A TYPE-THEORETICAL DEFINITION OF WEAK ω-CATEGORIES

Samuel Mimram

École Polytechnique

Logic In Computer Science

June 21, 2017

Higher categories

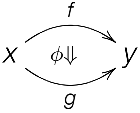
The definition of (strict) ω**-category** generalizes categories by taking higher cells into account.

Higher categories

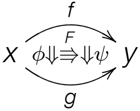
The definition of (strict) ω-**category** generalizes categories by taking higher cells into account.

In such a category, you have

- 0-cells (objects): x
- 1-cells (morphisms): $x \xrightarrow{f} y$
- 2-cells:



- 3-cells:

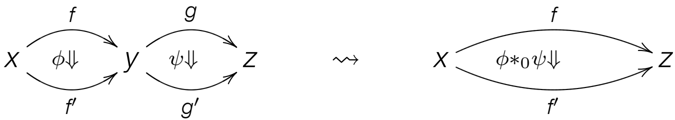


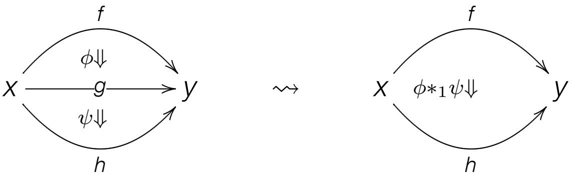
- …

Higher categories

The definition of (strict) ω-**category** generalizes categories by taking higher cells into account.

In such a category, you have **compositions**





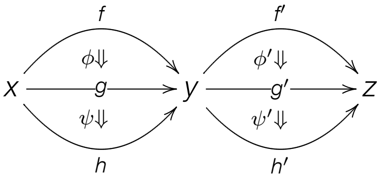
More generally, n-cells ϕ and ψ can be composed in dimension i, with $0 \leq i < n$, when their type match.

Higher categories

The definition of (strict) ω-**category** generalizes categories by taking higher cells into account.

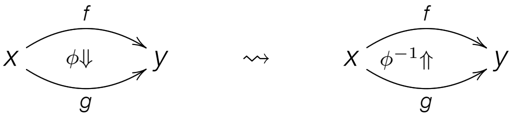
In such a category, you have **axioms** such as

- associativity of composition and neutrality of identities,
- exchange laws:



The definition of (strict) ω**-category** generalizes categories by taking higher cells into account.

In the case where the orientation of arrows is not really relevant, you can consider (strict) ω**-groupoids** which are ω-categories in which all n-cells are invertible.



Weak ω-groupoids

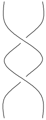
It turns out that this definition is too strict.

Given a topological space X, one expects to be able to build an ω-groupoid whose

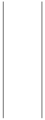
- 0-cells are the points of X,
- 1-cells are the paths in X,
  (we do have concatenation, constant paths, and inverses)
- 2-cells are homotopies,
- 3-cells are homotopies between homotopies,
- etc.

However,

- concatenation is only associative up to homotopy
- exchange is not strict



$\neq$



Partial history of weak ω-categories

- **1983**: a definition of weak ω-groupoids
  Grothendieck, *Pursuing Stacks*
- **2007**: a definition weak ω-categories (after Grothendieck)
  Maltsiniotis, *Infini catégories non strictes, une nouvelle définition*
- **2009**: homotopy types are weak ω-groupoids
  Lumsdaine, *Weak ω-categories from intensional type theory*
  van Den Berg, Garner, *Types are weak ω-groupoids*
- **2016**: a type-theoretic definition of weak ω-groupoids
  Brunerie, *On the homotopy groups of spheres in homotopy type theory*

Type-theoretic weak ω-categories

Here, we fill the following gap:

	groupoids	categories
category theory	Grothendieck	Maltsiniotis
type theory	Brunerie	**Finster-Mimram**

Why is this useful

- We have a **simple definition**
  (no advanced categorical concepts, a few inference rules)
- We have a **syntax**
  (we can reason by induction, etc.)
- We have **tools**
  (we can have the machine check our terms)
- A step toward **directed homotopy type theory**?
  (we are still far from handling variance, univalence, etc.)

