# Some constructions on $\omega$-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

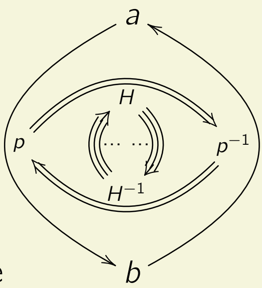
# Outline

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

# Introduction to weak $\omega$-groupoids I

**What are weak $\omega$-groupoids?**

- A higher dimensional category, infinite levels of morphisms
- Generalization of setoids, groupoids
- Every morphism is an equivalence $\cong$
- generalization of isomorphism: $f \circ g \cong 1_B$ and $g \circ f \cong 1_A$
- Why is it weak? equalities are equivalence e.g. $(f \circ g) \circ h \to f \circ (g \circ h)$



# Introduction to weak $\omega$-groupoids II

**Why are we interested in weak $\omega$-groupoids?**

- Main motivation : a weak $\omega$-groupoid model of Homotopy Type Theory
- Interpretation of types
- Univalence: isomorphic types are equal
- To reason about types abstractly
- Extensional concepts

# Introduction to weak $\omega$-groupoids III

**How to define weak $\omega$-groupoids in type theory?**

- Warren's *strict* $\omega$-groupoid model (not univalent)
- Altenkirch and Rypacek's syntactic approach
- Brunerie's syntactic approach $\mathcal{T}_{\infty-\text{groupoid}}$ (TIG): A type theory to describe the **internal** structure of weak $\omega$-groupoids
- In this paper we:
    - implement $\mathcal{T}_{\infty-\text{groupoid}}$ in Agda
    - develop some constructions

**Agda**

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

# Formalizing weak $\omega$-groupoids

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

- Fundamental elements:
  - Context: $\Gamma$ : Set
  - Type: $A$ : Ty $\Gamma$
  - Term: $t$ : Tm $A$
  - Substitution: $\delta : \Gamma \Rightarrow \Delta$ , $A[\delta]$, $t[\delta]$
- Context

$$\frac{}{\epsilon \text{ context}} \qquad \frac{\Gamma \text{ context} \quad \Gamma \vdash A \text{ type}}{\Gamma, A \text{ context}}$$

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

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

# Term I

- Terms are either variables or constants like operations or laws in weak $\omega$-groupoids
- Operations and equality (all weak in weak $\omega$-groupoids) 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**:

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

# Term II

- **weak $\omega$-groupoid** : we have many 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?
- Fact: All coherences arising automatically from induction principle for identity type (or J eliminator)

# Term III

- A simulation of J using contractible context
- **contractible contexts**
  - $x : *$ $\quad x : *, y : *, \alpha : x = y$ ...
  - $\Gamma, y : A, \alpha : x = y$ (Given $\Gamma \vdash A$ and $\Gamma \vdash x : A$)
- In $x : *$ we have id : $x = x$, $\mathrm{id}^2 : id = id$...
- The extending process helps us build context for other coherences: Example: Assume $\epsilon, x : * \vdash x = x$ (weakening) $\Rightarrow \epsilon, x : *, x : *, \alpha : x = x \vdash x = x$ (J-eliminator) $\Rightarrow \epsilon, x : *, y : *, \alpha : x = y \vdash y = x$
- We can prove that all constructions in contractible contexts are coherences
- In general

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

# Construction of Coherences I

- To prove we have all coherences by construction
- Our approach: for each kind of coherence e.g. composition
  1. minimum version
     $\epsilon, x : *, y : *, \alpha : x = y, z : *, \beta : y = z \vdash x = z$
  2. general version
     $\Gamma, x : A, y : A, \alpha : x = y, z : A, \beta : y = z \vdash x = z$
- **Replacement**: to obtain the general version from minimum

$$\frac{\Gamma \vdash A \quad \vdash \Delta \text{ contractible} \quad \Delta \vdash \mathrm{coh}^{*}_{B} : B}{\Gamma, \Delta^{A} \vdash \mathrm{coh}^{A}_{B} : B^{A}}$$

- Unfortunately there is no $\Gamma, \Delta^{A} \Rightarrow \Delta$ (to define $\mathrm{coh}^{A}_{B}$)

# Construction of Coherences II

- Solution: a contractible context where the expected coherence exists
- Ideally, filter out variables in $\Gamma, \Delta^A$ which are unnessary to build $A$
- Think inversely: why not build a "filtered" context?
- **Suspension**: Assume $A$ is of level $n$. suspend $\Delta$ $n$ times (lift $\Delta$ $n$ levels higher), i.e. build a *stalk* in front of $\Delta$

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

- A stalk is a context of $2n$ variables: $x : *, y : *, p : x = y, q : x = y \ldots$ which contains minimum ingredients for $A$ and also $\Delta^A$ built upon $A$
- $\Sigma_A\, \Delta$ can also be denoted as $\mathrm{stalk}_A, \Delta^A$

# Construction of Coherences III

- Important: The suspension preserves contractibility, suspension of $\Delta$ is also contractible
- With the simple substitution called **filter**

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

- 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, $\mathrm{coh}_B^A := (\Sigma_A\ (\mathrm{coh}_B))[\mathrm{filter}_A]$

# 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 $\Gamma$
    By suspension:

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

    Replacement defined using filter

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

# 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 $\Gamma$ are sets (in the sense of UIP)

# Semantics

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

# Conclusion

- Types bear the structure of weak $\omega$-groupoids: the tower of iterated identity types
- An implementation of syntactic weak $\omega$-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 $\omega$-groupoid model of type theory
- A computational interpretation of univalent axiom in Intensional Type Theory
- Question?