An Implementation of Syntactic Weak ω-Groupoids in Agda

Nuo Li

School of Computer Science
University of Nottingham, UK

13/06/13

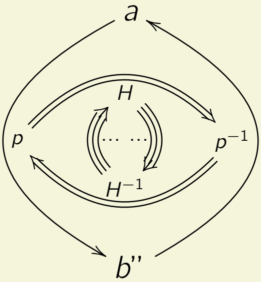
What are weak ω-groupoids?

- Generalisation of Setoids : Every morphism is an equivalence
- As higher dimensional categories: Sets Setoids Groupoids ω-groupoids

a

b

$$\begin{tikzcd} a \arrow[d, bend right, "p"'] \\ b \arrow[u, bend right, "p^{-1}"'] \end{tikzcd}$$



Why do we need weak ω-groupoids ?

- In Homotopy Type Theory, we reject the uniqueness of identity proof (UIP)
- The setoid interpretation of types has to be generalised to weak ω-groupoids
- It is weak, because the equalities are not definitionally equal
- To eliminate the univalence axiom

History of ω-groupoids

- Grothedieck introduced the notion of ω-groupoids in 1983 in a manuscript called *Pursuing Stacks*
- Maltsiniotis and Ara simplified the original definition
- In Type Theory:
  - Altenkirch and Rypacek's syntactic approach
  - Brunerie's syntactic definition

What we have done

- Following Bruneries's raw definition

Outline

- The syntax of the type theory $\mathcal{T}_{\infty-groupoid}$
- Some derivable constructions from the syntax
- The semantic interpretation of the syntax with globular sets

- Main contributions:
  1. Heterogeneous equality for terms
  2. Some construction derived from the syntax
  3. Suspensions

The basic Objects

$$
\begin{array}{ll}
\mathsf{data}\ \mathsf{Con} & : \mathsf{Set} \\
\mathsf{data}\ \mathsf{Ty}\ (\Gamma : \mathsf{Con}) & : \mathsf{Set} \\
\mathsf{data}\ \mathsf{Tm} & : \{\Gamma : \mathsf{Con}\}(A : \mathsf{Ty}\ \Gamma) \to \mathsf{Set} \\
\mathsf{data}\ \mathsf{Var} & : \{\Gamma : \mathsf{Con}\}(A : \mathsf{Ty}\ \Gamma) \to \mathsf{Set} \\
\mathsf{data}\ \_\Rightarrow\_ & : \mathsf{Con} \to \mathsf{Con} \to \mathsf{Set} \\
\\
\mathsf{data}\ \mathsf{isContr} & : \mathsf{Con} \to \mathsf{Set}
\end{array}
$$

The Basic Objects 2

```
data Con where
  ε   : Con
  _,_ : (Γ : Con)(A : Ty Γ) → Con

data Ty Γ where
  *    : Ty Γ
  _=h_ : {A : Ty Γ}(a b : Tm A) → Ty Γ
```

The Basic Objects 3

```
data Var where
  v0 : {Γ : Con}{A : Ty Γ}    → Var (A +T A)
  vS : {Γ : Con}{A B : Ty Γ}(x : Var A) → Var (A +T B)

data Tm where
  var : {Γ : Con}{A : Ty Γ} → Var A → Tm A
  JJ  : {Γ Δ : Con} → isContr Δ → (δ : Γ ⇒ Δ) → (A : Ty Δ)
        → Tm (A [ δ ]T)
```

The Basic Objects 4

```
data isContr where
  c*   : isContr (ε , *)
  ext : {Γ : Con}
        → isContr Γ → {A : Ty Γ}(x : Var A)
          → isContr ((Γ , A) , (var (vS x) =h var v0))
```

```
data _⇒_ where
  •    : {Γ : Con} → Γ ⇒ ε
  _,_ : {Γ Δ : Con}(δ : Γ ⇒ Δ){A : Ty Δ}(a : Tm (A [ δ ]T))
     → Γ ⇒ (Δ , A)
```

Heterogeneous equality for terms

- The first challenge: technically too many "Subst"s when reasoning about terms. unable to proceed
- We found that the equality of types is decidable : an h-set
- The proof of the equality is unique. We could use heterogeneous equality for Tm which depends on Ty

