1-Types vs Groupoids

Niels van der Weide   Dan Frumin   Herman Geuvers

Radboud University Nijmegen, The Netherlands

June 18, 2018

Motivation

A higher inductive type is given by constructors, equations, ...

```
Inductive ℤ :=
| Z : ℤ
| S : ℤ → ℤ
| P : ℤ → ℤ
```

Motivation

A higher inductive type is given by constructors, equations, ...

$\texttt{Inductive}\ \mathbb{Z} :=$
$|\ \mathsf{Z} : \mathbb{Z}$
$|\ \mathsf{S} : \mathbb{Z} \to \mathbb{Z}$
$|\ \mathsf{P} : \mathbb{Z} \to \mathbb{Z}$
$|\ \mathsf{PS} : \prod (x : \mathbb{Z}), \mathsf{S}(\mathsf{P}\ x) = x$
$|\ \mathsf{SP} : \prod (x : \mathbb{Z}), \mathsf{P}(\mathsf{S}\ x) = x$

Motivation

A higher inductive type is given by constructors, equations, ...

```
Inductive ℤ :=
| Z : ℤ
| S : ℤ → ℤ
| P : ℤ → ℤ
| PS : ∏(x : ℤ), S(P x) = x
| SP : ∏(x : ℤ), P(S x) = x
| coh : ∏(x : ℤ), PS(S x) = ap S (SP x)
```

This type will be discussed in Pinyo's talk.
More examples:

- Type Theory in Type Theory (Kaposi, Altenkirch)
- Partiality (Altenkirch, Danielsson, Kraus)
- Finite Sets in HoTT (Frumin, Geuvers, Gondelman, Van der Weide)

Motivation

Dybjer and Moenclaey give a semantics of HITs in the groupoid model of type theory.
A groupoid G consists of

- A set G (*objects*)
- For each x, y in G a set $G\ x\ y$ (*hom sets*)
- For each x in G an element $e\ x : G\ x\ x$ (*identity*)
- For each x, y in G an *inverse* map

$$(\cdot)^{-1} : G\ x\ y \to G\ y\ x$$

- For all x, y, z in G a *composition*

$$\cdot : G\ y\ z \times G\ x\ y \to G\ x\ z$$

such that $\cdot$ is associative, e is a neutral element for $\cdot$, and inv are inverses for $\cdot$.

Motivation

In addition, we have 1-types.

A *set* is a type for which equality is proof irrelevant. All inhabitants of $x = y$ are equal.

A *1-type* is a type for which every $x = y$ is a set.

Questions

1-types and groupoids are related.

- What's the precise relation between 1-types and groupoids?
- Can we use this to give a semantics of HITs in 1-types?

We look into the first item.

Path Groupoid

We can map 1-types to groupoids.
Given a 1-type A, define the *path groupoid $P\ A$ on A*:

- Objects are inhabitants of A
- The hom set is $x = y$ (this is a set, because A is a 1-type)
- Identity: equality is reflexive
- Inverse: equality is symmetric
- Composition: equality is transitive

The laws hold by path induction.

Groupoid Quotient

Given a groupoid G, we define the *groupoid quotient of G* as the following HIT:

```
Inductive gquot G :=
| gcl : G → gquot G
| gcleq : ∏(x, y : A), G x y → gcl x = gcl y
| ge : ∏(x : A), gcleq x (e x) = refl
| ginv : ∏(x, y : A), ∏_{g:G x y} gcleq y x (g^{-1}) = (gcleq x y g)^{-1}
| gconcat : ∏(x, y, z : A), ∏(g₁ : G x y), ∏(g₂ : G y z),
    gcleq x z (g₁ · g₂) = gcleq x y g₁ @ gcleq y z g₂
| gtrunc : ∏(x, y : gquot A G), ∏(p, q : x = y), ∏(r, s : p = q), r = s
```

Note the similarities to the Rezk completion (Ahrens, Kapulkin, Shulman).

Groupoid Quotient

Equality in the groupoid quotient of G is described by G.

Proposition

For every $x, y : A$ *the types* $\mathrm{gcl}\ x = \mathrm{gcl}\ y$ *and* $G\ x\ y$ *are equivalent.*

Proposition

For all 1-types A*, we have* $A \simeq \mathrm{gquot}\ A\,(P\ A)$.

Bicategories

A bicategory is like a category, but it also has arrows between arrows.

Proposition

- *We have a bicategory of groupoids with functors and natural transformation.*
- *We have a bicategory of 1-types with functions and equality*

See the talk by Ahrens and Maggesi.

Main Conjecture

Conjecture

We have an biadjoint biequivalence gquot $\dashv P$.

What does this amount to?

- We need to make adjoint equivalences $\text{gquot}(P\,A) \to A$ and $G \to P(\text{gquot}\,G)$
- and many, many, many coherencies

See the talk by Ahrens and Maggesi

What we did/will do

We formalized

- The propositions ($\text{gcl } x = \text{gcl } y$ and $G\ x\ y$ are equivalent, $A \simeq \text{gquot } A\,(P\ A)$)
- Some notions in bicategory theory (bicategories, lax functors, transformations, ...)

Still remaining:

- Formalize the notions of biadjoints and biequivalences
- Prove the conjecture

See `https://github.com/nmvdw/groupoids`