Type theoretic definitions of structured weak higher categories

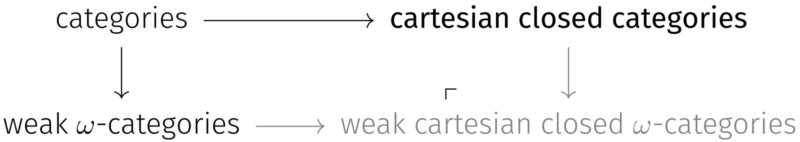
Samuel Mimram

École polytechnique

MathStic Category Theory Workshop
June 16, 2025

Goals

The goal is to define type theories whose models are (weak higher) (structured) categories



This is based on joint work with Éric Finster and Thibaut Benjamin:

- *A type-theoretical definition of weak ω-categories*, LICS 2017.
- *Globular weak ω-categories as models of a type theory*, Higher Struct. 2024.

Based on earlier work by Ara, Batanin, Gothendieck, Leinster, Maltsiniotis, …

Main ideas

What I want to convey here is that in order to define weak higher structures

- it is often easier to be unbiased / generic / non parcimonious
- it is enough to formally make generic composition situations contractible
- this can be done using type theory

Part I

Categories

Categories

A **category** is a graph equipped with composition and identities such that

$$h \circ (g \circ f) = (h \circ g) \circ f \qquad \mathrm{id} \circ f = f = f \circ \mathrm{id}$$

Why is this a nice definition?

Categories

A **category** is a graph equipped with composition and identities such that

$$h \circ (g \circ f) = (h \circ g) \circ f \qquad \mathrm{id} \circ f = f = f \circ \mathrm{id}$$

Why is this a nice definition?

We have a well-defined notion of composition for composable morphisms!

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

e.g.

$$i \circ (h \circ (g \circ f)) \qquad \text{or} \qquad ((\mathrm{id} \circ i) \circ \mathrm{id}) \circ (h \circ (g \circ ((\mathrm{id} \circ \mathrm{id}) \circ f)))$$

Categories

In some sense, what we really want to implement is an **unbiased** notion of category where we have a unique composite

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

for every $n \in \mathbb{N}$ but

- the binary compositions and identities are enough to generate all of them,
- the associativity and unitality axioms ensure uniqueness of composite.

Categories

We only want compositions for composable situations such as

$$X_0 \xrightarrow{f_1} X_1 \xrightarrow{f_2} \cdots \xrightarrow{f_n} X_n$$

Categories

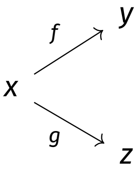
We only want compositions for composable situations such as

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

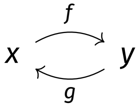
but not

$$x \xrightarrow{f} y \xleftarrow{g} z$$

or



or



Categories

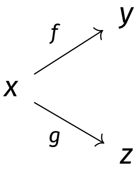
We only want compositions for composable situations such as

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

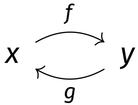
but not

$$x \xrightarrow{f} y \xleftarrow{g} z$$

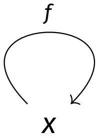
or



or



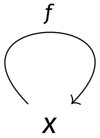
nor



which could mean f or $f \circ f$ or $f \circ f \circ f$

Categories

In a situation such as



if we want to compute

$$f \circ f \circ f$$

we can consider the composite of

$$x_0 \xrightarrow{f_1} x_1 \xrightarrow{f_2} x_2 \xrightarrow{f_3} x_3$$

and then instantiate to $x_i = x$ and $f_i = f$.

Judgments in type-theory

- $\Gamma \equiv x_1 : A_1, \ldots, x_n : A_n$ 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:

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

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 $[\![\mathrm{id}]\!]_x \in [\![\to]\!]_{x,x}$
- $\dots$

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

Unbiased definition

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

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

is uniquely defined.

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

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

and associated axioms.

The models of this **unbiased** definition would still be categories.

Part II

A type theory for globular sets

Higher categories

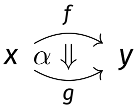
The definition of ω-**category** generalizes categories by taking higher cells into account.

Higher categories

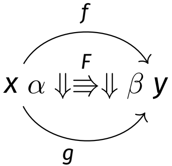
The definition of ω-**category** generalizes categories by taking higher cells into account.

In such a category, you have

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



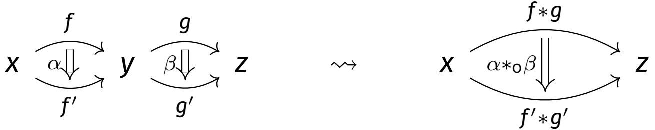
- 3-cells:

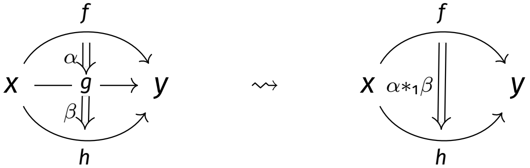


Higher categories

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

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





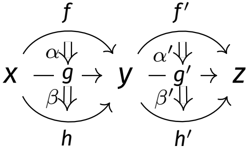
More generally, *n*-cells α and β can be composed in dimension *i*, with $0 \leq i < n$.

Higher categories

The definition of 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:



(more on this later)

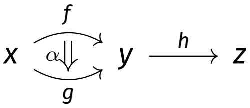
Globular sets

Definition

A **globular set** consists of

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

For instance



corresponds to

$$G = \{x, y, z\} \qquad G_y^x = \{f, g\} \qquad (G_y^x)_g^f = \{\alpha\} \qquad ((G_y^x)_g^f)_\alpha^\alpha = \emptyset \qquad \ldots$$

Globular sets

Definition
A **globular set** consists of

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

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.

$$G_0 \underset{t_0}{\overset{s_0}{\leftleftarrows}} G_1 \underset{t_1}{\overset{s_1}{\leftleftarrows}} G_2 \underset{t_2}{\overset{s_2}{\leftleftarrows}} \cdots$$

Globular sets

Proposition

Globular sets are precisely the models of the type theory

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

Globular sets

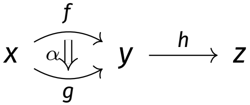
Proposition

Globular sets are precisely the models of the type theory

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

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

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

Proposition

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

Part III

Weak higher categories

Weak higher categories

A strict higher category is a globular set with compositions and satisfying axioms: associativity, unitality and exchange.

Weak higher categories

A strict higher category is a globular set with compositions and satisfying axioms: associativity, unitality and exchange.

In a **weak** higher category, all the axioms should hold up to a higher cell, which should be unique up to higher cells.

Those can be thought of as an *intensional* variant of higher categories.

