# The weak $\omega$-groupoid of identity types

## (an application of the theory of weak $\omega$-categories)

(after recent work of Lumsdaine and Garner - van den Berg)

**Pierre-Louis Curien** (CNRS, University Paris 7, and INRIA)

Tianjin, Operads and Universal Algebra Conference : July 5 – July 9, 2010

*Corrected in March 2011*

## A brief description of the research area

A triangle of correspondences :

categories
(objects, morphisms)

proof theory
(theorems, proofs)

programming languages
(types, programs)

Martin-Löf type theory (MLTT) (about 40 years history) is a good blend of a language that “mixes” programs and proofs.

## Goal of the talk

Explain some homotopy aspects of type theory.

Hint at some of its categorical interpretations (fibrations, groupoids, $n$-groupoids). Applications to the separation of a hierarchy of type theories (Warren) (reference 4 below).

Exhibit the underlying weak $\omega$-groupoid structure of identity types in Martin-Löf's type theory (references 1 and 2 below). This yields a sort of free weak $\omega$-category construction, although "free on what" is still under elaboration (reference 3 below).

References :


1. van den Berg, Garner, Types are weak $\omega$-groupoids
2. Lumsdaine, Weak $\omega$-categories from intensional type theory
3. Awodey, Hofstra, Warren, Martin-Löf complexes
4. Warren, Homotopy theoretic aspects of constructive type theory, Ph.D. thesis (2008)

## Plan of the talk

Part I : Syntax : what are (some of the) entities (types, contexts, judgements, terms, rules, equations) governing the formal description of a programming language in general, and of Martin-Löf type theory in particular ? Some questions of decidability. Some hints on the interpretation of type theory in toposes, and in higher-dimensional groupoids.

Part II : The weak $\omega$-category structure of identity contexts.

# PART I

## The main syntactic entities (generic)

- types $A, B, \ldots$
- contexts $\Gamma, \Delta, \ldots$ = lists of variable declarations $x : A$
- terms : $t, d, a, \ldots$
- typing judgements, like $\Gamma \vdash t : A$, or $\Gamma \vdash s = t : A$. Read $t$ as a formal expression depending on parameters listed in $\Gamma$. For example :

  $$m : \mathbb{R} \vdash \lambda x.(x^2 + 3mx - (m - 4)) : \mathbb{R} \to \mathbb{R}$$

  (where $\lambda x.t$ is the *function* mapping $x$ to $t$).
  Compare with the presentation of an algebraic structure by generators ($t$) and relations ($s = t$).

## **The main syntactic entities of** MLTT

– dependent types

$$\begin{array}{l} A \\ B(x) \ \text{ depending on } \ x : A \\ C(x,y) \ \text{ depending on } \ x : A \ \text{ and } \ y : B(x), \quad \text{etc...} \end{array}$$

– contexts $\Gamma, \Delta, \ldots$ = such towers of variable declarations
Prototypical example : $(n : \mathsf{Nat}, l : \mathsf{List}(n))$

– terms : $t, d, a, \ldots$

– typing judgements :

$$\begin{array}{ll} \vdash \Gamma & \vdash \Gamma = \Delta \\ \Gamma \vdash A & \Gamma \vdash A = B \\ \Gamma \vdash t : A & \Gamma \vdash s = t : A \end{array}$$

## **Structural rules of** MLTT

Usually, some constant types and terms are thrown in.

Rules for the reflexivity, symmetry and transitivity of equality judgements

context extension

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

weakening

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

variables

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

substitution

$$\frac{\Gamma \vdash a : A \quad \Gamma, x : A, \Delta(x) \vdash t : B}{\Gamma, \Delta(a) \vdash t(a) : B(a)}$$

## Dependent products

(not in this talk) (there are also dependent sums)

$\Pi$ — type formation

$$\frac{\Gamma, x : A \vdash B(x)}{\Gamma \vdash \prod_{x:A} B(x)}$$

$\Pi$ — type introduction

$$\frac{\Gamma, x : A \vdash t(x) : B(x)}{\Gamma \vdash \lambda x.t(x) : \prod_{x:A} B(x)}$$

$\Pi$ — type elimination

$$\frac{\Gamma \vdash t : \prod_{x:A} B(x) \qquad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{app}(t, a) : B(a)}$$

$\Pi$ — type conversion

$$\frac{\dots}{\Gamma \vdash \mathsf{app}(\lambda x.t(x), a) = t(a)}$$

## Intensional vs extensional type theory

In MLTT, there are two kinds of equality :

– extensional : $\Gamma \vdash a = b : A$

– Intensional : $\Gamma \vdash p : Id_A(a, b)$

If we confuse the two, then (Streicher) the following problem, called *type-checking*, becomes undecidable :

$$\text{Given } \Gamma, A \text{ such that } \Gamma \vdash A, \text{ and given } t, \text{ do we have } \Gamma \vdash t : A\,?$$