A TYPE-THEORETIC DEFINITION OF CATEGORIES

Judgments in type-theory

- Γ is a well-formed context:

$$\Gamma \vdash$$

- A is a well-formed type in context Γ:

$$\Gamma \vdash A$$

- t is a term of type A in context Γ:

$$\Gamma \vdash t : A$$

- t and u are equal terms of type A in context Γ:

$$\Gamma \vdash t = u : A$$

A type-theoretic definition of categories

Cartmell, 1984:

- type constructors:

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash x : \star \qquad \Gamma \vdash y : \star}{\Gamma \vdash x \to y}$$

A type-theoretic definition of categories

Cartmell, 1984:

- type constructors:

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash x : \star \qquad \Gamma \vdash y : \star}{\Gamma \vdash x \to y}$$

- term constructors:

$$\frac{}{x : \star \vdash \mathrm{id}(x) : x \to x}$$

$$\frac{}{x : \star, y : \star, f : x \to y, z : \star, g : y \to z \vdash \mathrm{comp}(f, g) : x \to z}$$

A type-theoretic definition of categories

Cartmell, 1984:

- type constructors:

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash x : \star \qquad \Gamma \vdash y : \star}{\Gamma \vdash x \to y}$$

- term constructors:

$$\frac{}{x : \star \vdash \mathrm{id}(x) : x \to x}$$

$$\frac{}{x : \star, y : \star, f : x \to y, z : \star, g : y \to z \vdash \mathrm{comp}(f, g) : x \to z}$$

- axioms:

$$\frac{\Gamma \vdash f : x \to y}{\Gamma \vdash \mathrm{comp}(\mathrm{id}(x), f) = f} \qquad \frac{\Gamma \vdash f : x \to y}{\Gamma \vdash \mathrm{comp}(f, \mathrm{id}(y)) = f} \qquad \dots$$

A type-theoretic definition of categories

Cartmell, 1984:

- type constructors:

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash x : \star \qquad \Gamma \vdash y : \star}{\Gamma \vdash x \to y}$$

- term constructors:

$$\frac{}{x : \star \vdash \mathrm{id}(x) : x \to x}$$

$$\frac{}{x : \star, y : \star, f : x \to y, z : \star, g : y \to z \vdash \mathrm{comp}(f, g) : x \to z}$$

- axioms:

$$\frac{\Gamma \vdash f : x \to y}{\Gamma \vdash \mathrm{comp}(\mathrm{id}(x), f) = f} \qquad \frac{\Gamma \vdash f : x \to y}{\Gamma \vdash \mathrm{comp}(f, \mathrm{id}(y)) = f} \qquad \dots$$

- plus “standard rules” (contexts, weakening, substitutions, …)

Models of the type theory

A **model** of the type theory consists in interpreting

- closed types as sets,
- closed terms as elements of their type,

in such a way that axioms are satisfied.

Models of the type theory

A **model** of the type theory consists in interpreting

- closed types as sets,
- closed terms as elements of their type,

in such a way that axioms are satisfied.

A model of the previous type theory consists of

- a set $[\![\star]\!]$
- for each $x, y \in [\![\star]\!]$, a set $[\![\to]\!]_{x,y}$
- for each $x \in [\![\star]\!]$, an element $[\![\mathrm{id}]\!]_x \in [\![\to]\!]_{x,x}$
- …

Models of the type theory

A **model** of the type theory consists in interpreting

- closed types as sets,
- closed terms as elements of their type,

in such a way that axioms are satisfied.

A model of the previous type theory consists of

- a set $[\![\star]\!]$
- for each $x, y \in [\![\star]\!]$, a set $[\![\to]\!]_{x,y}$
- for each $x \in [\![\star]\!]$, an element $[\![\mathrm{id}]\!]_x \in [\![\to]\!]_{x,x}$
- …

In other words, a model of the type theory is precisely a **category** (and a morphism is a functor).