Bicategories

The notion of **bicategory** is defined almost as for 2-categories, excepting that we replace the requirement that composition of 1-cells is associative and unital by

- **weak associativity**: given

$$x \xrightarrow{a} y \xrightarrow{b} z \xrightarrow{c} w$$

  there is an invertible 2-cell, the **associator**,

$$\alpha_{a,b,c} : (a *_0 b) *_0 c \Rightarrow a *_0 (b *_0 c)$$

Bicategories

The notion of **bicategory** is defined almost as for 2-categories, excepting that we replace the requirement that composition of 1-cells is associative and unital by

- **weak associativity**: given

  $$x \xrightarrow{a} y \xrightarrow{b} z \xrightarrow{c} w$$

  there is an invertible 2-cell, the **associator**,

  $$\alpha_{a,b,c} : (a *_0 b) *_0 c \Rightarrow a *_0 (b *_0 c)$$

- **weak unitality**: given

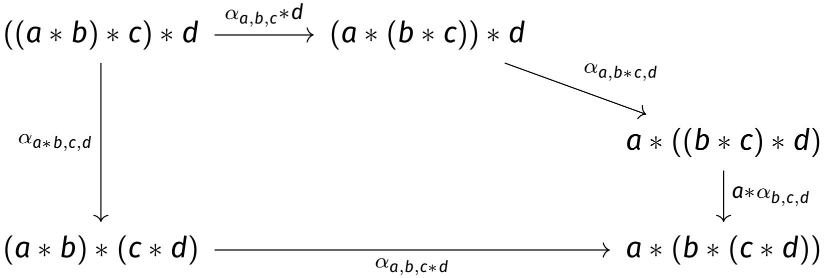
  $$x \xrightarrow{a} y$$

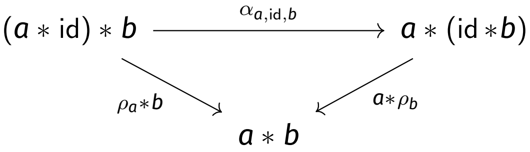
  there are invertible 2-cells, the **left** and **right unitors**,

  $$\lambda_a : \mathrm{id}_x *_0 a \Rightarrow a \qquad \rho_a : a *_0 \mathrm{id}_y \Rightarrow a$$

Bicategories: axioms

We also need to ensure that those satisfy suitable axioms,
the **pentagon** and the **triangle**:





This notion is pleasant because

Theorem (Mac Lane's coherence theorem)
Any two ways of composing 1*-cells are isomorphic and there is one such structural isomorphism.*

For instance,

$$f_1 * (f_2 * (f_3 * f_4)) \cong (f_1 * f_2 * f_3) * (\mathrm{id} * f_4)$$

Tricategories

Defining **tricategories** can be done starting from the definition of 3-categories and

1. replacing all equalities between 0-, 1- and 2- cells by 1-, 2- and 3- cells,
2. making those coherent by adding the suitable axioms.

Tricategories

Defining **tricategories** can be done starting from the definition of 3-categories and

1. replacing all equalities between 0-, 1- and 2- cells by 1-, 2- and 3- cells,
2. making those coherent by adding the suitable axioms.

For instance, we replace associativity of composition between 1-cells

$$(f *_0 g) *_0 h = f *_0 (g *_0 h)$$

by an invertible **associator** 2-cell

$$\alpha_{f,g,h} : (f *_0 g) *_0 h \Rightarrow f *_0 (g *_0 h)$$

Tricategories

Defining **tricategories** can be done starting from the definition of 3-categories and

1. replacing all equalities between 0-, 1- and 2- cells by 1-, 2- and 3- cells,
2. making those coherent by adding the suitable axioms.

For instance, we replace associativity of composition between 1-cells

$$(f *_0 g) *_0 h = f *_0 (g *_0 h)$$

by an invertible **associator** 2-cell

$$\alpha_{f,g,h} : (f *_0 g) *_0 h \Rightarrow f *_0 (g *_0 h)$$

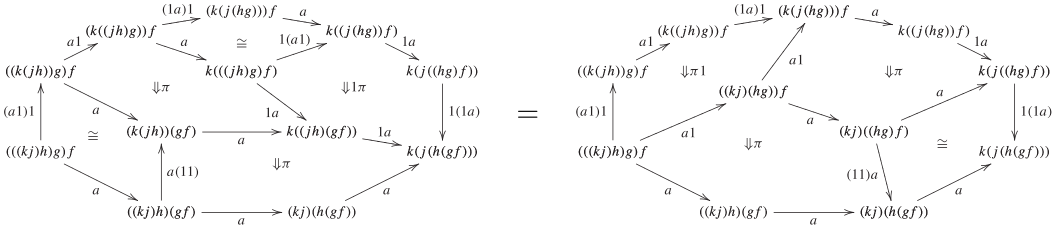
but by "invertible", we mean here that $\alpha_{f,g,h}$ should be an equivalence:

$$\eta : \mathrm{Id} \Rrightarrow \alpha_{f,g,h} *_1 \overline{\alpha}_{f,g,h} \qquad\qquad \varepsilon : \overline{\alpha}_{f,g,h} *_1 \alpha_{f,g,h} \Rrightarrow \mathrm{Id}$$

and so on...

Tricategories

The definition of tricategories takes roughly 4 pages with axioms such as



Tetracategories

The process can be generalized to define **weak *n*-categories**.

Tetracategories

The process can be generalized to define weak *n*-categories.

No one has ever tried to give a definition of a pentacategory in this way.

Tetracategories

The process can be generalized to define **weak n-categories**.

No one has ever tried to give a definition of a **pentacategory** in this way.

Instead of explicitly trying to give the axioms of weak higher categories, one should try to find a systematic way of generating those.

Tetracategories

The process can be generalized to define **weak *n*-categories**.

No one has ever tried to give a definition of a **pentacategory** in this way.

Instead of explicitly trying to give the axioms of weak higher categories, one should try to find a systematic way of generating those.

If we go all the way, we obtain **weak ω-categories** aka **(∞, ω)-categories**.

In those, we never have axioms, only higher cells. This can be thought of as very constructive definition: we want to have witnesses for all the laws.

The general scheme

Instead of trying to carefully craft compositions and coherences, it is actually easier to take an **unbiased** approach.

The general pattern is that

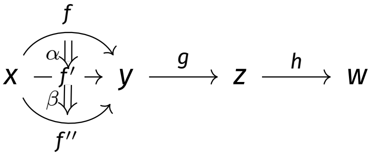
- we identify situation that should be contractible (the *pasting schemes*)
- and formally make them contractible

Part IV

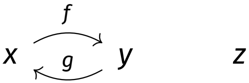
Pasting schemes

Pasting schemes

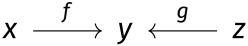
We now want to define **pasting schemes** which are diagrams for which we expect to have a composition. For instance,



is a pasting scheme, but not



or



Disks

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

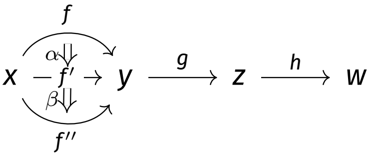
D_0 $\qquad$ D_1 $\qquad$ D_2 $\qquad$ D_3

Those are basic building blocks of globular sets: any globular set can be obtained by gluing such disks.

