{1-Type}$$

We also have a pseudofunctor

$$\mathsf{PathGrpd} : \text{1-Type} \to \mathsf{Grpd}$$

(takes fundamental groupoid)

Proposition

We have: $\mathsf{GQuot} \dashv \mathsf{PathGrpd}$.

How to lift the biadjunction?

Again we want a biadjunction between $\mathsf{Alg}(\Sigma)$ and $\mathsf{Alg}_{\mathsf{Grpd}}(\Sigma)$

- Disadvantage of direct approach: requires reconstruction
- Instead we construct the biadjunction in a layered fashion

To do so, we use **displayed biadjunctions**

Displayed Biadjunctions: the idea

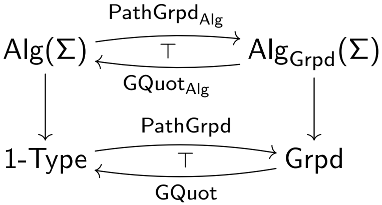
Situation:

- Displayed bicategories D_1 and D_2 over B_1 and B_2
- A biadjunction $L \dashv R$ with $L : B_1 \to B_2$

Displayed biadjunctions give a way to

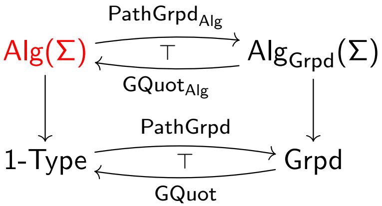
- construct biadjunctions between the totals $\int D_1$ and $\int D_2$
- on the first coordinate, it is given by $L \dashv R$
- the displayed biadjunction specifies the second coordinate

Concluding the construction



To construct a HIT on Σ, we do

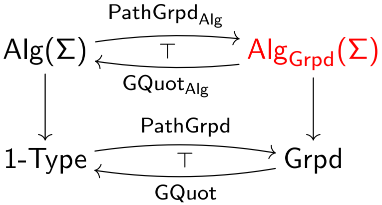
Concluding the construction



To construct a HIT on Σ, we do

- By biinitial algebra semantics, find biinitial object in $\mathsf{Alg}(\Sigma)$

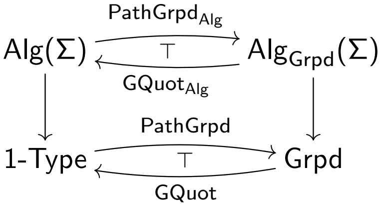
Concluding the construction



To construct a HIT on Σ, we do

- By biinitial algebra semantics, find biinitial object in $\mathsf{Alg}(\Sigma)$
- By biadjunction, find biinitial object in $\mathsf{Alg}_{\mathsf{Grpd}}(\Sigma)$

Concluding the construction



To construct a HIT on Σ, we do

- By biinitial algebra semantics, find biinitial object in Alg(Σ)
- By biadjunction, find biinitial object in Alg$_{\mathsf{Grpd}}$(Σ)
- Technical, see formalization and (Dybjer, Moeneclaey, 2018)

Future Work

- More permissive syntax (HIITs by Kaposi, Kovács, 2018)
- Generalize this construction to the non-truncated case
- Use the notion of HIT signature to do algebra in bicategories