Whence the need to have the two theories available (extensional for the user, intensional for the machine). The mathematics is here on the machine side. Think of the concatenation of two slogans :

– algebras as morphisms from syntax (operad) to semantics (an endomorphism operad)

– of a morphism as a compilation from one source language to a target language

## The rules for identity types

formation

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : A}{\Gamma \vdash Id_A(a, b)}$$

introduction

$$\frac{}{\Gamma, x : A \vdash \mathsf{r}(x) : id_A(x, x)}$$

elimination

$$\frac{\begin{array}{c}\Gamma, x : A, y : A, z : Id_A(x, y) \vdash B(x, y, z) \\ \Gamma, x : A \vdash d(x) : B(x, x, \mathsf{r}(x))\end{array}}{\Gamma, x : A, y : A, z : Id_A(x, y) \vdash \mathsf{J}(x \cdot d, x, y, z) : B(x, y, z)}$$

conversion

$$\frac{\Gamma \vdash a : A \quad \Gamma, x : A \vdash d(x) : B(x, x, \mathsf{r}(x))}{\Gamma \vdash \mathsf{J}(x \cdot d, a, a, \mathsf{r}(a)) = d(a) : B(a, a, \mathsf{r}(a))}$$

## Some derived rules

Leibniz

$$\frac{\Gamma \vdash p : Id_A(a,b) \quad \Gamma \vdash e : C(a)}{\Gamma \vdash (p \circ e) : C(b)}$$

composition

$$\frac{\Gamma \vdash p : Id_A(a,b) \quad \Gamma \vdash q : Id_A(b,c)}{(q \circ p) : Id_A(a,c)}$$

coherence

$$\frac{\Gamma \vdash p : Id_A(a,b) \quad \Gamma \vdash q : Id_A(b,c) \quad \Gamma \vdash r : Id(c,d)_A}{\alpha : Id_{Id(a,d)}(r \circ (q \circ p), (r \circ q) \circ p)}$$

where $q \circ p$, $\alpha$, $p \circ e$ are operations obtained from the generators r and J.

Think of $a, b, c$ as points, of $p, q, r$ as paths, and of $\alpha$ as a homotopy.

## A technical remark

In the absence of dependent products, we have to reinforce the rules for identity types by inserting a context $\Delta$ on the left of $\vdash$ :

elimination

$$\frac{\Gamma, x : A, y : A, z : Id_A(x,y), v : \Delta(x,y,z) \vdash B(x,y,z) \qquad \Gamma, x : A, u : \Delta(x,x,\mathsf{r}(x)) \vdash d(x,u) : B(x,x,\mathsf{r}(x))}{\Gamma, x : A, y : A, z : Id_A(x,y), v : \Delta(x,y,z) \vdash \mathsf{J}((x,u)\cdot d, x, y, z, v) : B(x,y,z)}$$

– Proof of Leibniz. Take $\Delta(x,y,z) = C(x)$, $B(x,y,z) = C(y)$ and then instantiate $x,y,z,v$ as $a,b,p,e$, yielding $p \circ e \stackrel{\Delta}{=} \mathsf{J}((x,u)\cdot u, a, b, p, e)$

– Proof of (composition). In Leibniz, replace $a,b,p$ with $b,c,q$, take $C(x) = Id_A(a,x)$ and $p$ for $e$.

## Truncation principles

We set (Warren) :

$$\begin{array}{l} \underline{A}_0() = A \\ \underline{A}_1(a,b) = \underline{A}(a,b) = Id_A(a,b) \\ \underline{A}_2(a,b;p,q) = Id_{\underline{A}(a,b)}(p,q) \quad \ldots \end{array}$$

$$\frac{\Gamma \vdash p : \underline{A}_{n+1}(a_1,b_1;\ldots;a_{n+1},b_{n+1})}{\Gamma \vdash a_{n+1} = b_{n+1} : \underline{A}_n(a_1,b_1;\ldots;a_n,b_n)}\; n - \text{truncation}$$

Note that 0-truncation amounts to collapse the intensional and extensional equalities

Warren : these principles form a strict hierarchy (one can find a weak $n$-groupoid model that validates $n$-truncation and invalidates $n-1$-truncation).

## An aside : strict versus up-to-iso substitution 1/2

The first notion of model for MLTT was provided by Seely in locally cartesian closed categories (i.e. categories all the slice categories of which are cartesian-closed – any topos is a LCCC).

But then substitution $(B(a), t(a))$ is modelled by pullbacks, that is only up to (coherent) isomorphisms.

One can rephrase this as : the Grothendieck fibration associated with the functor $F : \mathbf{C}^{op} \to \mathbf{Cat}$ (sending $c$ to the slice $\mathbf{C}/c$) is non split.