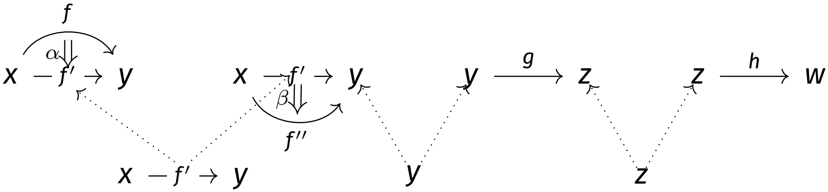
(those are the representable globular sets)

Pasting schemes

A **pasting scheme** is a globular set

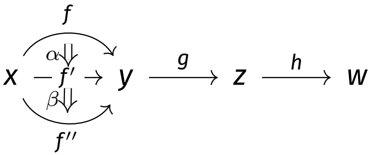


- *Grothendieck*: which can be obtained as a particular colimit of disks

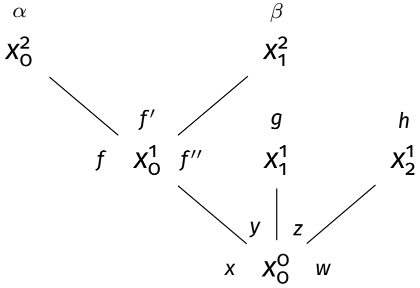


Pasting schemes

A **pasting scheme** is a globular set

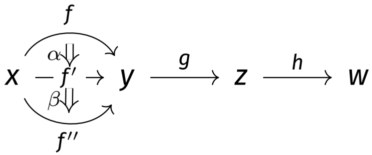


- *Batanin*: which is described by a particular tree



Pasting schemes

A **pasting scheme** is a globular set



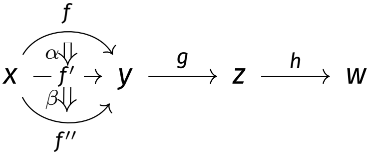
- *Finster-Mimram*: which is “totally ordered”

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:*

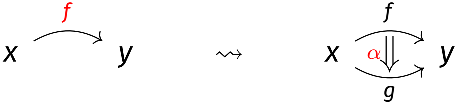
- *we start from a* 0*-cell* x

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*



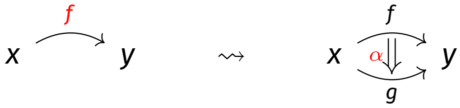
Construction of pasting schemes

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

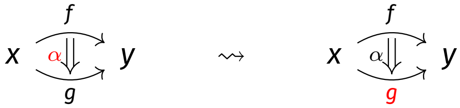
Theorem

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

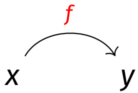
x

corresponds to its order

x

Construction of pasting schemes

The construction of the pasting scheme

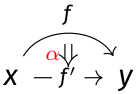


corresponds to its order

$x \quad \triangleleft \quad f$

Construction of pasting schemes

The construction of the pasting scheme

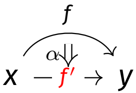


corresponds to its order

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

Construction of pasting schemes

The construction of the pasting scheme

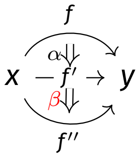


corresponds to its order

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

Construction of pasting schemes

The construction of the pasting scheme

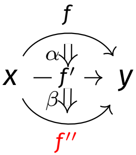


corresponds to its order

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

Construction of pasting schemes

The construction of the pasting scheme

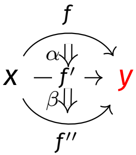


corresponds to its order

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

Construction of pasting schemes

The construction of the pasting scheme

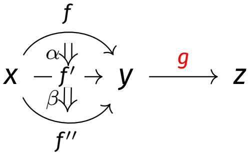


corresponds to its order

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

Construction of pasting schemes

The construction of the pasting scheme

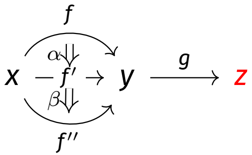


corresponds to its order

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

Construction of pasting schemes

The construction of the pasting scheme

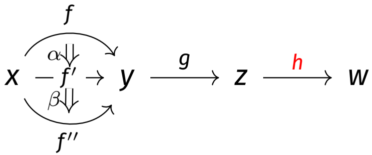


corresponds to its order

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

Construction of pasting schemes

The construction of the pasting scheme

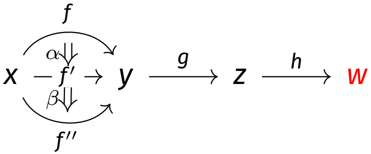


corresponds to its order

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

Construction of pasting schemes

The construction of the pasting scheme

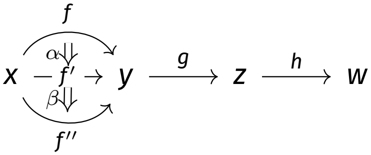


corresponds to its order

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

Type-theoretic pasting schemes

Now, recall that a pasting scheme



can be seen as a context

$$
\begin{gathered}
x : \star, y : \star, f : x \to y, f' : x \to y, \\
\alpha : f \to f', f'' : x \to y, \beta : f' \to f'', \\
z : \star, g : y \to z, w : \star, h : z \to w
\end{gathered}
$$

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

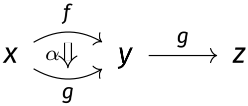
$$\frac{}{x : \star \vdash_{\mathsf{ps}} x : \star} \qquad \frac{\Gamma \vdash_{\mathsf{ps}} x : \star}{\Gamma \vdash_{\mathsf{ps}}}$$

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

Type-theoretic pasting schemes

Note that with those rules

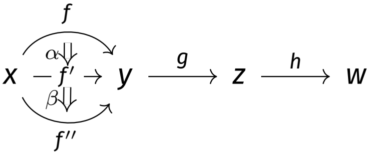
- the order of cells matters:



- because of this we can easily check
- proofs are canonical

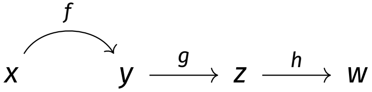
Source and targets

A pasting scheme Γ

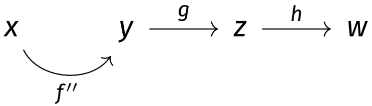


has

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



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



both of which can be defined by induction on contexts.

Part V

A type-theoretic definition of weak ω-categories

Type-theoretic ω-categories

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-categories

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:

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

Type-theoretic ω-categories

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**

Type-theoretic ω-categories

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

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

- coherences:

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

  whenever

$$\mathsf{FV}(A) = \mathsf{FV}(\Gamma)$$

Type-theoretic ω-categories

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

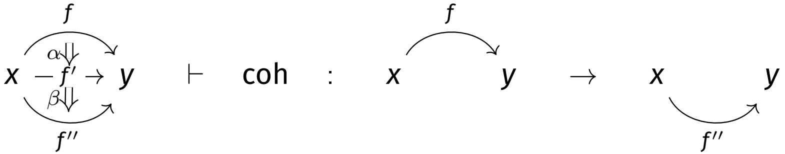
Type-theoretic ω-categories

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

Theorem
This definition coincides with the one of Grothendieck-Maltsiniotis.

