Some constructions on ω-groupoids

Thorsten Altenkirch [1] Nuo Li [1] Ondřej Rypáček [2]

[1]School of Computer Science
University of Nottingham, UK

[2]University of Oxford, UK

17/07/14

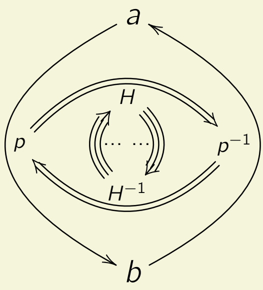
Outline

- Introduction to weak ω-groupoids
- Basic syntax of $\mathcal{T}_{\infty-\text{groupoid}}$ (describing the structure of weak ω-groupoids)
- Heterogeneous equality for syntactic terms
- Coherences constructions
- Semantic interpretation

Introduction to weak ω-groupoids I

What are weak ω-groupoids?

- A higher dimensional category (ω-category)
- Infinite levels of morphisms
- Generalization of setoids, groupoids
- Every morphism is an equivalence (generalization of isomorphism)
- Equalities are weak e.g. $(f \circ g) \circ h \to f \circ (g \circ h)$



Introduction to weak ω-groupoids II

Why are we interested in weak ω-groupoids?

- Interpretation of types in Homotopy Type Theory
- Isomorphic types are equal
- Abstract datatype, abstract reansoning
- Extensional concepts
- Weak ω-groupoid model

Introduction to weak ω-groupoids III

Formalizations of weak ω-groupoids in type theory

- Warren's *strict* ω-groupoid model
- Altenkirch and Rypacek's syntactic approach
- Brunerie's syntactic approach: $\mathcal{T}_{\infty-\text{groupoid}}$ (TIG)
- This paper:
  - implement $\mathcal{T}_{\infty-\text{groupoid}}$ in Agda
  - develop constructions

Agda

- Dependently typed programming languages, theorem prover
- An implementation of intensional Martin-Löf type theory

Basic syntax of $\mathcal{T}_{\infty-\text{groupoid}}$

- Fundamental elements (in Agda code)

```
data Con          : Set
data Ty (Γ : Con) : Set
data Tm           : {Γ : Con}(A : Ty Γ) → Set
```

- Types: basic objects, equality of objects, equality of equality...

$$\frac{}{\Gamma \vdash * \text{ type}} \qquad \frac{\Gamma \vdash a, b : A}{\Gamma \vdash a =_A b \text{ type}}$$

Operations and equalities I

- Operations and equality in
  - **Setoid**:

$$\begin{aligned} &\mathrm{id} : x = x \\ &\_^{-1} : x = y \to y = x \\ &\_ \circ \_ : y = z \to x = y \to x = z \end{aligned}$$

  - **Groupoid**:

$$\begin{aligned} &\lambda : id \circ p = p \\ &\rho : p \circ id = p \\ &\alpha : p \circ (q \circ r) = (p \circ q) \circ r \\ &\kappa : p^{-1} \circ p = id \\ &\kappa' : p \circ p^{-1} = id \end{aligned}$$

Operations and equalities II

- **weak ω-groupoid** : we have much more operations e.g. vertical/horizontal composition, and provable equalities on higher dimensions e.g. interchange law, coherence laws
  Example: There are two ways to show $(f \circ id) \circ g = f \circ g$

$$\begin{tikzcd}
(f \circ id) \circ g \arrow[r, "\alpha"] \arrow[dr, "\rho \cdot id"'] & f \circ (id \circ g) \arrow[d, "id \cdot \lambda"] \\
& f \circ g
\end{tikzcd}$$

- In general we call them **coherence constants** (or **coherences**)
- Infinitely many coherence constants, How can we encode them?

Contractible Contexts and Coherences I

- Fact: All coherences arising automatically from induction principle for identity type (or J eliminator)
- **contractible contexts**
  - $\epsilon, *$
  - $\epsilon, x : *, y : *, \alpha : x = y$
  - ... $\Gamma, y : A, \alpha : x = y$ (Given $\Gamma \vdash A$ and $\Gamma \vdash x : A$)
- Assume $\epsilon, x : * \vdash x = x$ (weakening)
  $\Rightarrow \epsilon, x : *, x : *, \alpha : x = x \vdash x = x$ (J-eliminator)
  $\Rightarrow \epsilon, x : *, y : *, \alpha : x = y \vdash y = x$
- How about coherences in non-cont contexts?
- In general

$$\frac{\vdash \Delta \text{ contractible} \quad \Delta \vdash B \quad \delta : \Gamma \to \Delta}{\Gamma \vdash \mathrm{coh}_B : B[\delta]}$$