Heterogeneous equality for Tm

$$
\begin{array}{l}
\mathsf{data}\ \_\cong\_\ \{\Gamma : \mathsf{Con}\}\{A : \mathsf{Ty}\ \Gamma\} \\
\quad : \{B : \mathsf{Ty}\ \Gamma\} \to \mathsf{Tm}\ A \to \mathsf{Tm}\ B \to \mathsf{Set}\ \mathsf{where} \\
\quad \mathsf{refl} : (b : \mathsf{Tm}\ A) \to b \cong b
\end{array}
$$

Heterogeneous equality for terms 2

"Coercion"(proof-irrelevant substitution)

$$\_\llbracket\_\rangle\!\rangle : \{\Gamma : \mathsf{Con}\}\{A\ B : \mathsf{Ty}\ \Gamma\}(a : \ \mathsf{Tm}\ \ B) \rightarrow A \equiv B \rightarrow \mathsf{Tm}\ A$$
$$a \llbracket\ \mathsf{refl}\ \rangle\!\rangle = a$$

"Coherence Operator"

$$\mathsf{cohOp} : \{\Gamma : \mathsf{Con}\}\{A\ B : \mathsf{Ty}\ \Gamma\}\{a : \ \mathsf{Tm}\ \ B\}(p : \ A \equiv B)$$
$$\quad \rightarrow a \llbracket\ p\ \rangle\!\rangle \cong a$$
$$\mathsf{cohOp}\ \mathsf{refl} = \mathsf{refl}\ \_$$

Substitutions

$$
\begin{array}{lll}
\_[\_]T & : \{\Gamma\ \Delta : \mathsf{Con}\}(A : \mathsf{Ty}\ \Delta) & (\delta : \Gamma \Rightarrow \Delta) \rightarrow \mathsf{Ty}\ \Gamma \\
\_[\_]V & : \{\Gamma\ \Delta : \mathsf{Con}\}\{A : \mathsf{Ty}\ \Delta\}(a : \mathsf{Var}\ A)(\delta : \Gamma \Rightarrow \Delta) \rightarrow \mathsf{Tm}\ (A\ [\ \delta\ ]T) \\
\_[\_]tm & : \{\Gamma\ \Delta : \mathsf{Con}\}\{A : \mathsf{Ty}\ \Delta\}(a : \mathsf{Tm}\ A)\ (\delta : \Gamma \Rightarrow \Delta) \rightarrow \mathsf{Tm}\ (A\ [\ \delta\ ]T) \\
\_\circledcirc\_ & : \{\Gamma\ \Delta\ \Theta : \mathsf{Con}\} \quad \rightarrow \quad \Delta \Rightarrow \Theta \rightarrow \Gamma \Rightarrow \Delta \rightarrow \Gamma \Rightarrow \Theta
\end{array}
$$

The composition of context morphism can also be understood as the substitution for context morphism

Special Case Substitution

$$
\begin{array}{l}
\mathsf{var}\ x \quad [\ \delta\ ]tm = x\ [\ \delta\ ]V \\
\mathsf{JJ}\ c\Delta\ \gamma\ A\ [\ \delta\ ]tm = \mathsf{JJ}\ c\Delta\ (\gamma \circledcirc \delta)\ A\ [\![\ \mathsf{sym}\ [\circledcirc]T\ \rangle\!\rangle
\end{array}
$$

Weakening rules

Weakening rules

$$
\begin{aligned}
\_{+}T\_ &: \{\Gamma : \mathsf{Con}\}(A : \mathsf{Ty}\ \Gamma) \quad\ \to (B : \mathsf{Ty}\ \Gamma) \to \mathsf{Ty}\ (\Gamma\ ,\ B) \\
\_{+}tm\_ &: \{\Gamma : \mathsf{Con}\}\{A : \mathsf{Ty}\ \Gamma\}(a : \mathsf{Tm}\ A) \to (B : \mathsf{Ty}\ \Gamma) \to \mathsf{Tm}\ (A +T\ B) \\
\_{+}S\_ &: \{\Gamma : \mathsf{Con}\}\{\Delta : \mathsf{Con}\}(\delta : \Gamma \Rightarrow \Delta) \to (B : \mathsf{Ty}\ \Gamma) \to (\Gamma\ ,\ B) \Rightarrow \Delta
\end{aligned}
$$