Going higher

We could gradually implement weak n-categories:

- bicategories
- tricategories
- tetracategories
- pentacategories
- ...

The problem is that

- the number of axioms is exploding
- nobody knows the definition excepting in low dimensions
- we would like to have a “uniform” definition

Unbiased definition

Since the composition is associative for categories, the composite of any diagram like

$$x_0 \xrightarrow{f_1} x_1 \xrightarrow{f_2} \dots \xrightarrow{f_n} x_n$$

is uniquely defined.

So, instead of having a binary composition and identities, we could have a more general rule

$$\frac{}{x_0 : \star, x_1 : \star, f_1 : x_0 \to x_1, \dots, x_n : \star, f_n : x_{n-1} \to x_n \vdash \mathrm{comp}(f_1, \dots, f_n) : x_0 \to x_n}$$

Unbiased definition

We can axiomatize categories with n-ary composition.

- This is very redundant, for instance

$$\mathrm{comp}(\mathrm{comp}(f,g),h) = \mathrm{comp}(f,g,h) = \mathrm{comp}(f,\mathrm{comp}(g,h))$$

or even

$$\mathrm{comp}(f) \quad = \quad f$$

Unbiased definition

We can axiomatize categories with n-ary composition.

- This is very redundant, for instance

  $$\mathrm{comp}(\mathrm{comp}(f,g),h) = \mathrm{comp}(f,g,h) = \mathrm{comp}(f,\mathrm{comp}(g,h))$$

  or even

  $$\mathrm{comp}(f) \quad = \quad f$$

- We have to characterize what we want to compose exactly. For instance, should be able to compose

  $$x_0 \xrightarrow{f_1} x_1 \xrightarrow{f_2} \ldots \xrightarrow{f_n} x_n$$

  but not

  $$x \underset{g}{\overset{f}{\rightleftarrows}} y \qquad z \qquad \text{or} \qquad x \xrightarrow{f} y \xleftarrow{g} z$$

Unbiased definition

We can axiomatize categories with *n*-ary composition.

- This is very redundant, for instance

  $$\mathrm{comp}(\mathrm{comp}(f,g),h) = \mathrm{comp}(f,g,h) = \mathrm{comp}(f,\mathrm{comp}(g,h))$$

  or even

  $$\mathrm{comp}(f) \quad = \quad f$$

- We have to characterize what we want to compose exactly. For instance, should be able to compose

  $$x_0 \xrightarrow{f_1} x_1 \xrightarrow{f_2} \ldots \xrightarrow{f_n} x_n$$

  but not

  $$x \underset{g}{\overset{f}{\rightleftarrows}} y \qquad z \qquad\qquad \text{or} \qquad\qquad x \xrightarrow{f} y \xleftarrow{g} z$$

- However, this generalizes nicely in higher dimensions!

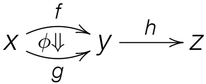
A TYPE-THEORETIC DEFINITION OF GLOBULAR SETS

Definition

A **globular set** consists of

- a set G, and
- for every $x, y \in G$, a globular set G^x_y.

Example



corresponds to

$$G = \{x, y, z\} \qquad G^x_y = \{f, g\} \qquad (G^x_y)^f_g = \{\phi\} \qquad ((G^x_y)^f_g)^\phi_\phi = \emptyset \qquad \ldots$$

Globular sets

Definition

A **globular set** consists of

- a set G, and
- for every $x, y \in G$, a globular set G^x_y.

Alternatively, this can be defined as

- a sequence of sets G_n of n-cells for $n \in \mathbb{N}$,
- with source and target maps

$$s_n, t_n : G_{n+1} \to G_n$$

satisfying suitable axioms.