Type-theoretic ω-categories

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



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

Type-theoretic ω-categories

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

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

$$\vdash$$

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

Coherences are reversible

Note that if we derive a coherence

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

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

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

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

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

“Demo”

- identity 1-cells

```
coh id {a : .} : a -> a
```

“Demo”

- identity 1-cells

```
coh id {a : .} : a -> a
```

- composition of 1-cells:

```
coh co {a b c : .} (f : a -> b) (g : b -> c) : a -> c
```

“Demo”

- identity 1-cells

```
coh id {a : .} : a -> a
```

- composition of 1-cells:

```
coh co {a b c : .} (f : a -> b) (g : b -> c) : a -> c
```

- associativity of composition of 1-cells:

```
coh assoc {a b c d : .} (f : a -> b) (g : b -> c) (h : c -> d) :
          co (co f g) h -> co f (co g h)
```

“Demo”

- identity 1-cells

```
coh id {a : .} : a -> a
```

- composition of 1-cells:

```
coh co {a b c : .} (f : a -> b) (g : b -> c) : a -> c
```

- associativity of composition of 1-cells:

```
coh assoc {a b c d : .} (f : a -> b) (g : b -> c) (h : c -> d) :
          co (co f g) h -> co f (co g h)
```

- …

“Demo”

- identity 1-cells

```
coh id {a : .} : a -> a
```

- composition of 1-cells:

```
coh co {a b c : .} (f : a -> b) (g : b -> c) : a -> c
```

- associativity of composition of 1-cells:

```
coh assoc {a b c d : .} (f : a -> b) (g : b -> c) (h : c -> d) :
          co (co f g) h -> co f (co g h)
```

- ...
- no inverses:

```
coh inv {a b : .} (f : a -> b) : b -> a
```

  produces and error!

Part VI

Unbiased cartesian closed categories

Next steps

Previous work is nice but

- the type theory is very limited (no Σ- or Π-types, etc.)
- we would like to be able to consider categories with structure ([locally] cartesian [closed]…)

Here

1. we restrict to 1-categories
2. we extend with products and internal homs

If we restrict our theory for weak ω-categories to consider that 2-cells (and higher are identities), we obtain a theory for **unbiased categories**:

$$x \xrightarrow{f} y \xrightarrow{g} z \xrightarrow{h} w \quad \vdash \quad \mathsf{coh} \quad : \quad x \quad \to \quad w$$

Unbiased cartesian closed 1-categories

If we restrict our theory for weak ω-categories to consider that 2-cells (and higher are identities), we obtain a theory for **unbiased categories**:

$$x \xrightarrow{f} y \xrightarrow{g} z \xrightarrow{h} w \quad \vdash \quad \mathsf{coh} \quad : \quad x \quad \to \quad w$$

Our aim is to extend this to have a definition for **unbiased cartesian closed categories** (which could hopefully extend in higher dimensions).

Unbiased cartesian closed 1-categories

There are various definition of cartesian closed categories:

- the traditional categorical definition
- simply-typed λ-calculus
- combinatory logic (I, K, S)
- categorical combinators
- …

We want here

- an agnostic approach in which we could implement most of the above
- a "nice" definition which does not require substitution/α-conversion, weird rules, etc.
- on the long term, we would like an "equality-free" definition of MLTT…

Simply-typed λ-calculus

We consider simply-typed λ-calculus where types are

$$A \quad ::= \quad X \quad | \quad A \to B \quad | \quad \ldots$$

terms are

$$t \quad ::= \quad x \quad | \quad \lambda x^A.t \quad | \quad t\,u$$

rules are

$$\frac{}{\Gamma, x : A, \Delta \vdash x : A}$$

$$\frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\,u : B} \qquad\qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x^A.t : A \to B}$$

and equality is *extensional equality*

$$(\lambda x^A.t)\,u = t[u/x] \qquad\qquad t = \lambda x^A.t\,x$$

Note: we consider the implicational fragment for simplicity

Simply typed λ-calculus

A λ-term is a *normal form* when it is normal with respect to β-reduction

$$(\lambda x^A.t)\,u \qquad \rightsquigarrow \qquad t[u/x]$$

Theorem
Any typable λ-term is β-equivalent to a unique (η-long) normal form.

Such a term is of the from

$$\lambda x_1 x_2 \ldots x_n . x_i t_1 t_2 \ldots t_k$$

with t_i normal forms.

Contractible types

We say that a type is

- **propositional** when there is at most one inhabitant (modulo extensional equality)
- **contractible** when there is exactly one inhabitant

Contractible types

We say that a type is

- **propositional** when there is at most one inhabitant (modulo extensional equality)
- **contractible** when there is exactly one inhabitant

For instance:

- $(A \to B) \to A \to B$ is
- $B \to A$ is
- $(A \to A) \to (A \to A)$ is

Contractible types

We say that a type is

- **propositional** when there is at most one inhabitant (modulo extensional equality)
- **contractible** when there is exactly one inhabitant

For instance:

- $(A \to B) \to A \to B$ is contractible
- $B \to A$ is
- $(A \to A) \to (A \to A)$ is

Contractible types

We say that a type is

- **propositional** when there is at most one inhabitant (modulo extensional equality)
- **contractible** when there is exactly one inhabitant

For instance:

- $(A \to B) \to A \to B$ is contractible
- $B \to A$ is propositional
- $(A \to A) \to (A \to A)$ is

Contractible types

We say that a type is

- **propositional** when there is at most one inhabitant (modulo extensional equality)
- **contractible** when there is exactly one inhabitant

For instance:

- $(A \to B) \to A \to B$ is contractible
- $B \to A$ is propositional
- $(A \to A) \to (A \to A)$ is not contractible

Komori's conjecture

We write $A \leq B$ when B can be obtained from A by replacing variables, for instance

$$A \to A \leq (A \to B) \to (A \to B)$$

Komori's conjecture

We write $A \leq B$ when B can be obtained from A by replacing variables, for instance

$$A \to A \leq (A \to B) \to (A \to B)$$

From Yuichi Komori. *BCK algebras and lambda calculus*. In Proceedings of the 10th Symposium on Semigroups, pages 5–11, 1987:

Conjecture

A minimal provable type is contractible.

For instance,

$$(A \to A) \to (A \to A)$$

is not contractible, but this is the case of the smaller

$$(A \to B) \to A \to B$$

Komori's conjecture

We write $A \leq B$ when B can be obtained from A by replacing variables, for instance

$$A \to A \leq (A \to B) \to (A \to B)$$

From Yuichi Komori. *BCK algebras and lambda calculus.* In Proceedings of the 10th Symposium on Semigroups, pages 5–11, 1987:

Conjecture

A minimal provable type is contractible.