Some important lemmas

The associativity lemmas for substitutions

$$
\begin{aligned}
&[\circledcirc]\mathsf{T} : \{\Gamma\ \Delta\ \Theta : \mathsf{Con}\}\\
&\quad \{\vartheta : \Delta \Rightarrow \Theta\}\{\delta : \Gamma \Rightarrow \Delta\}\{A : \mathsf{Ty}\ \Theta\}\\
&\quad \to A\ [\ \vartheta \circledcirc \delta\ ]\mathsf{T} \equiv (A\ [\ \vartheta\ ]\mathsf{T})[\ \delta\ ]\mathsf{T}
\end{aligned}
$$

$$
\begin{aligned}
&[\circledcirc]\mathsf{v} : \{\Gamma\ \Delta\ \Theta : \mathsf{Con}\}\\
&\qquad (\vartheta : \Delta \Rightarrow \Theta)(\delta : \Gamma \Rightarrow \Delta)(A : \mathsf{Ty}\ \Theta)(x : \mathsf{Var}\ A)\\
&\qquad \to x\ [\ \vartheta \circledcirc \delta\ ]\mathsf{V} \cong (x\ [\ \vartheta\ ]\mathsf{V})\ [\ \delta\ ]\mathsf{tm}
\end{aligned}
$$

$$
\begin{aligned}
&[\circledcirc]\mathsf{tm} : \{\Gamma\ \Delta\ \Theta : \mathsf{Con}\}\\
&\quad (\vartheta : \Delta \Rightarrow \Theta)(\delta : \Gamma \Rightarrow \Delta)(A : \mathsf{Ty}\ \Theta)(a : \mathsf{Tm}\ A)\\
&\quad \to\quad a\ [\ \vartheta \circledcirc \delta\ ]\mathsf{tm} \cong (a\ [\ \vartheta\ ]\mathsf{tm})\ [\ \delta\ ]\mathsf{tm}
\end{aligned}
$$

$$
\begin{aligned}
&\circledcirc\mathsf{assoc} : \{\Gamma\ \Delta\ \Theta\ \Delta_1 : \mathsf{Con}\}\\
&\quad (\gamma : \Theta \Rightarrow \Delta_1)(\vartheta : \Delta \Rightarrow \Theta)(\delta : \Gamma \Rightarrow \Delta)\\
&\quad \to (\gamma \circledcirc \vartheta) \circledcirc \delta \equiv \gamma \circledcirc (\vartheta \circledcirc \delta)
\end{aligned}
$$

Some important lemmas 2

Weakening inside substitution is equivalent to weakening outside.

$$
\begin{aligned}
&[+S]T : \{\Gamma\ \Delta : \mathsf{Con}\} \\
&\quad \{A : \mathsf{Ty}\ \Delta\}\{\delta : \Gamma \Rightarrow \Delta\} \\
&\quad \{B : \mathsf{Ty}\ \Gamma\} \\
&\quad \to A\ [\ \delta\ +S\ B\ ]T \equiv (A\ [\ \delta\ ]T)\ +T\ B
\end{aligned}
$$

$$
\begin{aligned}
&[+S]tm : \{\Gamma\ \Delta : \mathsf{Con}\}\{A : \mathsf{Ty}\ \Delta\} \\
&\qquad (a : \mathsf{Tm}\ A)\{\delta : \Gamma \Rightarrow \Delta\} \\
&\qquad \{B : \mathsf{Ty}\ \Gamma\} \\
&\qquad \to a\ [\ \delta\ +S\ B\ ]tm \cong (a\ [\ \delta\ ]tm)\ +tm\ B
\end{aligned}
$$

Some important lemmas 3

We could derive useful auxiliary functions from them. For example:

$$
\begin{array}{l}
\mathsf{wk\text{-}tm+} : \{\Gamma\ \Delta : \mathsf{Con}\} \\
\quad \{A : \mathsf{Ty}\ \Delta\}\{\delta : \Gamma \Rightarrow \Delta\} \\
\quad (B : \mathsf{Ty}\ \Gamma) \\
\quad \to \mathsf{Tm}\ (A\ [\ \delta\ ]\mathsf{T}\ +\mathsf{T}\ B) \to \mathsf{Tm}\ (A\ [\ \delta\ +\mathsf{S}\ B\ ]\mathsf{T}) \\
\mathsf{wk\text{-}tm+}\ B\ t = t\ [\![\ [+\mathsf{S}]\mathsf{T}\ \rangle\!\rangle
\end{array}
$$