Globular sets

Proposition

Globular sets are precisely the models of the type theory

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : A}{\Gamma \vdash t \underset{A}{\to} u} \qquad \ldots$$

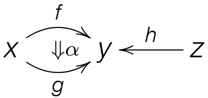
Proposition

Globular sets are precisely the models of the type theory

$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad\qquad \frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : A}{\Gamma \vdash t \underset{A}{\to} u} \qquad\qquad \dots$$

Remark

A finite globular set



can be encoded as a context

$$x : \star, y : \star, z : \star, f : x \underset{\star}{\to} y, g : x \underset{\star}{\to} y, h : z \underset{\star}{\to} y, \alpha : f \underset{x \underset{\star}{\to} y}{\to} g$$

Globular sets

Proposition

Globular sets are precisely the models of the type theory

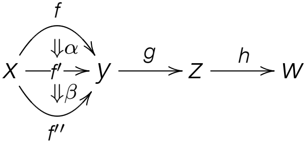
$$\frac{\Gamma \vdash}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash t : A \qquad \Gamma \vdash u : A}{\Gamma \vdash t \underset{A}{\to} u} \qquad \ldots$$

Proposition

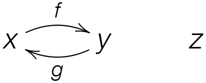
The syntactic category (of contexts and substitutions) of this type theory is the opposite of the category of finite globular sets.

PASTING SCHEMES

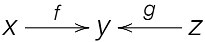
We now want to define **pasting schemes** which are diagrams for which we expect to have a composition. For instance,



is a pasting scheme, but not



or



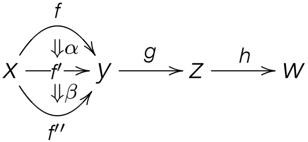
Given $n \in \mathbb{N}$, the ***n*-disk** D_n is the globular set corresponding to a general n-cell:

D_0 D_1 D_2 D_3

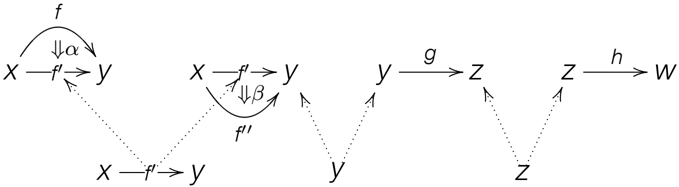
(these are the representable globular sets)

Pasting schemes

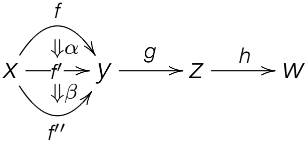
A **pasting scheme** is a globular set



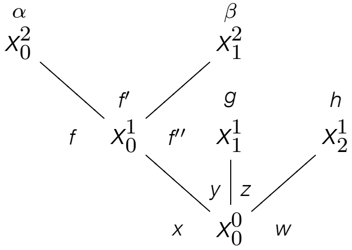
- *Grothendieck*: which can be obtained as a particular colimit of disks



A **pasting scheme** is a globular set

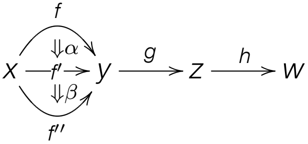


- *Batanin*: which is described by a particular tree



Pasting schemes

A **pasting scheme** is a globular set



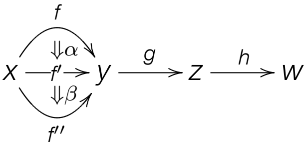
- *Finster-Mimram*: which is “totally ordered”

Order relation

We can define a preorder $\triangleleft$ on the cells of a globular set by

$$\mathrm{source}(x) \triangleleft x \qquad \text{and} \qquad x \triangleleft \mathrm{target}(x)$$

For the globular set



we have

$$x \triangleleft f \triangleleft \alpha \triangleleft f' \triangleleft \beta \triangleleft f'' \triangleleft y \triangleleft g \triangleleft z \triangleleft h \triangleleft w$$

Characterization of pasting schemes

Theorem

*A globular set is a **pasting scheme** if and only if it is*

- *non-empty,*
- *finite, and*
- *the relation $\triangleleft$ is a total order.*

Construction of pasting schemes

A *pointed globular set* is a globular set with a distinguished cell.

Construction of pasting schemes

A *pointed globular set* is a globular set with a distinguished cell.