## An aside : strict versus up-to-iso substitution 2/2

Forthcoming work : we would like to revisit “old” works :

- Curien 1991 : Modify MLTT so as to give an explicit account in the syntax for these substitution isos, interpret the modified syntax in any LCCC $\mathbf{C}$ and show that the interpretation is coherent (cf. Mac Lane’s coherence theorem)
- Hofmann 1993 : “Strictify” the target LCCC $\mathbf{C}$ (that is, associate with it a split fibration) and interpret the *original* MLTT in there.

This leads to a picture

$$\begin{array}{lcll} \text{modified MLTT} & \longrightarrow & \mathbf{C} & \qquad\qquad (\mathbf{C}\ \text{LCCC}) \\ \text{MLTT} & \longrightarrow & \text{strictification of } \mathbf{C} & \end{array}$$

which seems to suggest underlying adjunctions relating “modified” and “strictified”, and deformations (cofibrant replacement).

Warning : in the sequel, we deal with strict substitution, and concentrate our attention on the coherence issues of identity types, which are orthogonal to the substitution-up-to-iso issue.

# PART II

## The category of contexts

In the sequel we concentrate our attention on the theory of identity types only, i.e. we work with a syntax that admits (at least, or only, as you want) the rules for identity types. Let us call it $\mathbb{T}$. We define the *category of contexts* $\mathbf{C}(\mathbb{T})$ (also called classifying category) as follows :

– objects : (valid) contexts, modulo extensional equality $\vdash \Gamma = \Delta$. If, say, $\Gamma = x_1 : A_1, x_2 : A_1(x_1)$, we write $x = (x_1, x_2) : \Gamma$.

– morphisms are (again modulo extensional equality) valid sequences of terms $\Gamma \vdash f = (f_1, \ldots, f_n) : \Delta$, i.e.

$$\Delta = y_1 : B_1, \ldots, y_n : B_n(y_1, \ldots, y_{n-1})$$

and

$$x : \Gamma \vdash f_1(x) : B_1 \quad , \ldots , \quad f_n(x) : B_n(f_1(x), \ldots, f_{n-1}(x))$$

– composition by substitution

– identities = tuples of variables

## Key intuition : pasting diagrams as contexts

Consider the pasting diagram represented by the following Batanin tree (see Appendix) :

$$\begin{array}{l} \pi = (\pi_1, \pi_2, \pi_3) \\ \text{where } \ \pi_1 = (\pi_4) \ , \ \pi_2 = () \ , \ \pi_3 = (\pi_5, \pi_6) \\ \text{where } \ \pi_4 = \pi_5 = \pi_6 = () \end{array}$$

and name the angles between the branches as

| level 0 | level 1 | level 2 |
|---|---|---|
| $x, y, z, t$ | $f, g \quad h \quad k, l, m$ | $p \qquad q \quad r$ |

Then we can *transcribe* this information as :

$$\begin{array}{l} \Gamma = x : A \ , y : A \ , z : A \ , t : A, \\ f : \underline{A}(x, y) \ , g : \underline{A}(x, y) \ , h : \underline{A}(y, z) \ , k : \underline{A}(z, t) \ , l : \underline{A}(z, t) \ , m : \underline{A}(z, t), \\ p : \underline{A}_2(x, y; f, g) \ , q : \underline{A}_2(z, t; k, l) \ , r : \underline{A}_2(z, t; l, m) \end{array}$$

## Towers of identity types

**Theorem** (van den Berg - Garner / Lumsdaine) For any type $A$ (in fact, for every context Γ) the following tower $\underline{A}$ of contexts ($x, y : A$ abbreviates $x : A, y : A$) :

$$A(0) = (z_0 : A) \quad A(1) = (x_0, y_0 : A, z_1 : \underline{A}(x_0, y_0)) \qquad \ldots$$
$$A(n) = (x_0, y_0 : A, \ldots, z_n : \underline{A}_n(x_0, y_0; \ldots; x_{n_1}, y_{n-1})) \qquad \ldots$$

has the structure of a weak $\omega$-category (in fact, weak $\omega$-groupoid)

We need
– a globular operad $P$ equipped with a contraction, and
– an action of this operad on $\underline{A}$
Recall also that a globular operad is indexed by *pasting diagrams*.

## Architecture of the proof (following van den Berg - Garner)

1. Define an intermediate axiomatic framework : identity type categories, and (reflexive) globular context objects therein
2. Check that the category of contexts has this structure.
3. Then show that any identity type category with a reflexive globular context carries a weak $\omega$-category structure.

In programming languages terminology, we have encapsulated the details of MLTT in an “abstract data type”.