For instance,

$$(A \to A) \to (A \to A)$$

is not contractible, but this is the case of the smaller

$$(A \to B) \to A \to B$$

This does not hold, but we will see that contractible types still generate all proofs.

Komori's conjecture

The conjecture does hold for formulas of depth ≤ 2

Komori's conjecture

The conjecture does hold for formulas of depth ≤ 2
but various counter-examples to the conjecture were found:

- Mint'90: $((((A \to B) \to A) \to A) \to B) \to B$
- Aoto'99: $((A \to B) \to A) \to ((A \to B) \to B)$

```
fun x -> fun y -> y (x (fun z -> y z))
fun x -> fun y -> y (x (fun z -> y (x (fun z -> y z))))
fun x -> fun y -> y (x (fun z -> y (x (fun z -> y (x (fun z -> y z))))))
...
```

Propositional formulas

People tried to come up with conditions which would imply that a formula has at most one proof:

- Hirokawa'93:
  in implication fragments of BCI and BCK (no contraction), minimal inhabited types are contractible.
- Aoto'99:
  provable without *non-prime contraction*, i.e. an implication introduction rule whose canceled assumption differs from a propositional variable and appears more than once in the proof.

Propositional formulas

Another trend of work is in

- Mint'82, Babaev&Solov'ev'82 (coherence for CCC):
  a formula which is balanced (no variable occurs more than twice) admits at most one inhabitant

For instance,

- $(A \to B) \to (A \to B)$ is balanced and thus contractible
- $(A \to B) \to (B \to A)$ is balanced by not inhabited
- $(A \to B \to C) \to (A \to B) \to A \to C$ is not balanced but contractible (this is S!)

Propositional formulas

This is generalized in

- Takahito Aoto and Hiroakira Ono. *Uniqueness of normal proofs in $\{\to, \wedge\}$-fragment of NJ*. Technical Report IS-RR-94-0024F, School of Information Science, JAIST, 1994. Research report.
- Pierre Bourreau and Sylvain Salvati. *Game semantics and uniqueness of type inhabitance in the simply-typed λ-calculus*. In 10th International Conference, TLCA 2011, Novi Sad, Serbia, June 1-3, 2011. Proceedings, volume 6690 of LNCS, pages 61–75. Springer, 2011.
- Sabine Broda and Luiís Damas. *On long normal inhabitants of a type*. J. Log. Comput., 15(3):353–390, 2005.

Propositional formulas

A type is **non-negatively duplicating** when every variable has at most one negative occurrence, e.g.

$$S \quad : \quad (A^+ \to B^+ \to C^-) \to (A^+ \to B^-) \to A^- \to C^+$$

Propositional formulas

A type is **non-negatively duplicating** when every variable has at most one negative occurrence, e.g.

$$\mathsf{S} \quad : \quad (A^+ \to B^+ \to C^-) \to (A^+ \to B^-) \to A^- \to C^+$$

Theorem

A non-negatively duplicating type is propositional.

Propositional formulas

A type is **non-negatively duplicating** when every variable has at most one negative occurrence, e.g.

$$S \quad : \quad (A^+ \to B^+ \to C^-) \to (A^+ \to B^-) \to A^- \to C^+$$

Theorem

A non-negatively duplicating type is propositional.

Proof.

By cut-elimination, we can look for a proof which corresponds to a λ-term which is β-normal and η-long (we take as many variables as there are arrows), i.e. terms of the form

$$\lambda x_1 x_2 \ldots x_n . x_i t_1 t_2 \ldots t_k$$

the choice of the head variable must have a type whose target is *the* negative occurrence of the variable being proved. □

Propositional formulas

Proof search in $\beta\eta$-long form can be performed with

- the introduction

$$\frac{\Gamma, x_1 : A_1, \ldots, x_n : A_n \vdash t : B}{\Gamma \vdash \lambda x_1 \ldots x_n . t : A_1 \to \ldots \to A_n \to B} \; (\to_{\mathrm{I}})$$

  where B is not an arrow,

- the elimination

$$\frac{\Gamma \vdash t_1 : A_1 \qquad \ldots \qquad \Gamma \vdash t_n : A_n}{\Gamma \vdash x\, t_1 \ldots t_n : X} \; (\to_{\mathrm{E}})$$

  with $x : A_1 \to \ldots \to A_n \to X$ in Γ.

Propositional formulas

$$\vdash \qquad : (A^+ \to B^+ \to C^-) \to (A^+ \to B^-) \to A^- \to C^+$$

Propositional formulas

$$\frac{f : A^+ \to B^+ \to C^-, g : A^+ \to B^-, x : A^- \vdash \qquad\qquad : C^+}{\vdash \lambda fgx. \qquad\qquad : (A^+ \to B^+ \to C^-) \to (A^+ \to B^-) \to A^- \to C^+}$$

Propositional formulas

$$\dfrac{\dfrac{\Gamma \vdash \quad : A^{+} \qquad\qquad \Gamma \vdash \qquad : B^{+}}{f : A^{+} \to B^{+} \to C^{-}, g : A^{+} \to B^{-}, x : A^{-} \vdash f \qquad\qquad : C^{+}}}{\vdash \lambda fgx.f \qquad\qquad : (A^{+} \to B^{+} \to C^{-}) \to (A^{+} \to B^{-}) \to A^{-} \to C^{+}}$$

Propositional formulas

$$\dfrac{\dfrac{\overline{\Gamma \vdash x : A^+} \qquad \Gamma \vdash \quad : B^+}{f : A^+ \to B^+ \to C^-, g : A^+ \to B^-, x : A^- \vdash f\,x \qquad : C^+}}{\vdash \lambda fgx.f\,x \qquad : (A^+ \to B^+ \to C^-) \to (A^+ \to B^-) \to A^- \to C^+}$$

Propositional formulas

$$\dfrac{\dfrac{}{\Gamma \vdash x : A^+} \qquad \dfrac{\Gamma \vdash \quad : A^+}{\Gamma \vdash g \quad : B^+}}{\dfrac{f : A^+ \to B^+ \to C^-, g : A^+ \to B^-, x : A^- \vdash f\, x\, (g \quad) : C^+}{\vdash \lambda fgx.f\, x\, (g \quad) : (A^+ \to B^+ \to C^-) \to (A^+ \to B^-) \to A^- \to C^+}}$$

Propositional formulas

$$\dfrac{\dfrac{}{\Gamma \vdash x : A^+} \qquad \dfrac{\dfrac{}{\Gamma \vdash x : A^+}}{\Gamma \vdash g\,x : B^+}}{\dfrac{f : A^+ \to B^+ \to C^-, g : A^+ \to B^-, x : A^- \vdash f\,x\,(g\,x) : C^+}{\vdash \lambda fgx.f\,x\,(g\,x) : (A^+ \to B^+ \to C^-) \to (A^+ \to B^-) \to A^- \to C^+}}$$

Contractible formulas

A (apparently new) remark is that in the rule

$$\frac{\Gamma \vdash t_1 : A_1 \qquad \dots \qquad \Gamma \vdash t_n : A_n}{\Gamma \vdash x\,t_1 \dots t_n : X} \ (\to_E)$$

(with $x : A_1 \to \dots \to A_n \to X$ in Γ), we never use x in the t_i (otherwise the proof would be "infinite" by determinism). Because of this,

Proposition

Contractibility is (very easily) decidable.