Theorem

A ***pasting scheme*** *is a pointed globular set which can be constructed as follows:*

Construction of pasting schemes

A *pointed globular set* is a globular set with a distinguished cell.

Theorem

*A **pasting scheme** is a pointed globular set which can be constructed as follows:*

- *we start from a 0-cell* x

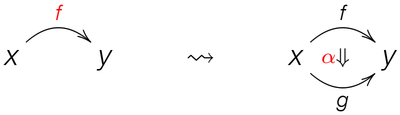
Construction of pasting schemes

A *pointed globular set* is a globular set with a distinguished cell.

Theorem

A ***pasting scheme*** *is a pointed globular set which can be constructed as follows:*

- *we start from a* 0*-cell* x
- *we can add a new* $(n+1)$*-cell and its new target, its source being the distinguished* n*-cell*



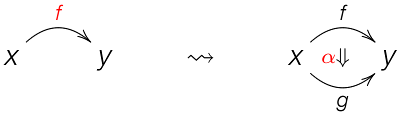
Construction of pasting schemes

A *pointed globular set* is a globular set with a distinguished cell.

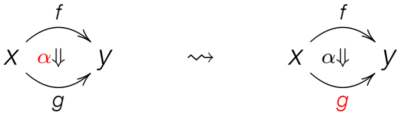
Theorem

A ***pasting scheme*** *is a pointed globular set which can be constructed as follows:*

- *we start from a* 0*-cell* x
- *we can add a new* $(n+1)$*-cell and its new target, its source being the distinguished n-cell*



- *or the distinguished cell becomes the target of the previous one*



Construction of pasting schemes

The construction of the pasting scheme

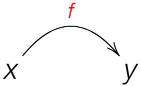
X

corresponds to its order

X

Construction of pasting schemes

The construction of the pasting scheme

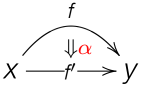


corresponds to its order

$x \quad \triangleleft \quad f$

Construction of pasting schemes

The construction of the pasting scheme

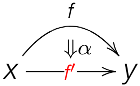


corresponds to its order

$$x \quad \triangleleft \quad f \quad \triangleleft \quad \alpha$$

Construction of pasting schemes

The construction of the pasting scheme

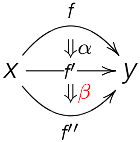


corresponds to its order

$$x \quad \lhd \quad f \quad \lhd \quad \alpha \quad \lhd \quad f'$$

Construction of pasting schemes

The construction of the pasting scheme

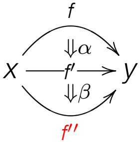


corresponds to its order

$$x \quad \lhd \quad f \quad \lhd \quad \alpha \quad \lhd \quad f' \quad \lhd \quad \beta$$

Construction of pasting schemes

The construction of the pasting scheme

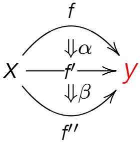


corresponds to its order

$$x \quad \triangleleft \quad f \quad \triangleleft \quad \alpha \quad \triangleleft \quad f' \quad \triangleleft \quad \beta \quad \triangleleft \quad f''$$

Construction of pasting schemes

The construction of the pasting scheme

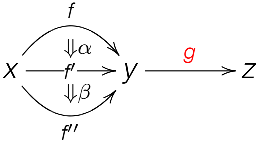


corresponds to its order

$$x \quad \triangleleft \quad f \quad \triangleleft \quad \alpha \quad \triangleleft \quad f' \quad \triangleleft \quad \beta \quad \triangleleft \quad f'' \quad \triangleleft \quad y$$