Some important lemmas 4

Weakening doesn't introduce new variables in types and terms.

$$
\begin{aligned}
&+\mathsf{T}[,]\mathsf{T} : \{\Gamma\ \Delta : \mathsf{Con}\}\\
&\quad \{A : \mathsf{Ty}\ \Delta\}\{\delta : \Gamma \Rightarrow \Delta\}\\
&\quad \{B : \mathsf{Ty}\ \Delta\}\{b : \mathsf{Tm}\ (B\ [\ \delta\ ]\mathsf{T})\}\\
&\quad \rightarrow (A\ +\mathsf{T}\ B)\ [\ \delta\ ,\ b\ ]\mathsf{T} \equiv A\ [\ \delta\ ]\mathsf{T}
\end{aligned}
$$

$$
\begin{aligned}
&+\mathsf{tm}[,]\mathsf{tm} : \{\Gamma\ \Delta : \mathsf{Con}\}\{A : \mathsf{Ty}\ \Delta\}\\
&\quad (a : \mathsf{Tm}\ A)(\delta : \Gamma \Rightarrow \Delta)(B : \mathsf{Ty}\ \Delta)\\
&\quad (c : \mathsf{Tm}\ (B\ [\ \delta\ ]\mathsf{T}))\\
&\quad \rightarrow (a\ +\mathsf{tm}\ B)\ [\ \delta\ ,\ c\ ]\mathsf{tm} \cong a\ [\ \delta\ ]\mathsf{tm}
\end{aligned}
$$

Identity morphism

Identity morphism is not primitive notion in this setting.

$$\mathsf{IdCm} : \forall\ \Gamma \to \Gamma \Rightarrow \Gamma$$

Laws for Id

$$\mathsf{IC\text{-}T} \quad : \forall\ \{\Gamma : \mathsf{Con}\}(A : \mathsf{Ty}\ \Gamma) \to A\ [\ \mathsf{IdCm}\ \Gamma\ ]\mathsf{T} \equiv A$$
$$\mathsf{IC\text{-}v} : \forall\{\Gamma : \mathsf{Con}\}\{A : \mathsf{Ty}\ \Gamma\}(x : \mathsf{Var}\ A) \to \quad x\ [\ \mathsf{IdCm}\ \Gamma\ ]\mathsf{V} \cong \mathsf{var}\ x$$
$$\mathsf{IC\text{-}}\circledcirc : \forall\{\Gamma\ \Delta : \mathsf{Con}\}(\delta : \Gamma \Rightarrow \Delta) \to \delta \circledcirc \mathsf{IdCm}\ \Gamma \equiv \delta$$
$$\mathsf{IC\text{-}tm} : \forall\{\Gamma : \mathsf{Con}\}\{A : \mathsf{Ty}\ \Gamma\}(a : \mathsf{Tm}\ A) \to a\ [\ \mathsf{IdCm}\ \Gamma\ ]\mathsf{tm} \cong a$$

Some important properties

Any type in contractible context should be inhabited.

$$\mathsf{anyTypeInh} : \forall\{\Gamma\} \to \{A : \mathsf{Ty}\ \Gamma\} \to \mathsf{isContr}\ \Gamma \to \mathsf{Tm}\ \{\Gamma\}\ A$$
$$\mathsf{anyTypeInh}\ \{\mathsf{A} = A\}\ ctr = \mathsf{JJ}\ ctr\ (\mathsf{IdCm}\ \_)\quad A\ [\![\ \mathsf{sym}\ (\mathsf{IC\text{-}T}\ \_)\ \rangle\!\rangle$$

Some important properties 2

We use dependent product to define non-empty context.

```
Con* = Σ Con Ty

preCon : Con* → Con
preCon = proj₁

‖_‖ : Con* → Con
‖_‖ = uncurry _,_

lastTy : (Γ : Con*) → Ty (preCon Γ)
lastTy    = proj₂

lastTy' : (Γ : Con*) → Ty ‖ Γ ‖
lastTy' (_ ,, A) = A +T A
```