# STEPS 1 and 2

## Identity type categories

It is a category $\mathbf{C}$ endowed with two classes of morphisms $\mathcal{I}, \mathcal{P} \subseteq \mathbf{C}_1$ such that :

1. $\mathbf{C}$ has a terminal object $1$, and $! : A \to 1 \; \in \; \mathcal{P}$ for all $A \in \mathbf{C}_0$
2. The classes $\mathcal{I}, \mathcal{P}$ contain the identities and are closed under composition
3. Pullbacks of $\mathcal{P}$-maps along arbitrary maps exist, and are again $\mathcal{P}$-maps
4. The pullback of an $\mathcal{I}$-map along a $\mathcal{P}$-map is an $\mathcal{I}$-map
5. Every commutative square with an $\mathcal{I}$-map on the West and a $\mathcal{P}$-map on the East has a diagonal filler
6. For every $\mathcal{P}$-map $p : C \to D$, the diagonal map $\Delta : C \to C \times_D C$ has a factorisation $\Delta = e \circ r$, with $e \in \mathcal{P}$ and $r \in \mathcal{I}$ :

$$C \xrightarrow{r} C^I \xrightarrow{e} C \times_D C$$

(Think of $\mathcal{I}$-maps as trivial cofibrations, and of $\mathcal{P}$-maps as fibrations.).

## The Identity type category structure of contexts 1/4

The category $\mathbf{C}(\mathbb{T})$ is an identity type category, taking

– $\mathcal{I}$ = injective equivalences, which are defined as the context morphisms $f : \Gamma \vdash \Delta$ for which the following rule is derivable :

$$\frac{y : \Delta \vdash \Lambda(y) \qquad x : \Gamma \vdash d(x) : \Lambda(f(x)) \qquad \vdash b : \Delta}{\vdash \mathsf{E}_d(b) : \Lambda(b)}$$

plus the corresponding equality $\vdash \mathsf{E}_d(f(a)) = d(a) : \Lambda(f(a))$

Prototypical case : Take

$(x, x, \mathsf{r}(x)) : (x : A) \to (x : A, y : A, p : Id_A(x, y))$ for $f$
$(a, b, p)$ for $b$

– $\mathcal{P}$ = dependent projections, defined as (partial) tuples of variables

$$x : \Gamma, y : \Delta(x) \vdash x : \Gamma$$

or as composites of *basic* dependent projections $x : \Gamma, y : A(x) \vdash x : \Gamma$.

## The Identity type category structure of contexts 2/4

Proof (that the above yields an identity type category structure) :

1. The empty context is terminal (the unique morphism is the tuple formed by $n = 0$ morphisms)
2. Dependent projections are closed under $n$-ary composition ($n \geq 0$) by definition. For injective equivalences, this comes from the strong version of property 5 (lifting) that is proved below.
3. The pullback of $p : (\Delta, \Lambda(x)) \to \Delta)$ along $f : \Gamma \to \Delta$ is $p : (\Gamma, \Lambda(f(x)) \to \Gamma)$. (Note that if $\Delta \vdash A(x)$ and $f : \Gamma \to \Delta$, then $\Gamma \vdash A(f(x))$.)
4. The proof relies on the following characterisation :

   A context morphism $f : \Gamma \to \Delta$ is an injective equivalence ifif
   - there exists $g : \Delta \to \Gamma$ such that $g \circ f = Id$,
   - there exis ts a term $y : \Delta \vdash p(y) : Id_\Delta(f(g(y)), y)$
   - $x : \Gamma \vdash p(f(x)) = \mathsf{r}(f(x))$

## The Identity type category structure of contexts 3/4

5. Lifting. The judgement $x : \Gamma \vdash d(x) : \Lambda(f(x))$ amounts to having a square $p \circ (f, d) = Id \circ f$, where $p$ is the dependent projection from $(\Delta, \Lambda)$ to $\Delta$. The compound operation $\mathsf{E}_d$ provides the filling (commutation of the upper triangle = the equation satisfied by $\mathsf{E}_d$, while commutation of the lower triangle says that $\mathsf{E}_d$ is a section, i.e., respects fibers.

   Note that we actually have $\mathcal{I} = \mathsf{LLP}(\mathcal{P})$ (LLP= Left Lifting Property).

6. We have actually more. In fact, *every morphism* $f : \Gamma \to \Delta$ (not only the diagonal) factorises as $f = p \circ i$. Define

$$Id(f) = (x : \Gamma, y : \Delta, u : Id_\Delta(fx, y))$$

   Then we can take

$$\begin{array}{l} i(x) = (x, fx, \mathsf{r}(fx)) : \Gamma \to Id(f) \\ p(x, y, u) = y : Id(f) \to \Delta \end{array}$$

In the proof of property 6, we have implicitly extended identity *types* to identity *contexts*.

These can be defined by repeated use of the identity context rules.

## Reflexive globular contexts in an identity type category $\mathbb{C}$ 1/2

A globular context is :

– a globular object in $\mathbb{C}$, given by a collection of objects $A(0), \ldots, A(n), \ldots$ equipped with source and target morphisms $s, t : A(n+1) \to A(n)$ satisfying $s \circ s = s \circ t$ and $t \circ s = t \circ t$ (i.e, a functor $A : \mathbf{G}^{op} \to \mathbf{C}$ (see Appendix)

– such that all the induced maps $q$ from $A(n)$ to its boundary $\partial A(n)$ are $\mathcal{P}$-maps

(definition of boundaries : $\partial A(0) = 1$, $\partial A(n+1) = A(n) \times_{\partial A(n)} A(n)$ with $q$ the universal morphism)

A globular context is *reflexive* when it is moreover equipped

– with morphisms $r : A(n) \to A(n+1)$ such that $s \circ r = t \circ r = Id$

– with all these morphisms being $\mathcal{I}$ maps.