Construction of pasting schemes

The construction of the pasting scheme

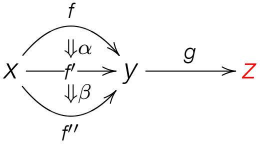


corresponds to its order

$$x \lhd f \lhd \alpha \lhd f' \lhd \beta \lhd f'' \lhd y \lhd g$$

Construction of pasting schemes

The construction of the pasting scheme

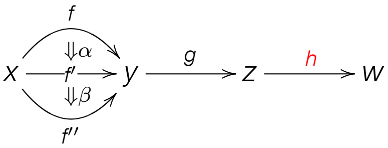


corresponds to its order

$$x \quad \triangleleft \quad f \quad \triangleleft \quad \alpha \quad \triangleleft \quad f' \quad \triangleleft \quad \beta \quad \triangleleft \quad f'' \quad \triangleleft \quad y \quad \triangleleft \quad g \quad \triangleleft \quad z$$

Construction of pasting schemes

The construction of the pasting scheme

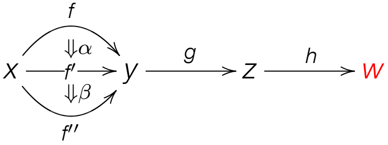


corresponds to its order

$$x \triangleleft f \triangleleft \alpha \triangleleft f' \triangleleft \beta \triangleleft f'' \triangleleft y \triangleleft g \triangleleft z \triangleleft h$$

Construction of pasting schemes

The construction of the pasting scheme

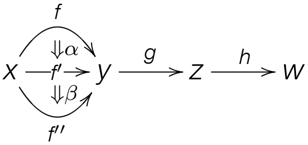


corresponds to its order

$$x \triangleleft f \triangleleft \alpha \triangleleft f' \triangleleft \beta \triangleleft f'' \triangleleft y \triangleleft g \triangleleft z \triangleleft h \triangleleft w$$

Type-theoretic pasting schemes

Now, recall that a pasting scheme



can be seen as a context

$$x : \star, y : \star, f : x \to y, f' : x \to y,$$
$$\alpha : f \to f', f'' : x \to y, \beta : f' \to f'',$$
$$z : \star, g : y \to z, w : \star, h : z \to w$$

Type-theoretic pasting schemes

A context Γ (seen as a globular set) is a **pasting scheme** iff

$$\Gamma \vdash_{\mathsf{ps}}$$

is derivable with the rules

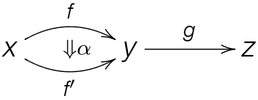
$$\frac{}{x : \star \vdash_{\mathsf{ps}} x : \star} \qquad \frac{\Gamma \vdash_{\mathsf{ps}} x : \star}{\Gamma \vdash_{\mathsf{ps}}}$$

$$\frac{\Gamma \vdash_{\mathsf{ps}} x : A}{\Gamma, y : A, f : x \underset{A}{\to} y \vdash_{\mathsf{ps}} f : x \underset{A}{\to} y} \qquad \frac{\Gamma \vdash_{\mathsf{ps}} f : x \underset{A}{\to} y}{\Gamma \vdash_{\mathsf{ps}} y : A}$$

Type-theoretic pasting schemes

Note that with those rules

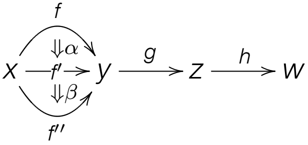
- the order of cells matters:



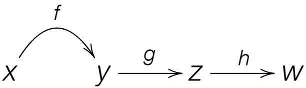
- because of this we can check

Source and targets

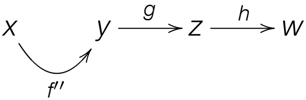
A pasting scheme Γ has



- a **source** $\partial^-(\Gamma)$:



- a **target** $\partial^+(\Gamma)$:



both of which can be defined by induction on contexts.

