Category theory in Univalent Foundations

Benedikt Ahrens[1], Krzysztof Kapulkin[2], Michael Shulman[1]

[1] Institute for Advanced Study, Princeton
[2] University of Pittsburgh, Pittsburgh

University Potsdam, June 26, 2013

3 kinds of sameness for categories

Equality	$\mathcal{C} = \mathcal{D}$
Isomorphism	$\mathcal{C} \cong \mathcal{D}$
Equivalence	$\mathcal{C} \simeq \mathcal{D}$

- most properties of categories invariant under **equivalence**
- we can only substitute **equals for equals**
- in set-theoretic foundations these notions are worlds apart

In this talk:

Define categories in the **Univalent Foundations** for which all three coincide

Outline

Univalent Foundations

What are the Univalent Foundations?

- Intensional Martin-Löf Type Theory

$\rightsquigarrow$ *Types as Spaces* interpretation, i.e. Homotopy Type Theory

$+$ **Univalence Axiom**

The 4 Kinds of Judgements of Type Theory

Contexts & Judgements

Γ	sequence of variable declarations
	$x_1 : A_1, x_2 : A_2(x_1), \ldots, x_n : A_n(\vec{x}_i)$
$\Gamma \vdash A$	A is well–formed **type** in context Γ
$\Gamma \vdash a : A$	**term** a is of type A in context Γ
$\Gamma \vdash A \equiv B$	types A and B are **convertible**
$\Gamma \vdash a \equiv b : A$	a is convertible to b in type A

In particular: dependent type B over A

$$x : A \vdash B(x)$$

Specifying a type

A type is specified by 4 pieces of information:

1. **Type former:** declaring a new type
2. **Term former:** way to construct terms of this type
3. **Elimination:** way to use terms of type 1 to construct other terms
4. **Computation:** what if 2 followed by 3

Example (Function types)

1. if A and B are types, then $A \to B$ is a type
2. if $\quad \Gamma, (x : A) \vdash b(x) : B \quad$ then $\quad \Gamma \vdash \lambda x.b(x) : A \to B$
3. if $f : A \to B$ and $a : A$, then $f@a : B$
4. $\lambda x.b(x)@a \quad \equiv \quad b[x := a]$

Example II: Dependent Product

1. if $\Gamma, x : A \vdash B(x)$ then $\Gamma \vdash \prod_{x:A} B(x)$
2. if $x : A \vdash b(x) : B(x)$ then $\vdash \lambda x.b(x) : \prod_{x:A} B(x)$
3. if $f : \prod_{x:A} B(x)$ and $a : A$ then $f@a : B(a)$
4. $\lambda x.b(x)@a \quad \equiv \quad b[x := a]$

Remark

If B does not depend on x in 1 , we obtain $A \to B$

Convention 1: omit leading Γ
Convention 2: omit leading $\vdash$ when context is empty

Example III: Dependent Sum

1. if $x : A \vdash B(x)$ then $\sum_{x:A} B(x)$ is a type
2. if $a : A$ and $b : B(a)$ then $(a, b) : \sum_{x:A} B(x)$
3. if $p : \sum_{x:A} B(x)$ then $fst(p) : A$ and $snd(p) : B(fst(p))$
4. $fst(a, b) \equiv a$ and $snd(a, b) \equiv b$

Remark

If B does not depend on x in 1, we obtain $A \times B$.

Martin-Löf TT and its Homotopy Interpretation

Type theory	Notation	Interpretation
Inhabitant	$a : A$	a is a point in space A
Dependent type	$x : A \vdash B(x)$	fibration $\coprod_{x:A} B(x) \to A$
Sigma type	$\sum_{x:A} B(x)$	total space of a fibration
Product type	$\prod_{x:A} B(x)$	space of sections of a fibration
Coproduct type	$A + B$	disjoint union
Identity type	$\mathrm{Id}_A(a, b)$	space of **paths** $p : a \rightsquigarrow b$

- other types as needed (type $\mathbf{N}$ of naturals, empty type)

Identity Type

1. $(x, y) : A \times A \vdash \mathrm{Id}_A(x, y)$
2. if $\quad a : A \quad$ then $\quad \mathrm{refl}(a) : \mathrm{Id}_A(a, a)$
3. if $\quad p : \mathrm{Id}(x, y) \quad$ and $\quad x : A \vdash c : C(x, x, \mathrm{refl}(x)) \quad$ then $\ldots \vdash J(c, x, y, p) : C(x, y, p)$
4. ...