The construction of reflexivity

- To construct the reflexivity for any equality $x = x$ on any level.
- For any variable in some context, we could filter out all unrelated variables to get the minimum context to make the variable well-typed
- We need to define a notion called loop context for this special context

Loop Context

$\Omega\text{Con} : \mathbb{N} \to \text{Con*}$
$\Omega\text{Con}\ 0 = \varepsilon\ ,,\ *$
$\Omega\text{Con}\ (\text{suc}\ n) = \text{let}\ (\Gamma\ ,,\ A) = \Omega\text{Con}\ n\ \text{in}$
$\quad (\Gamma\ ,\ A\ ,\ A\ {+}\text{T}\ A)\ ,,\ (\text{var}\ (\text{vS}\ \text{v0})\ {=}\text{h}\ \text{var}\ \text{v0})$

The construction of reflexivity 2

To generate the loop context for any given non-empty context

```
loopΩ' : (Γ : Con)(A : Ty Γ)
      → Σ[ Ω : Con ] Σ[ ω : Ty Ω ] Σ[ γ : Γ ⇒ Ω ]
  Σ[ prf : ω [ γ ]T ≡ A ] isContr (Ω , ω)
loopΩ' Γ * = ε   ,, * ,, • ,, refl ,, c*
loopΩ' Γ (_=h_ {A} a b) with loopΩ' Γ A
... | (Γ' ,, A' ,, γ' ,, prf' ,, isc) =
     Γ' , A' , A' +T A' ,,
     (var (vS v0) =h var v0) ,,
     γ' , (a ⟦ prf' ⟫) , wk-tm (b ⟦ prf' ⟫) ,,
     (trans wk-hom (trans wk-hom (cohOp-hom prf'))) ,,
     ext isc v0

loopΩ : Con* → Con*
loopΩ (Γ ,, A) with (loopΩ' Γ A)
... |   (Γ' ,, A' ,, γ' ,, prf' ,, isc) = Γ' ,, A'
```

The construction of reflexivity 3

Finally we could generate the reflexivity terms.
Despite the loop context generator, the substitution and the coherence proof are both difficult problems.

Special Case Reflexivity

```
Tm-refl :   (ne : Con*) → Tm {|| ne ||} (var v0 =h var v0)
Tm-refl (Γ ,, A) with loopΩ' Γ A
... | ΩΓ ,, ΩA ,, γ ,, prf ,, isc =
    JJ isc (γ +S A , wk-tm+ A (var v0 ⟦ wk-T prf ⟫)) (var v0 =h var v0)
    ⟦ sym (trans wk-hom (trans wk-hom+ (hom≡ (cohOp (wk-T prf))
    (cohOp (wk-T prf))))) ⟫
```

The construction of reflexivity 4

General Case Reflexivity

Tm-refl' : $(\Gamma : \text{Con})(A : \text{Ty}\ \Gamma)(x : \text{Tm}\ A) \rightarrow \text{Tm}\ (x =\text{h}\ x)$
Tm-refl' Γ A x =
  (Tm-refl (Γ ,, A) [(IdCm _) , (x ⟦ IC-T A ⟫)]tm)
  ⟦ sym (trans wk-hom (hom≡ (cohOp (IC-T A)) (cohOp (IC-T A)))) ⟫

The construction of symmetry

A special case of symmetry.

```
Tm-sym : (Γ : Con)(A : Ty Γ)
       → Tm {Γ , A , A +T A} (var (vS v0) =h var v0)
       → Tm {Γ , A , A +T A} (var v0 =h var (vS v0))
Tm-sym Γ A t = (t [ (((IdCm _ +S A) +S (A +T A)) ,
  (var v0 ⟦ trans [+S]T (wk-T (trans [+S]T (wk-T (IC-T _)))) ⟫))
  (var (vS v0) ⟦ trans +T[,]T
    (trans [+S]T (wk-T (trans [+S]T (wk-T (IC-T _))))) ⟫) ]tm)
  ⟦ sym (trans wk-hom (hom≡ (htrans (cohOp +T[,]T)
  (cohOp (trans [+S]T (wk-T (trans [+S]T (wk-T (IC-T A)))))))
  (cohOp (trans +T[,]T (trans [+S]T (wk-T
    (trans [+S]T (wk-T (IC-T A))))))))) ⟫
```

To construct the general case, it could be easier to use other approaches.