A TYPE-THEORETIC DEFINITION OF ω-CATEGORIES

Type-theoretic ω-groupoids

We expect that in an ω-category every pasting scheme has a composite:

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \Gamma \vdash A}{\Gamma \vdash \mathsf{coh}_{\Gamma,A} : A}$$

Type-theoretic ω-groupoids

We expect that in an ω-category every pasting scheme has a composite:

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \Gamma \vdash A}{\Gamma \vdash \mathsf{coh}_{\Gamma,A} : A}$$

You can derive expected operations, such as composition:

$$x : \star, y : \star, f : x \underset{\star}{\rightarrow} y, z : \star, g : y \underset{\star}{\rightarrow} z \vdash \mathsf{coh} : x \underset{\star}{\rightarrow} z$$

Type-theoretic ω-groupoids

We expect that in an ω-category every pasting scheme has a composite:

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \Gamma \vdash A}{\Gamma \vdash \mathsf{coh}_{\Gamma,A} : A}$$

You can derive expected operations, such as composition:

$$x : \star, y : \star, f : x \underset{\star}{\to} y, z : \star, g : y \underset{\star}{\to} z \vdash \mathsf{coh} : x \underset{\star}{\to} z$$

However, you can derive too much:

$$x : \star, y : \star, f : x \underset{\star}{\to} y \vdash \mathsf{coh} : y \underset{\star}{\to} x$$

We have in fact a definition of ω-**groupoids** (close to Brunerie's).

Type-theoretic ω-groupoids

We need to take care of side-conditions and in fact split the rule in two:

- operations:

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \Gamma \vdash t \underset{A}{\to} u \qquad \partial^-(\Gamma) \vdash t : A \qquad \partial^+(\Gamma) \vdash u : A}{\Gamma \vdash \mathsf{coh}_{\Gamma, t \underset{A}{\to} u} : t \underset{A}{\to} u}$$

  whenever

$$FV(t) = FV(\partial^-(\Gamma)) \qquad \text{and} \qquad FV(u) = FV(\partial^+(\Gamma))$$

- coherences:

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \Gamma \vdash A}{\Gamma \vdash \mathsf{coh}_{\Gamma, A} : A}$$

  whenever

$$FV(A) = FV(\Gamma)$$

Type-theoretic ω-groupoids

Definition

An ω**-category** is a model of this type theory.

Type-theoretic ω-groupoids

Definition

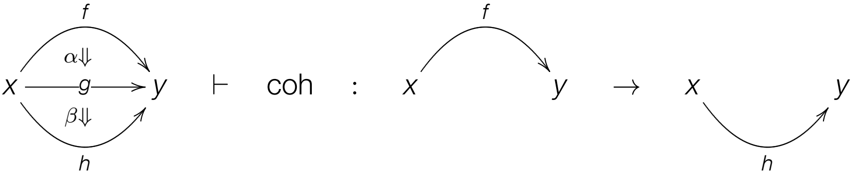
An ω**-category** is a model of this type theory.

Conjecture

This definition coincides with Grothendieck-Maltsiniotis'.

Type-theoretic ω-groupoids

A typical example of **operation** is *composition*



(this coherence is noted “comp” in the following).

Type-theoretic ω-groupoids

A typical example of **coherence** is *associativity*

$$x \xrightarrow{f} y \xrightarrow{g} z \xrightarrow{h} w$$

$$\vdash$$

$$\text{coh} \quad : \quad x \xrightarrow{\text{comp}(\text{comp}(f,g),h)} w \quad \to \quad x \xrightarrow{\text{comp}(f,\text{comp}(g,h))} w$$

Coherences are reversible

Note that if we derive a coherence

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \Gamma \vdash A}{\Gamma \vdash \mathsf{coh}_{\Gamma,A} : A} \qquad \text{with} \qquad FV(A) = FV(\Gamma)$$

where

$$A \quad = \quad t \to u \; ,$$

there is also one with

$$A \quad = \quad u \to t \; .$$

Coherences are reversible

Note that if we derive a coherence

$$\frac{\Gamma \vdash_{\mathsf{ps}} \qquad \Gamma \vdash A}{\Gamma \vdash \mathsf{coh}_{\Gamma,A} : A} \qquad \text{with} \qquad FV(A) = FV(\Gamma)$$

where

$$A \quad = \quad t \to u \, ,$$

there is also one with

$$A \quad = \quad u \to t \, .$$

Definition

An n-cell $f : x \to y$ is **reversible** when there exists

- an n-cell $g : y \to x$ and
- reversible $(n+1)$-cells

$$\alpha : f *_{n-1} g \to \mathsf{id}_x \qquad\qquad \beta : g *_{n-1} f \to \mathsf{id}_y$$