## Reflexive globular contexts in an identity type category $\mathbb{C}$ 2/2

For any context $\Gamma$, the tower of identity contexts over $\Gamma$ forms a reflexive globular object :

$$\begin{array}{ll} \Gamma(0) = (z_0 : \Gamma) & \Gamma(1) = (x_0, y_0 : \Gamma, z_1 : \underline{\Gamma}(x_0, y_0) \\ s = x_0\ ,\ t = y_0 & r = (z_0, z_0, \mathsf{r}(z_0)) \end{array}$$

Actually, this too is part of the axiomatic framework : *any* identity type category admits for *any* of its objects $A$ a reflexive globular object built as a tower over it :

Take $A(0) = A$ (with associated $\partial A(0) = 1$). Suppose we have built $A(n)$ with associated $s, t : A(n) \to A(n-1), q : A(n) \to \partial A(n)$, and $\partial A(n+1) = A(n) \times_{\partial A(n)} A(n)$. Then $A(n+1)$ is defined as the cylinder object for $\Delta = e \circ r : A(n) \to \partial A(n+1)$ (thus with $r : A(n) \to A(n+1) \in \mathcal{I}$), and $s, t : A(n+1) \to A(n)$ are defined by composition of $e$ with the projections $\partial A(n+1) \to A(n)$, from which we get that $q = e$ by universality, hence $q \in \mathcal{P}$.

The above is just the instantiation of this general abstract construction in the category of contexts.

# STEP 3

## The weak $\omega$-category structure 1/3

Let $\mathbf{C}$ be an identity type category, and $A$ an object of $\mathbf{C}$. Let $\underline{A}$ be the associated reflexive globular context.

With every pasting diagram $\pi$ we associate an object $A^\pi$ of $\mathbf{C}$, as follows : if $\pi = ((),())$, then build the diagram formed by three copies of $A(0)$, two copies of $A(1)$ with sources and targets, and take the limit. One can also define source and target maps $\sigma, \tau : A^\pi \to A^{\partial\pi}$ by the universality. (Think of $A^\pi$ as an internal version of $\mathbf{Gset}[\widehat{\pi}, A]$.)

The operad $P$ has as operations of shape $\pi$ (where $\pi$ is of dimension $n$) "ladder" diagrams that stem from $A^\pi$. The horizontal morphisms are from $A^\pi$ to $A(n)$, $A^{\partial\pi}$ to $A^{n-1}$ etc... The action corresponding to this operation is just the top morphism of this ladder.

## The weak $\omega$-category structure 2/3

More precisely, we take $P$ to consist of the *pointed* such diagrams, that is, which live in the slice under $A(0)$, (thanks to the reflexive structure of $\underline{A}$).

This operad is equipped with a contraction, thanks to the lifting axiom 5 :

$$
\begin{tikzcd}
A^{\pi} \arrow[d, shift right, "\tau"'] \arrow[d, shift left, "\sigma"] & A(n) \arrow[d, shift right, "t"'] \arrow[d, shift left, "s"] \\
A^{\partial\pi} \arrow[r, shift left, "f_{n-1}"] \arrow[r, shift right, "g_{n-1}"'] & A(n-1) \\
\vdots & \vdots \\
A(0) & A(0)
\end{tikzcd}
$$

Consider the square with $q : A(n) \to \partial A(n)$ on the East and the reflexivity morphism $r : A(0) \to A^{\pi}$ on the West)

The latter map is an $\mathcal{I}$-map by axioms 1 through 4.

## The weak $\omega$-category structure 3/3

As an illustration, we show how to define composition of 1-morphisms. The intuition is (cf. the derived rule (composition) in MLTT) to extend it from the identity law $Id \circ f = f$.

We can inject $A(1)$ into $A^{((),())}$ by taking the universal map (call it $\mathsf{r}'$) for the commuting square $\mathsf{t} \circ Id = \mathsf{s} \circ (\mathsf{r} \circ \mathsf{t})$.

Contemplating the square $\tau \circ r' = \mathsf{r} \circ t$, we see that it is a pullback since the rectangle $\mathsf{t} \circ (\sigma \circ \mathsf{r}') = (\mathsf{s} \circ \mathsf{r}) \circ \mathsf{t}$ boils down to $\mathsf{t} \circ Id = Id \circ \mathsf{t}$.

Hence $\mathsf{r}'$ is an $\mathcal{I}$-map (axiom 4).