For the Homotopy Interpretation

- we do not assume Identity Reflection
- we do not assume Uniqueness of Identity Proofs (UIP)

Identity terms form a setoid

For any type A, the pair (A, Id_A) forms a **setoid**:

$$\mathrm{id}/\mathrm{refl} : 1 \to \mathrm{Id}_A(a, a) \qquad (a : A)$$

$$(\_)^{-1}/\mathrm{sym} : \mathrm{Id}_A(a, b) \to \mathrm{Id}_A(b, a) \qquad (a, b : A)$$

$$\star\, /\mathrm{trans} : \mathrm{Id}_A(a, b) \times \mathrm{Id}_A(b, c) \to \mathrm{Id}_A(a, c) \qquad (a, b, c : A)$$

Example (Symmetry)

- want to prove $\mathrm{Id}_A(x, y) \to \mathrm{Id}_A(y, x)$
- assume $p : \mathrm{Id}_A(x, y)$, need to construct term of $\mathrm{Id}_A(y, x)$
- by 3 assume y is x, suffices to give term of $\mathrm{Id}_A(x, x)$
- take this term to be $\mathrm{refl}(x)$

The homotopy interpretation of identity types

Type theory	Interpretation	Notation
reflexivity	constant path on a	$\mathrm{refl}_A(a)$
symmetry	path reversal	p^{-1}
transitivity	path concatenation	$p \star q$
higher identity type	paths between paths	

Identity terms form a groupoid

Theorem: (A, Id_A) forms a groupoid [Hofmann & Streicher '98]

- up to propositional equality, i.e. up to paths between paths:
- $\alpha_{p,q,r} : \mathsf{Id}_{\mathsf{Id}(a,d)}(p \star (q \star r), (p \star q) \star r)$
- $\beta_p : \mathsf{Id}_{\mathsf{Id}(a,a)}(p \star p^{-1}, \mathsf{id}) \quad \ldots$
- $\gamma_p : \mathsf{Id}_{\mathsf{Id}(a,b)}(\mathsf{refl}(a) \star p, p) \quad \ldots$

Identity terms form a groupoid à la Aczel & Huet–Saïbi

Such a groupoid is given by:

- a **type** of objects A
- for any $a, b : A$, a **setoid** of morphisms

$$a \to b := (\mathrm{Id}_A(a,b), \mathrm{Id}_{\mathrm{Id}_A(a,b)})$$

- setoid structures are compatible with id, $(\_)^{-1}$ and $\star$, e.g. composition is setoid morphism

$$\mathrm{Id}_A(a,b) \times \mathrm{Id}_A(b,c) \to \mathrm{Id}_A(a,c)$$
$$\mathrm{Id}_{\mathrm{Id}_A(a,b)}(p,q) \times \mathrm{Id}_{\mathrm{Id}_A(b,c)}(r,s) \to \mathrm{Id}_{\mathrm{Id}_A(a,c)}(p \star r, q \star s)$$

- associativity, unitality, inverse witnessed by setoid equalities on hom-sets

Iterate:

Each hom-setoid is again a groupoid up to higher identity.

About ∞-groupoids

Idea of iterating this process is made precise in

Definition (∞-groupoid)

- 0-cells a, b,...
- 1-cells $f, g : a \to b$, $h : b \to c$,...
- 2-cells $\alpha : f \Rightarrow g : a \to b$,...
- ...
- compositions
  - $f \star g$
  - horizontal and vertical composition of 2-cells
  - ...
- coherence laws

Types are ∞-groupoids

Theorem (Lumsdaine, Garner & van den Berg)

The terms belonging to the iterated identity types of any type A form an ∞-groupoid.