Pasting types

Definition
A type A is a **pasting type** when it is non-negatively duplicated and inhabited.

Pasting types

Definition

A type A is a **pasting type** when it is non-negatively duplicated and inhabited.

Proposition

Any pasting type is contractible.

Pasting types

Definition
A type A is a **pasting type** when it is non-negatively duplicated and inhabited.

Proposition
Any pasting type is contractible.

Definition
We say that

$$A_1, \ldots, A_n \vdash A$$

is a **pasting scheme** when

$$A_1 \to \ldots \to A_n \to A$$

is a pasting type.

Pasting types

The rules for **pasting types** are

$$\frac{\Theta, \mathrm{tgt}(A); \Gamma, A \vdash_{\mathrm{ps}} B}{\Theta; \Gamma \vdash_{\mathrm{ps}} A \to B}$$

when $\mathrm{tgt}(A) \notin \Theta$, and

$$\frac{\Theta; \Gamma, \Gamma' \vdash A_1 \qquad \ldots \qquad \Theta; \Gamma, \Gamma' \vdash A_n}{\Theta; \Gamma, A_1 \to \ldots \to A_n \to A, \Gamma' \vdash A}$$

when A is a variable.

Pasting types

The rules for **pasting types** are

$$\frac{\Theta, \mathrm{tgt}(A); \Gamma, A \vdash_{\mathrm{ps}} B}{\Theta; \Gamma \vdash_{\mathrm{ps}} A \to B}$$

when $\mathrm{tgt}(A) \notin \Theta$, and

$$\frac{\Theta; \Gamma, \Gamma' \vdash A_1 \qquad \ldots \qquad \Theta; \Gamma, \Gamma' \vdash A_n}{\Theta; \Gamma, A_1 \to \ldots \to A_n \to A, \Gamma' \vdash A}$$

when A is a variable.

This is in between non-negatively duplicated and deterministic.

CCCaTT

The type theory **CCCaTT** has rules

$$\frac{}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash a : \star \qquad \Gamma \vdash b : \star}{\Gamma \vdash a \to b : \star} \qquad \frac{\Gamma \vdash a : \star}{\Gamma \vdash a}$$

$$\frac{\Gamma \vdash_{\mathsf{ps}} a}{\Gamma \vdash \mathsf{coh} : a} \qquad \frac{\Gamma \vdash_{\mathsf{ps}} a \qquad \Gamma \vdash t : a \qquad \Gamma \vdash u : a}{\Gamma \vdash t = u : a}$$

plus

- $=$ is a congruence
- closure under substitution so that

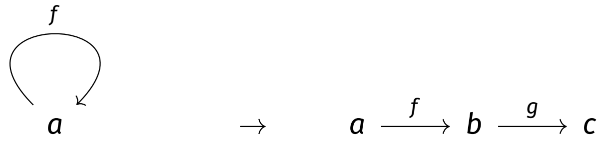
$$\frac{\Delta \vdash \sigma : \Gamma \qquad \Gamma \vdash t : a}{\Delta \vdash t[\sigma] : a[\sigma]}$$

  is derivable

Substitutions

Note that substitutions can replace both types and morphisms.

For instance, we have a substitution



$$(a : \star, f : a \to a) \qquad \vdash \qquad \langle a, a, a, f, g \rangle : (a : \star, b : \star, c : \star, f : a \to b, g : b \to c)$$

CCCaTT

For instance, we can derive

- $\mathsf{I} = \lambda x.x : A \to A$
- $\mathsf{K} = \lambda xy.x : A \to B \to A$
- $\mathsf{S} = \lambda fgx.fx(gx) : (A \to B \to C) \to (A \to B) \to A \to C$
- application $f : A \to B, x : A \vdash fx : B$
- expected equalities such as $\mathsf{I}\ x = x$

`https://cccatt.mimram.fr/`

Combinatory logic

Combinatory logic is defined as the closure under applications of I, K and S.

Combinatory logic

Combinatory logic is defined as the closure under applications of I, K and S.

With rules

$$\mathsf{I}\,t = t \qquad \mathsf{K}\,tu = t \qquad \mathsf{S}\,tuv = tv(uv)$$

Combinatory logic

Combinatory logic is defined as the closure under applications of I, K and S.

With rules

$$\mathsf{I}\,t = t \qquad\qquad \mathsf{K}\,tu = t \qquad\qquad \mathsf{S}\,tuv = tv(uv)$$

along with

- $\mathsf{S}(\mathsf{S}(\mathsf{K}\,\mathsf{S})(\mathsf{S}(\mathsf{K}\,\mathsf{K})(\mathsf{S}(\mathsf{K}\,\mathsf{S})\,\mathsf{K})))(\mathsf{K}\,\mathsf{K}) = \mathsf{S}(\mathsf{K}\,\mathsf{K})$
- $\mathsf{S}(\mathsf{S}(\mathsf{K}\,\mathsf{S})\,\mathsf{K})(\mathsf{K}\,\mathsf{I}) = \mathsf{I}$
- $\mathsf{S}(\mathsf{K}\,\mathsf{I}) = \mathsf{I}$
- $\mathsf{S}(\mathsf{K}\,\mathsf{S})(\mathsf{S}(\mathsf{K}\,\mathsf{K})) = \mathsf{K}$
- $\mathsf{S}(\mathsf{K}(\mathsf{S}(\mathsf{K}\,\mathsf{S})))(\mathsf{S}(\mathsf{K}\,\mathsf{S})(\mathsf{S}(\mathsf{K}\,\mathsf{S}))) = \mathsf{S}(\mathsf{S}(\mathsf{K}\,\mathsf{S})(\mathsf{S}(\mathsf{K}\,\mathsf{K})(\mathsf{S}(\mathsf{K}\,\mathsf{S})(\mathsf{S}(\mathsf{K}(\mathsf{S}(\mathsf{K}\,\mathsf{S})))\,\mathsf{S}))))(\mathsf{K}\,\mathsf{S})$

Combinatory logic vs λ-calculus

We can translate from CC to λ:

$$\mathsf{I} = \lambda x.x \qquad \mathsf{K} = \lambda xy.x \qquad \mathsf{S} = \lambda fgx.fx(gx)$$

Combinatory logic vs λ-calculus

We can translate from CC to λ:

$$\mathsf{I} = \lambda x.x \qquad \mathsf{K} = \lambda xy.x \qquad \mathsf{S} = \lambda fgx.fx(gx)$$