Then (using axiom 5) we define compostion as a chosen diagonal filling in the square

$$\langle \mathsf{s}, \mathsf{t} \rangle \circ Id = \langle \mathsf{s} \circ \sigma, \mathsf{t} \circ \tau \rangle \circ \mathsf{r}'$$

# APPENDIX

Extracted (and corrected in March 2011) from my lectures in the Operads and Universal Algebra Summer School, June 28 – July 2, 2010, Tianjin

## Burroni's notion of T-category

Let $\mathbf{C}$ be a category with pull-backs. Let $T$ be a monad on $\mathbf{C}$. It is well known that the objects of $\mathbf{C}$, the spans of $\mathbf{C}$ (= pairs of morphisms $(f : C \to A, g : C \to B)$), and the morphisms between spans in the same "homset", i.e., between, say $(f, g)$ and $(f' : C' \to A, g' : C' \to B)$ (which are morphisms $C \to C'$ making the obvious diagrams commute) form a bicategory.

Now we can twist this (a Kleisli category construction) into a bicategory which has still the objects of $\mathbf{C}$ as objects, but in which the 1-morphisms from $A$ to $B$ are the spans from $TA$ to $B$.

That this makes a bicategory is equivalent to requiring that $T$ preserves pull-backs and that $\mu$ and $\eta$ (the multiplication and unit of the monad) are cartesian, i.e. all naturality squares are pull-backs.

A $T$-category is a monad (often called a monoid) $(C_1 \to TC_0, C_1 \to C_0)$ in this bicategoy.

When $C_0 = 1$ (the terminal object), we speak of a $T$-operad.

Special case : $(id : T1 \to T1, T1 \to 1)$ is always an operad ("exactly one operation per shape").

## Algebra for a $T$-operad

With a $T$-category $C = (C_0, C_1, f : C_1 \to TC_0, g : C_1 \to C_0)$, one associates a monad (on $\mathbf{E}/C_0$), as follows. Take $h : X \to C_0$. Take the pullback $T_C X = TX \times_{TC_0} C_1$ (of $Th$ and $f$), and define $T_C(h)$ as the right leg of the picture.

Define an algebra for the $T$-category $C$ as an algebra for the monad $T_C$.

## Globular sets

A globular set is an object of $\mathbf{Gset} = \mathbf{Set}^{\mathbf{G}^{op}}$, where $\mathbf{G}$ is defined as follows : the objects of $\mathbf{G}$ are the natural numbers, and as generators we have morphisms $\sigma_n, \tau_n : n-1 \to n$ for every $n$, subject to the equations $\sigma_n \circ \sigma_{n-1} = \tau_n \circ \sigma_{n-1}$ and $\sigma_n \circ \tau_{n-1} = \tau_n \circ \tau_{n-1}$.

(Note that there are only two morphisms in $\mathbf{G}[m,n]$ $(n > m)$, namely $\ldots \circ \sigma_n$ and $\ldots \circ \tau_n$, whatever the $\ldots$ is. For $n < m$, $\mathbf{G}[m,n]$ is empty, and for $m = n$, $\mathbf{G}[m,m]$ has only the identity as endomorphism.)

## The free $\omega$-category monad 1/4

Let 1 be the terminal globular set (constant 1={*} presheaf).

$$(T1)(n) = \text{pasting diagrams of } n - \text{cells}$$

We set $T1 = \mathtt{pd}$ (pasting diagrams), and we define $\mathtt{pd}(n)$, and the (equal) source and target functions (that we write $s = \partial = t$), as follows :

$$\mathtt{pd}(0) = \{\cdot\} \quad \mathtt{pd}(n+1) = \mathtt{pd}(n)^*$$
$$\partial : \mathtt{pd}(n+1) \to \mathtt{pd}(n) = (\partial : \mathtt{pd}(n) \to \mathtt{pd}(n-1))^*$$

It is more intuitive to represent the elements of $\mathtt{pd}(n)$ as trees (the **Batanin trees**) of height $n$, and then the operation $\partial$ amounts to cut it to the level $n-1$, by removing leaves of height $n$.

But the tree representation is not completely precise : we need a tree + a dimension ($\geq$ the actual height of the tree). For example, $(\cdot, \cdot, \cdot)$ and $((), (), ())$ are different pasting diagrams of dimension 1 and 2, resp.