A model of MLTT in simplicial sets

Types-as-spaces intuition is made precise by a model of MLTT:

- The category sSET of simplicial sets is Quillen-equivalent to the category TOP of topological spaces.
- There is a model of MLTT in simplicial sets [Voevodsky].
- This model satisfies an additional property: **univalence**
- This suggests adding univalence as an additional axiom (UA) to MLTT.

Remark

Traditional set-theoretic models of MLTT do not satisfy univalence and thus are not models of MLTT + UA.

Propositions as some types

Logic is **embedded** in type theory via Curry–Howard

- proving $P \Rightarrow Q$ amounts to giving a function $P \to Q$
- proving $\forall x : A.P(x)$ amounts to constructing a function

$$\lambda x : A.p(x) : \prod_{x:A} P(x)$$

- proving $\exists x : A.P(x)$ amounts to constructing a pair

$$\big(a, p(a)\big) : \sum_{x:A} P(x)$$

- Which types are “propositions”?
- Some more work is actually required for $\exists$.

Homotopy level of a type

Question:

Given type A, does it have non-trivial paths at arbitrary “height”?

Answer is given by the homotopy level of A:

A type is of h-level $n \in \mathbb{N}$ if it does not have non-trivial paths above height n.

- The homotopy level is **defined internally, i.e. in type theory**.
- There are types of arbitrary high homotopy level.
- There are types without homotopy level (“h-level ∞”).

Homotopy level of a type I

Definition (Propositions & Sets)

A type A is a **proposition** if any two $x, y : A$ are equal:

$$\text{isProp}(A) := \prod_{x\ y:A} \text{Id}(x, y)$$

A type A is a **set** if for any $x, y : A$, the type $\text{Id}(x, y)$ is a proposition:

$$\text{isSet}(A) := \prod_{x\ y:A} \text{isProp}(\text{Id}(x, y))$$

- Propositions are “proof–irrelevant” (“empty or singleton”)
- Points of a set are equal in a unique way, if they are.

Homotopy level of a type II

Definition

A type A is **contractible** if the type

$$\mathsf{isContr}(A) := \sum_{x:A} \prod_{y:A} \mathsf{Id}(x, y)$$

is inhabited.
A type A is **of homotopy level**

- 0 if it is contractible.
- $n + 1$ if for any $x, y : A$, the identity type $\mathsf{Id}_A(x, y)$ is of homotopy level n.

- Propositions are precisely types of h-level 1.
- Sets are precisely types of h-level 2.

Properties of homotopy levels

- For any A, the type $\mathsf{isofhlevel}_n(A)$ is a proposition.
- If A is of h-level n, then it is also of h-level $n+1$.
- $\prod_{x:A} B(x)$ is of h-level n if all $B(x)$ are.
- $\sum_{x:A} B(x)$ is of h-level n if A is and all $B(x)$ are.

Using a universe $\mathcal{U}$ (cf. later):

- $\sum_{A:\mathcal{U}} \mathsf{isofhlevel}_n(A)$ is of h-level $n+1$.

Having defined isomorphism of types (cf. later):

- Isomorphic types have the same homotopy level.

More about sets — Hedberg's theorem

Definition (Decidable equality)

A type A is said to have **decidable equality** if for all $x, y : A$ we have

$$(x \rightsquigarrow y) + \neg(x \rightsquigarrow y).$$

Theorem

- *A type with decidable equality is a set. [**Hedberg**]*
- *A type is a set iff it satisfies Axiom K, i.e.*
  iff every loop over base point a is equal to the constant path $\mathrm{refl}(a)$*.*

- Example: the type $\mathbf{N}$ of natural numbers is a set.

Truncation I

Some operations do not preserve homotopy levels, e.g.:

- $P + Q$ is not necessarily a proposition if P and Q are.
- $\mathrm{isEven}(n)$ is a proposition, but $\sum_{n \in \mathbf{N}} \mathrm{isEven}(n)$ is not.