On the other side, we have that $[\![\lambda x.t]\!] = \Lambda x.[\![t]\!]$ with

$$\begin{aligned} \Lambda x.x &= \mathsf{I} & \\ \Lambda x.t &= \mathsf{K}\ t & \text{for } x \notin \mathsf{FV}(t) \\ \Lambda x.t\,u &= \mathsf{S}(\Lambda x.t)(\Lambda x.u) & \text{otherwise} \end{aligned}$$

Combinatory logic vs λ-calculus

We can translate from CC to λ:

$$\mathsf{I} = \lambda x.x \qquad \mathsf{K} = \lambda xy.x \qquad \mathsf{S} = \lambda fgx.fx(gx)$$

On the other side, we have that $[\![\lambda x.t]\!] = \Lambda x.[\![t]\!]$ with

$$\begin{aligned} \Lambda x.x &= \mathsf{I} & \\ \Lambda x.t &= \mathsf{K}\ t & \text{for } x \notin \mathsf{FV}(t) \\ \Lambda x.t\,u &= \mathsf{S}(\Lambda x.t)(\Lambda x.u) & \text{otherwise} \end{aligned}$$

For instance,

$$[\![\lambda xy.x]\!] = \mathsf{S}\,(\mathsf{K}\ \mathsf{K})\,\mathsf{I}$$

Combinatory logic vs λ-calculus

We can translate from CC to λ:

$$\mathsf{I} = \lambda x.x \qquad \mathsf{K} = \lambda xy.x \qquad \mathsf{S} = \lambda fgx.fx(gx)$$

On the other side, we have that $[\![\lambda x.t]\!] = \Lambda x.[\![t]\!]$ with

$$\begin{aligned} \Lambda x.x &= \mathsf{I} \\ \Lambda x.t &= \mathsf{K}\ t && \text{for } x \notin \mathsf{FV}(t) \\ \Lambda x.t\,u &= \mathsf{S}(\Lambda x.t)(\Lambda x.u) && \text{otherwise} \end{aligned}$$

For instance,

$$[\![\lambda xy.x]\!] = \mathsf{S}\,(\mathsf{K}\ \mathsf{K})\,\mathsf{I}\ \neq \mathsf{K}$$

Adding products

Type isomorphism in cartesian closed categories is the congruence generated by

$$(A \times B) \times C = A \times (B \times C) \qquad 1 \times A = A$$
$$A \times B = B \times A \qquad A \times 1 = A$$
$$A \to (B \times C) = (A \to B) \times (A \to C) \qquad A \to 1 = 1$$
$$(A \times B) \to C = A \to B \to C \qquad 1 \to A = A$$

Adding products

Type isomorphism in cartesian closed categories is the congruence generated by

$$(A \times B) \times C = A \times (B \times C) \qquad 1 \times A = A$$
$$A \times B = B \times A \qquad A \times 1 = A$$
$$A \to (B \times C) = (A \to B) \times (A \to C) \qquad A \to 1 = 1$$
$$(A \times B) \to C = A \to B \to C \qquad 1 \to A = A$$

We consider the rewriting system

$$A \to (B \times C) \Rightarrow (A \to B) \times (A \to C) \qquad A \to 1 \Rightarrow 1$$
$$(A \times B) \to C \Rightarrow A \to B \to C \qquad 1 \to A \Rightarrow A$$

Adding products

Type isomorphism in cartesian closed categories is the congruence generated by

$$(A \times B) \times C = A \times (B \times C) \qquad 1 \times A = A$$
$$A \times B = B \times A \qquad A \times 1 = A$$
$$A \to (B \times C) = (A \to B) \times (A \to C) \qquad A \to 1 = 1$$
$$(A \times B) \to C = A \to B \to C \qquad 1 \to A = A$$

We consider the rewriting system

$$A \to (B \times C) \Rightarrow (A \to B) \times (A \to C) \qquad A \to 1 \Rightarrow 1$$
$$(A \times B) \to C \Rightarrow A \to B \to C \qquad 1 \to A \Rightarrow A$$

A product $A_1 \times \ldots \times A_n$ is **pasting** when all the A_i are.

Adding products

For combinators we should add

$$\mathsf{P} : A \to B \to A \times B \qquad \mathsf{P}_1 : A \times B \to A \qquad \mathsf{P}_2 : A \times B \to A \qquad \mathsf{T} : \mathsf{1}$$

along with the obvious equations

Adding products

For combinators we should add

$$\mathsf{P} : A \to B \to A \times B \qquad \mathsf{P}_1 : A \times B \to A \qquad \mathsf{P}_2 : A \times B \to A \qquad \mathsf{T} : 1$$

along with the obvious equations

$$\begin{aligned} \mathsf{S}(\mathsf{K}(\mathsf{S}(\mathsf{K}(\mathsf{S}(\mathsf{K}\,\mathsf{P}_1)))))(\mathsf{S}(\mathsf{K}\,\mathsf{S})(\mathsf{S}(\mathsf{K}\,\mathsf{P}))) &= \mathsf{K} \\ \mathsf{S}(\mathsf{K}(\mathsf{S}(\mathsf{K}(\mathsf{S}(\mathsf{K}\,\mathsf{P}_2)))))(\mathsf{S}(\mathsf{K}\,\mathsf{S})(\mathsf{S}(\mathsf{K}\,\mathsf{P}))) &= \mathsf{K}\,\mathsf{I} \\ \mathsf{S}(\mathsf{S}(\mathsf{K}\,\mathsf{S})(\mathsf{S}(\mathsf{K}(\mathsf{S}(\mathsf{K}\,\mathsf{P})))(\mathsf{S}(\mathsf{K}\,\mathsf{P}_1))))(\mathsf{S}(\mathsf{K}\,\mathsf{P}_2)) &= \mathsf{I} \end{aligned}$$

The theorem

We have axiomatized cartesian closed categories.

Theorem
There is a bijection between

- *terms* $\vdash \boldsymbol{t} : \boldsymbol{A}$ *modulo equality (in contexts containing only type definitions),*
- λ*-terms of type* $\boldsymbol{A}$ *modulo* $\beta\eta$*-equality.*

Proof.
$\Rightarrow$ Pasting types are contractible so they correspond to (unique) λ-terms.
$\Leftarrow$ λ-terms can be implemented with combinators, which can be derived in CCCaTT_1. □

Categorical combinators

AA_K:

(Ass)	$(x^{\sigma_3 \Rightarrow \sigma_4} \circ y^{\sigma_2 \Rightarrow \sigma_3}) \circ z^{\sigma_1 \Rightarrow \sigma_2} = x \circ (y \circ z)$
(IdL)	$\mathrm{Id}^{\tau \Rightarrow \tau} \circ x^{\sigma \Rightarrow \tau} = x^{\sigma \Rightarrow \tau}$
(IdR)	$x^{\sigma \Rightarrow \tau} \circ \mathrm{Id}^{\sigma \Rightarrow \sigma} = x$
(Fst)	$\mathrm{Fst}^{\tau_1, \tau_2} \circ \langle x^{\sigma \Rightarrow \tau_1}, y^{\sigma \Rightarrow \tau_2} \rangle = x$
(Snd)	$\mathrm{Snd}^{\tau_1, \tau_2} \circ \langle x^{\sigma \Rightarrow \tau_1}, y^{\sigma \Rightarrow \tau_2} \rangle = y$
(DPair)	$\langle x^{\sigma_1 \Rightarrow \tau_1}, y^{\sigma_1 \Rightarrow \tau_2} \rangle \circ z^{\sigma \Rightarrow \sigma_1} = \langle x \circ z, y \circ z \rangle$
(Beta)	$\mathrm{App}^{\sigma_2, \sigma_3} \circ \langle \Lambda(x^{\sigma_1 \times \sigma_2 \Rightarrow \sigma_3}), y^{\sigma_1 \Rightarrow \sigma_2} \rangle = x \circ \langle \mathrm{Id}^{\sigma_1 \Rightarrow \sigma_1}, y \rangle$
$(D\Lambda)$	$\Lambda(x^{\sigma_1 \times \sigma_2 \Rightarrow \sigma_3}) \circ y^{\sigma \Rightarrow \sigma_1} = \Lambda(x \circ \langle y \circ \mathrm{Fst}^{\sigma, \sigma_2}, \mathrm{Snd}^{\sigma, \sigma_2} \rangle)$
(AI)	$\Lambda(\mathrm{App}^{\sigma, \tau}) = \mathrm{Id}^{(\sigma \Rightarrow \tau) \Rightarrow (\sigma \Rightarrow \tau)}$
(FSI)	$\langle \mathrm{Fst}^{\sigma, \tau}, \mathrm{Snd}^{\sigma, \tau} \rangle = \mathrm{Id}^{\sigma \times \tau \Rightarrow \sigma \times \tau}$
(ass)	$(x^{\sigma_1 \Rightarrow \sigma_2} \circ y^{\sigma \Rightarrow \sigma_1})\, z^{\sigma} = x(yz)$
(fst)	$\mathrm{Fst}^{\sigma_1, \sigma_2}(x^{\sigma_1}, y^{\sigma_2}) = x$
(snd)	$\mathrm{Snd}^{\sigma_1, \sigma_2}(x^{\sigma_1}, y^{\sigma_2}) = y$
(dpair)	$\langle x^{\sigma \Rightarrow \tau_1}, y^{\sigma \Rightarrow \tau_2} \rangle\, z^{\sigma} = (xz, yz)$
(app)	$\mathrm{App}^{\sigma, \tau}(x^{\sigma \Rightarrow \tau}, y^{\sigma}) = xy$
(Quote$_1$)	$\Lambda(\mathrm{Fst}^{\sigma, \sigma_2})\, x^{\sigma} \circ y^{\sigma_1 \Rightarrow \sigma_2} = \Lambda(\mathrm{Fst}^{\sigma, \sigma_1})\, x$
(Quote$_2$)	$\mathrm{App}^{\sigma_2, \sigma_3} \circ \langle x^{\sigma \Rightarrow (\sigma_2 \Rightarrow \sigma_3)} \circ \Lambda(\mathrm{Fst}^{\sigma, \sigma_1})\, y^{\sigma}, z^{\sigma_1 \Rightarrow \sigma_2} \rangle = xy \circ z.$

Abstraction vs meta-abstraction

Because of **ap** we have that if we have a coherence

$$\Gamma \vdash \mathsf{coh} : A \to B$$

then we have a coherence

$$\Gamma, x : A \vdash \mathsf{coh} : B$$

```
coh ap {a b : .} (f : a -> b) (x : a) : b
coh I {a : .} : a -> a
let id {a : .} (x : a) := ap I x
```

Abstraction vs meta-abstraction

Because of ap we have that if we have a coherence

$$\Gamma \vdash \mathsf{coh} : A \to B$$

then we have a coherence

$$\Gamma, x : A \vdash \mathsf{coh} : B$$

```
coh ap {a b : .} (f : a -> b) (x : a) : b
coh I {a : .} : a -> a
let id {a : .} (x : a) := ap I x
```

The converse is true, but at the “meta-level”, which corresponds to λ-abstraction!

Unbiasing equalities

Equalities are (for now) imposed to be congruences, e.g.

$$\frac{\Gamma \vdash t = u \qquad \Gamma \vdash u = v}{\Gamma \vdash t = v}$$

or

$$\frac{\Gamma \vdash t = u \qquad \Gamma \vdash t_1 = u_1 \qquad \dots \qquad \Gamma \vdash t_n = u_n}{\Gamma \vdash t\langle t_1, \dots, t_n\rangle = u\langle u_1, \dots, u_n\rangle}$$

e.g.

```
coh ap-cong
  {a b : .}
  {t t’ : a → b} {u u’ : a}
  (p : t = t’) (q : u = u’) : ap t u = ap t’ u’
```

Unbiasing equalities

Equalities are (for now) imposed to be congruences, e.g.

$$\frac{\Gamma \vdash t = u \qquad \Gamma \vdash u = v}{\Gamma \vdash t = v}$$

or

$$\frac{\Gamma \vdash t = u \qquad \Gamma \vdash t_1 = u_1 \qquad \ldots \qquad \Gamma \vdash t_n = u_n}{\Gamma \vdash t\langle t_1, \ldots, t_n \rangle = u\langle u_1, \ldots, u_n \rangle}$$

e.g.

```
coh ap-cong
  {a b : .}
  {t t' : a → b} {u u' : a}
  (p : t = t') (q : u = u') : ap t u = ap t' u'
```

which is biased…

Unbiasing equalities

Equalities are (for now) imposed to be congruences, e.g.

$$\frac{\Gamma \vdash t = u \qquad \Gamma \vdash u = v}{\Gamma \vdash t = v}$$

or

$$\frac{\Gamma \vdash t = u \qquad \Gamma \vdash t_1 = u_1 \qquad \dots \qquad \Gamma \vdash t_n = u_n}{\Gamma \vdash t\langle t_1, \dots, t_n \rangle = u\langle u_1, \dots, u_n \rangle}$$

e.g.

```
coh ap-cong
  {a b : .}
  {t t' : a → b} {u u' : a}
  (p : t = t') (q : u = u') : ap t u = ap t' u'
```

which is biased…We can also give unbiased rules!

Interesting subsystems

Interesting subsystems can be defined including

- monoidal categories
- symmetric monoidal categories
- cartesian categories

Toward higher dimensions

Claim
We can also define higher-dimensional pasting schemes in order to define cartesian closed $(\infty, 1)$-categories.

Part VII

Conclusion

We have defined simply-typed λ-calculus without anything which looks like reduction / evaluation / substitution.

As being unbiased, this unifies many known definitions of λ-calculus.

This should have applications in homotopy type theory!

This is implemented at

`https://cccatt.mimram.fr/`

Questions?