One embeds 1 into $T1$ (unit of the monad) by taking in each $(T1)(n)$ the one branch tree of height $n$

## The free $\omega$-category monad 2/4

One then transports $T$ to all objects as follows :

$$(TX)(m) = \sum_{\pi \in (T1)(m)} \mathbf{Gset}[\widehat{\pi}, X]$$

where $\widehat{\pi}$ is the following globular set : if $\pi = (\pi_1, \ldots, \pi_n)$, then

$$\widehat{\pi}(n+1) = \coprod_{i=1}^{i=n} \widehat{\pi_i}(n) \qquad \widehat{\pi}(0) = \{0, \ldots n\}$$

Alternative concrete description : $\widehat{\cdot}(0) = \{0\}$, and for $\pi = (\pi_1, \ldots, \pi_n)$ :

$$\begin{array}{l} \widehat{\pi}(0) = \{0, \ldots, n\} \\ \widehat{\pi}(n+1) = \{\underline{iu} \cdot j \mid \underline{u} \cdot j \in \widehat{\pi_i}(n)\} \end{array}$$

Thus the elements of $\widehat{\pi}(n)$ are pairs of a word $u$ (over $\mathbb{N}$) which denotes a branch of $\pi$ viewed as a tree and a natural number $i$, written as $\underline{u} \cdot i$ which represents “angles” determined by the branching (with $\underline{\epsilon} \cdot i$ abbreviated as $i$)

The source and target are defined as follows : $s(\underline{ui} \cdot j) = \underline{u} \cdot (i-1)$ , $t(\underline{ui} \cdot j) = \underline{u} \cdot i$

## The free $\omega$-category monad 3/4

The idea for the multiplication from $TT1$ to $T1$ is to zoom on each component of a pasting diagram and to "substitute".

Special case : $n$-ary compositions of pasting diagrams :

$$(\pi_1, \ldots, \pi_n) \star_0 (\rho_1, \ldots, \rho_m) = ((\pi_1, \ldots, \pi_n, \ldots, \rho_1, \ldots, \rho_m)$$
$$(\pi_1, \ldots, \pi_n) \star_i (\rho_1, \ldots, \rho_n) = (\pi_1 \star_{i-1} \rho_1, \ldots, \pi_n \star_{i-1} \rho_n)$$

## The free $\omega$-category monad 4/4

If $X$ is a globular set, define $\mathsf{s}X$ by $(\mathsf{s}X)(n) = X(n+1)$.

A formal definition of the multiplication $\mu$ of the monad (we write $\pi[G] = \mu(G : \widehat{\pi} \to \mathsf{pd})$) :

1. If $\pi = (\pi_1, \ldots, \pi_n)$, then every $G : \widehat{\pi} \to X$ induces $G_i : \widehat{\pi_i} \to (\mathsf{s}X)$ ($i = 1, \ldots, n$).
2. For every pasting diagram $\rho$, every $H : \widehat{\rho} \to \mathsf{s}(\mathsf{pd})$ can be written $H = (H_1, \ldots, H_k)$ (with $H_1, \ldots, H_k : \widehat{\rho} \to \mathsf{pd}$), where $k$ is the common length of $H(i)$ ($i$ being any 0-cell of $\widehat{\rho}$). One proves this using :
   – that $H$ is a morphism of globular sets
   – and that in $\mathsf{pd}$, the source and target functions coincide
3. Let $\pi = (\pi_1, \ldots, \pi_n)$, $G : \widehat{\pi} \to \mathsf{pd}$, inducing $G_{i,1}, \ldots, G_{i,k_i} : \widehat{\pi_i} \to \mathsf{pd}$, by (1) and (2). Then we set (by induction) :

$$\pi[G] = (\pi_1[G_{i,1}], \ldots, \pi_1[G_{1,k_1}], \pi_2[G_{2,1}], \ldots, \pi_n[G_{n,k_n}])$$

## Globular operads

A $T$-operad, for $T =$ the free $\omega$-category monad on $\mathbf{Gset}$, is called a *globular operad* (Batanin).

The general definition of $T$-algebra instantiates as follows for globular operads. If $P$ is a globular operad, then the associated monad $T_P$ is defined as follows :

$$(T_P A)(m) = \coprod_{\pi \in \mathsf{pd}(m)} P(\pi) \times \mathbf{Gset}[\widehat{\pi}, A]$$

The globular operad of (strict) $\omega$-categories is just $id : T1 \to T1$ (just one operation for each pasting diagram, cf. the non symmetric operad for associative algebras).

## Towards weak $\omega$-categories

The intuition is that pasting diagrams are enough to describe strict $\omega$-categories, because no matter how you got it by pasting smaller pasting diagrams together, the result will be the same.

What we now need is a *deformation*, that allows us to speak about all of these possible ways of assembling them, and to fill the space between these different compound composites (coherence).

So we need a globular set $P$, a *globular operad* $d : P \to T1$, and a *contraction* on $d$.

## The notion of contraction

Consider a morphism $p : A \to \mathtt{pd}$ of globular sets. We say that $p$ is a contraction (Leinster's definition) if for any pasting diagram $\pi$, for any two cells $d, d'$ in $A^{\partial\pi}$, there exists a cell in $A^{\pi}$ between them.

Note that contractions provide us *both*

– enough composition laws
  for example, given the pasting diagram $(.,.)$ and given two 0-cells $c, d$, contraction gives us *a* composition law for two consectutive 1-morphisms $f : c \to e$ and $g : e \to d$ for some $e$ a compostion

– enough coherence cells
  for example, given the pasting diagram $((),(),())$ and compositions $(h \circ g) \circ f$ and $h \circ (g \circ f)$, contraction gives us a 2-cell between them

## Leinster's definition of weak $\omega-$category

At last, we can give the definition of weak $\omega-$category :

A weak $\omega$-category is an algebra for a globular operad equipped with a contraction.