Semantics: globular sets

To interpret the syntax, we need a globular set.
Globular sets are defined coinductively.

Globular Sets

```
record Glob : Set₁ where
  constructor _||_
  field
    |_|   : Set
    homo : |_| → |_| → ∞ Glob
open Glob public
```

The 0-level object should be of type h-set.

Semantics: interpretations

Given a globular set G, we could interpret the objects in syntactic frameworks.

$$
\begin{aligned}
&[\![\_]\!]C : \mathsf{Con} \to \mathsf{Set} \\
&[\![\_]\!]cm : \forall\{\Gamma\ \Delta : \mathsf{Con}\} \to (\Gamma \Rightarrow \Delta) \to [\![\ \Gamma\ ]\!]C \to [\![\ \Delta\ ]\!]C \\
&[\![\_]\!]T : \forall\{\Gamma\}(A : \mathsf{Ty}\ \Gamma)(\gamma : [\![\ \Gamma\ ]\!]C) \to \mathsf{Glob} \\
&[\![\_]\!]tm : \forall\{\Gamma\ A\}(v : \mathsf{Tm}\ A)(\gamma : [\![\ \Gamma\ ]\!]C) \to |\ [\![\ A\ ]\!]T\ \gamma\ |
\end{aligned}
$$

We also need a function called Coh. It returns an object for any valid coherence.

$$\mathsf{Coh} : (\Theta : \mathsf{Con})(ic : \mathsf{isContr}\ \Theta)(A : \mathsf{Ty}\ \Theta) \to (\vartheta : [\![\ \Theta\ ]\!]C) \to |\ [\![\ A\ ]\!]T\ \vartheta\ |$$

Semantics: some lemmas

We also need to prove some lemmas for semantics.

Semantic Weakening

$\text{semWK-ty} : \forall \{\Gamma : \text{Con}\}(A\ B : \text{Ty}\ \Gamma)(\gamma : [\![\ \Gamma\ ]\!]\text{C})(v : |\ [\![\ B\ ]\!]\text{T}\ \gamma\ |)$
$\quad \to [\![\ A\ ]\!]\text{T}\ \gamma \equiv [\![\ A\ \text{+T}\ B\ ]\!]\text{T}\ (\gamma\ ,\ v)$

$\text{semWK-tm} : \forall \{\Gamma : \text{Con}\}(A\ B : \text{Ty}\ \Gamma)(\gamma : [\![\ \Gamma\ ]\!]\text{C})(v : |\ [\![\ B\ ]\!]\text{T}\ \gamma\ |)$
$\quad (a : \text{Tm}\ A) \to \text{subst}\ |\_|\ (\text{semWK-ty}\ A\ B\ \gamma\ v)\ ([\![\ a\ ]\!]\text{tm}\ \gamma)$
$\qquad \equiv [\![\ a\ \text{+tm}\ B\ ]\!]\text{tm}\ (\gamma\ ,\ v)$

$\text{semWK-cm} : \forall \{\Gamma\ \Delta : \text{Con}\}(B : \text{Ty}\ \Gamma)(\gamma : [\![\ \Gamma\ ]\!]\text{C})(v : |\ [\![\ B\ ]\!]\text{T}\ \gamma\ |)$
$\quad (\delta : \Gamma \Rightarrow \Delta) \to [\![\ \delta\ ]\!]\text{cm}\ \gamma \equiv [\![\ \delta\ \text{+S}\ B\ ]\!]\text{cm}\ (\gamma\ ,\ v)$

Summary and Future work

- We have presented an implementation of weak ω-groupoidsin Agda.
- We had made some contribution to the Brunerie's raw definition.
- There are a lot of possible work to do. For example, compare it with a type theory with equality types and J eliminator
- The final goal should be to model the type theory with weak ω-groupoidsand to eliminate the univalence axiom

To prove every equivalence relation is a weak ω-groupoid

- We have reflexivity, symmetry and transtivity.
- on the second level, we have 5 groupoid laws.
- There are much more on higher levels.
- However we have UIP, all higher level coherence laws hold trivially.