Implementation(s)

There are currently two implementations:

- `https://github.com/ericfinster/catt`
  - follows closely the rules of the article
- `https://github.com/smimram/catt`
  - has support for implicit arguments
  - has support for (some) Π-types
  - has support for “Hom” type variables:

```
let comp (X : Hom) =
  coh (x : X) (y : X) (f : x -> y)  (z : X) (g : y -> z)
        : (x -> z)
```

  - has a web interface

In practice,

- you simply enter a list of coherences (there is no reduction, etc.),
- if the program does not complain then they are valid operations in weak ω-categories.

“Demo”

- identity 1-cells

```
coh id (x : *) : * | x -> x ;
```

“Demo”

- identity 1-cells

```
coh id (x : *) : * | x -> x ;
```

- composition of 1-cells:

```
coh comp (x : *) (y : *) (f : * | x -> y)
         (z : *) (g : * | y -> z)
         : * | x -> z ;
```

"Demo"

- identity 1-cells

```
coh id (x : *) : * | x -> x ;
```

- composition of 1-cells:

```
coh comp (x : *) (y : *) (f : * | x -> y)
         (z : *) (g : * | y -> z)
         : * | x -> z ;
```

- associativity of composition of 1-cells:

```
coh assoc
    (x : *) (y : *) (f : * | x -> y) (z : *)
    (g : * | y -> z) (w : *) (h : * | z -> w)
    : * | x -> w
        | comp x z (comp x y f z g) w h ->
          comp x y f w (comp y z g w h) ;
```

“Demo”

- identity 1-cells

```
coh id (x : *) : * | x -> x ;
```

- composition of 1-cells:

```
coh comp (x : *) (y : *) (f : * | x -> y)
         (z : *) (g : * | y -> z)
         : * | x -> z ;
```

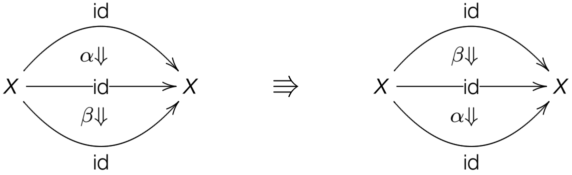
- associativity of composition of 1-cells:

```
coh assoc
    (x : *) (y : *) (f : * | x -> y) (z : *)
    (g : * | y -> z) (w : *) (h : * | z -> w)
    : * | x -> w
        | comp x z (comp x y f z g) w h ->
          comp x y f w (comp y z g w h) ;
```

- …

"Demo"

Only defining the Eckmann-Hilton morphism takes 300 lines



because you have to

- define usual operations and coherences,
- explicitly insert and remove identities,
- take care of bracketing of composites

```
let eh (X : Hom) (x : X) (a : id x -> id x) (b : id x -> id x)
     : (comp' a b -> comp' b a) =
    comp11 (comp' (unitl'- a) (unitr'- b)) (assoc3 _ _ _ _)
    (compl2r' _ _ (unitlr x) _) (compl2' _ _ (comp3 (assoc- _ _
    (compl' _ (assoc- _ _ _)) (complr' _ (ich b a) _)
    (complr' _ (compr' (comp (unitr- _) (compl' _ (unitr+-- _)))
    (comp (complr' _ (assoc3 _ _ _ _) _) (compl' _ (assoc4 _
```

“Demo”

- no inverses:

```
coh inv (x : *) (y : *) (f : * | x -> y)
        : * | y -> x ;
```

  produces

```
Checking coherence: inv
Valid tree context
Src/Tgt check forced
Source context: (x : *)
Target context: (y : *)
Failure: Source is not algebraic for y : *
```

CONCLUSION

Current work

Many things remain to be done:

- understand more exotic features
  (implicit arguments, reduction, etc.)
- links with Globular
- add functors and higher morphisms (Thibaut Benjamin)
- variant to define opetopic categories