To obtain operations $P \vee Q$ and $\exists x.P(x)$ as propositions, we need **truncation**:

Propositional truncation

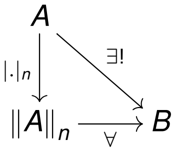
To any type A we associate the type $[A]$ with the following properties:

- $[A]$ is a proposition
- an assumption of type $[A]$ is as good as one of A **when proving a proposition**

Truncation II

1. if A is a type, then $[A]$ is a type
2. if $a : A$ then $|a| : [A]$
3. if $f : A \to P$ and isProp(P), then $|f| : [A] \to P$
4. $|f|(|a|) \equiv f(a)$

More generally, we can define n-**truncation** $\|A\|_n$ as the left adjoint to the inclusion of types of h-level n into types:



B of h-level n

Dependent types as maps to a universe

Types are stratified in universes

We suppose

- having a sequence of **universes** $(\mathcal{U}_n)_{n \in \mathbb{N}}$ (e.g. à la Russell)
- any type A is a point of some universe $A : \mathcal{U}_n$

Implicit universe polymorphism: omit the index n

A dependent type $x : A \vdash B(x)$

is a map $B : A \to \mathcal{U}$.

Univalence : isomorphic types are equal

Universes in MLTT

- Types in MLTT are stratified in **Universes** $\mathcal{U}_n$
- can consider $\mathrm{Id}_{\mathcal{U}}(A, B)$ (polymorphic in universe level n)
- **Univalence** allows to construct identities between A and B

Univalence

- Define type $\mathrm{Iso}(A, B)$ of **Isomorphisms from A to B**
- Univalence Axiom identifies $\mathrm{Iso}(A, B)$ with $\mathrm{Id}(A, B)$
- Can construct $f : \mathrm{Iso}(A, B)$ for suitable A, B

Isomorphism of types

Definition (Isomorphism of types)

A function $f : A \to B$ is an **isomorphism of types** if there are

- $$g : B \to A$$
- $$\eta : \prod_{a:A} \mathsf{Id}\Big(g(f(a)), a\Big) \qquad \epsilon : \prod_{b:B} \mathsf{Id}\Big(f(g(b)), b\Big)$$

together with a coherence condition $\tau : \prod_{x:A} \mathsf{Id}\Big(f(\eta x), \epsilon(fx)\Big)$

- "*f is an isomorphism*", ie. the type of quadruples $(g, \eta, \epsilon, \tau)$ is a proposition, written $\mathsf{isIso}(f)$

Alternative definitions of “isomorphism of types”

There are several equivalent definitions of isomorphism:

- “having a left- and a right-handed inverse”
- “having contractible fibers”, i.e. inverse image of each point is a singleton

Omitting τ yields **logically equivalent** definition:

- triples $(g, \eta, \epsilon) \quad \longleftrightarrow \quad$ quadruples $(g, \eta, \epsilon, \tau)$

Type of isomorphisms as a subtype of the function type

$$\mathsf{Iso}(A, B) := \sum_{f:A \to B} \mathsf{isIso}(f)$$

The Univalence Axiom

Definition (From paths to isomorphisms)

$$\begin{aligned} \mathrm{id\_to\_iso}_{A,B} : \mathrm{Id}(A, B) &\to \mathrm{Iso}(A, B) \\ \mathrm{refl}(A) &\mapsto (x \mapsto x, p) \end{aligned}$$

Univalence Axiom

$$\mathrm{univalence} : \prod_{A\ B:\mathcal{U}} \mathrm{isIso}(\mathrm{id\_to\_iso}_{A,B})$$

In particular, Univalence gives a map backwards:

$$\mathrm{iso\_to\_id}_{A,B} : \mathrm{Iso}(A, B) \to \mathrm{Id}(A, B)$$

Consequences of Univalence

- special case: propositional extensionality

$$(P \leftrightarrow Q) \to \mathrm{Id}(P, Q)$$

- function extensionality:

$$\prod_{x:A} \mathrm{Id}_B(fx, gx) \to \mathrm{Id}_{A \to B}(f, g)$$

  and its dependent variant
- quotient types exist (cf. later)