Reasoning about syntactic terms

- Using homogeneous equality to reason about syntactic terms, we have to eliminate **subst** in equalities like subst $p\ x = y$, subst $p(\text{subst } p^{-1}\ x) = x$
- **Heterogeneous equality** (JM equality) for Tm

```
data _≅_ {Γ : Con}{A : Ty Γ}
      : {B : Ty Γ} → Tm A → Tm B → Set where
   refl : (b : Tm A) → b ≅ b
```

- Justification: The equality of inductively defined types are decidable, From Hedberg's Theorem, it is safe to assert Ty Γ are sets (in the sense of UIP)

Construction of Coherences I

- Now we can construct all these infinite number of coherences
- For each coherence, two versions:
  1. minimum version e.g.
     $\epsilon, x : *, y : *, \alpha : x = y, z : A, \beta : y = z \vdash x = z$ (_ $\circ^*$ _)
  2. general version e.g.
     $\Gamma, x : A, y : A, \alpha : x = y, z : A, \beta : y = z \vdash x = z$ (_ $\circ$ _)
- A minimum version is always in a contractible context and can be obtained by identity substitution
- General version is more complicated
- **Replacement**: to obtain the general version from minimum

$$\frac{\Gamma \vdash A \quad \vdash \Delta \text{ contractible} \quad \Delta \vdash \mathsf{coh}^*_B : B}{\Gamma, \Delta^A \vdash \mathsf{coh}^A_B : B^A}$$

Construction of Coherences II

- There is no $\Gamma, \Delta^A \Rightarrow \Delta$
- Solution: Filter out variables in Γ, Δ^A which are unnessary to build A
- Think reversely: build a "filtered" context using
- **Suspension**: Assume A is of level n. suspend Δ n times, i.e. add a *stalk* in front of Δ

$$\frac{\Gamma \vdash A \quad \vdash \Delta \text{ contractible} \quad \Delta \vdash B}{\Sigma_A\ \Delta \vdash \Sigma_A\ \mathsf{coh}_B : \Sigma_A\ B}$$

Construction of Coherences III

- and naturally we have a substitution called **filter**

$$\mathrm{filter}_A : \Gamma, \Delta^A \Rightarrow \Sigma_A\ \Delta$$

- Case: Assume $\Delta = (x : *)$, $B = (x = x)$ and in $\Gamma = (a : *, b : *, c : *)$, $A = (a = b)$ (level 1)
  $\Sigma_A\ \Delta = (x_0 : *, x_1 : *, x : x_0 = x_1)$
  $\Gamma, \Delta^A = (a : *, b : *, c : *, x : a = b)$
  $x_0 \mapsto a, x_1 \mapsto b, x \mapsto x$
- Finally because suspension preserves contractibility, $\Sigma_A\ \Delta$ is contractible
- In general, $\mathrm{coh}_B^A := (\Sigma_A\ (\mathrm{coh}_B))[\mathrm{filter}_A]$

Construction of Coherences IV

- Application: **Reflexivity**
  - 1st step: reflexivity (id) in a minimum contractible context

$$x : * \vdash \mathsf{coh}_{x=x} : x = x$$

  - 2nd step: reflexivity for arbitrary type A in arbitrary context Γ
    By suspension:

$$\Sigma_A\ (x : *) \vdash \Sigma_A\ (\mathsf{coh}_{x=x}) : x = x$$

    Replacement defined using filter

$$\mathsf{coh}^A_{x=x} := (\Sigma_A\ (\mathsf{coh}_{x=x}))[\mathsf{filter}_A]$$

Semantics

- A syntactic Grothendieck weak ω-groupoids is a globular set with an interpretation of syntactic coherence terms (coh)
- A globular set A consists coinductively of:
  - A set obj_A
  - For every $x, y : \mathrm{obj}_A$, a globular set $\mathrm{Hom}_A(x, y)$
- Example: the identity globular set $Id^\omega A$
  - $\mathrm{obj}_{Id^\omega A} = A$
  - $\mathrm{Hom}_{Id^\omega A}(a, b) = Id^\omega A(a = b)$
- The interpretation of contexts, types and terms

Conclusion

- Types bear the structure of weak ω-groupoids: the tower of iterated identity types
- An implementation of syntactic weak ω-groupoids in Agda
  - Basic syntax of the type theory $\mathcal{T}_{\infty-\text{groupoid}}$
  - Heterogeneous equality for terms
  - Constructions of coherences
  - Semantic interpretation with globular sets
- To complete a weak ω-groupoid model of type theory
- A computational interpretation of univalent axiom in Intensional Type Theory
- Question?