Equivalently :

A weak $\omega$-category is an algebra for the initial globular operad $L$, equipped with a contraction.

Theorem (Leinster) : this initial operad exists ! (It's sort of easy to convince oneself by adding cells from scratch, but proving it is a bit more delicate. In particular, the two aspects : "globular operad", and "contraction" do not commute (in the sense of the theory of monads), which makes things a bit tricky. Nice analysis of this by Eugenia Cheng.)

## Initial operad with contraction as cofibrant replacement

Endow $\mathbf{Gset}$ with a nwfs realising the wfs that is cofibrantly generated by the sphere maps. (the $n$-dimensional sphere map has $\mathcal{Y}on(n)$ as codomain)

Transport the nwfs along the adjunction between the categories of globular operads and globular sets (free construction left adjoint to forgetful functor)

Theorem (Garner) : The initial globular operad with contraction is a cofibrant replacement of the globular operad for (strict) $\omega$-categories

Reference : his paper “A homotopy-theoretic universal property of Leinster’s operad for weak $\omega$-categories

## Examples of weak $\omega$-categories

But these crazy theoretical computer scientists never give examples ?

Strict $\omega$-categories are weak $\omega$-categories

(Infinitely) iterated loop spaces (the historically motivating example, next slide)

An example coming from programming languages / type theory : the weak $\omega$-groupoid of identity types in Martin-Lôf type theory (Lumsdaine, and Garner-van den Berg) : my talk next week !

## The fundamental $\omega$-groupoid 1/3

Let $X$ be a topological space. Let $D^n$ be the $n$-disk. This provides a functor from $\mathbf{G}$ to the category $\mathbf{Top}$ of topological spaces, where the two morphisms from $D^{n-1}$ to $D^n$ are obtained by first cutting the sphere $S^n$ in two halves (upper and lower), and then embedding $D^{n-1}$ in $D_n$ as these two halves.

By a general abstract machinery (cf. Gabriel-Zisman), any functor $F : \mathbf{C} \to \mathbf{D}$ induces and adjunction $G \dashv H$ between $\mathbf{D}$ and $\mathbf{Set}^{\mathbf{C}^{op}}$, with $Gd = \mathbf{D}[F\_, d]$, and $H$ the left Kan extension of $F$ along the Yoneda embedding $\mathbf{C} \to \mathbf{Set}^{\mathbf{C}^{op}}$.

In our case, $G$ instantiates as $\Pi_\omega$, the "singular globular set" construction :

$$\Pi_\omega X(n) = \mathbf{Top}[D^n, X]$$

and $H$ instantiates as the geometric realisation functor, written $X \mapsto |X|$.

## The fundamental $\omega$-groupoid 2/3

We now exhibit a weak $\omega$-category structure on $A = \Pi_\omega(X)$.
We define a globular operad as follows : $P(\pi)$ consists of the (continuous) maps from $D^n$ to $|\widehat{\pi}|$ that *hereditarily* respect the boundaries.

For example, for $n = 2$, the restriction to the upper part of the boundary of the disk should map to the realisation of the source of $\pi$, in such a way that the endpoints of this half-circle are mapped to the source of the source and target of the source of $\pi$, respectively.

That this operad is contractible is an easy consequence of the following two facts :

– each $|\widehat{\pi}|$ is a contractible space
– in a contractible space any map from a sphere can be extended to the disk of which it is a boundary.

## The fundamental $\omega$-groupoid 3/3

The algebra structure : for each $\pi$ of dimension $n$, for each $\alpha \in P(\pi)$, we have to exhibit a map from $\mathbf{Gset}[\widehat{\pi}, A]$ to $A(n)$, which by adjoint transposition amounts to giving a map from $\mathbf{Top}[|\widehat{\pi}|, X]$ to $A(n)$, which is obtained straightforwardly by composition (keep in mind that $\alpha : D^n \to |\widehat{\pi}|$ and that $A(n) = \mathbf{Top}[D^n, X]$).

The operad structure. An element $\alpha \in P(\pi)$ can be considered as (a cascade of) natural transformations $\mathbf{Top}[|\widehat{\pi}|, \_] \to \mathbf{Top}[D^n, \_]$. Fixing $S$, and using again the adjunction, this gives us (a cascade of) functions $\mathbf{Gset}[\widehat{\pi}, \Pi_\omega(S)] \to \Pi_\omega(S)(n)$ : endomorphism operad !

(compare with $O(n) = \mathbf{Set}[A^n, A]$, and read $A^n$ as $\mathbf{Set}[\widehat{n}, A]$, where $\widehat{n}$ is an $n$ elements set associated with number/shape $n$)