Outline

Outline

1. Introduction to Univalent Foundations
   - Martin-Löf Type Theory
   - Type theory and its homotopy interpretation
   - A closer look at the identity type
   - Homotopy levels
   - The Univalence Axiom
2. Category Theory in Univalent Foundations
   - Naïve definition
   - Univalence for categories
   - Some results on univalent categories
   - The Rezk completion

Notation

Write $p : x \rightsquigarrow y$ for $p : \mathrm{Id}_A(x, y)$

Categories in Univalent Foundations — Take I

A naïve definition of categories

A **category** $\mathcal{C}$ is given by

- a type $\mathcal{C}_0$ of **objects**
- for any $a, b : \mathcal{C}_0$, a type $\mathcal{C}(a, b)$ of **morphisms**
- operations: identity & composition

$$\mathrm{id}_a : \mathcal{C}(a, a)$$
$$(\circ)_{a,b,c} : \mathcal{C}(b, c) \to \mathcal{C}(a, b) \to \mathcal{C}(a, c)$$

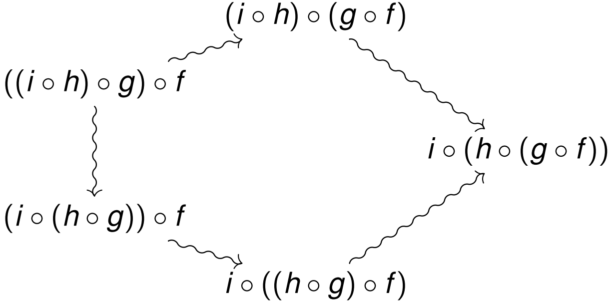
- axioms: unitality & associativity

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

Problem: Would require higher coherence data...

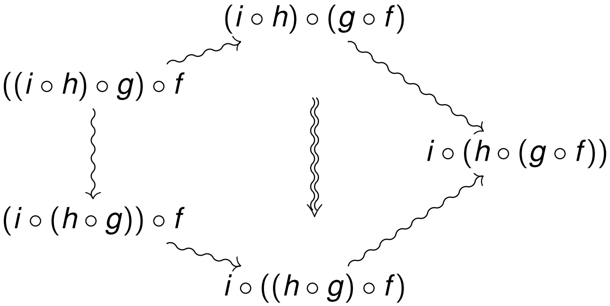
Coherence for associativity – Mac Lane's pentagon

Two ways to associate a composition of **four** morphisms from left to right:



Coherence for associativity – Mac Lane's pentagon

Two ways to associate a composition of **four** morphisms from left to right:



Would need to ask for higher coherence $\rightsquigarrow$, $\rightsquigarrow$ etc

Categories in Univalent Foundations — Take II

A less naïve definition of categories

A **category** $\mathcal{C}$ is given by

- a type $\mathcal{C}_0$ of objects
- for any $a, b : \mathcal{C}_0$, a **set** $\mathcal{C}(a, b)$ of morphisms
- operations: identity & composition
- axioms: unitality & associativity

For this definition of category, the pentagon is automatically coherent.

Isomorphism in a category

Definition (Isomorphism in a category)

A morphism $f : \mathcal{C}(a, b)$ is an **isomorphism** if there are

- $$g : \mathcal{C}(b, a)$$
- $$\eta : g \circ f \rightsquigarrow \mathrm{id}_a \qquad \epsilon : f \circ g \rightsquigarrow \mathrm{id}_b$$

- "*f is an isomorphism*" is a proposition, written $\mathrm{isIso}(f)$
- $$\mathrm{Iso}(a, b) := \sum_{f : \mathcal{C}(a,b)} \mathrm{isIso}(f)$$

From paths to isomorphisms

Definition (From paths to isomorphisms, univalent categories)

For objects $a, b : \mathcal{C}_0$ we define

$$\begin{aligned} \mathrm{id\_to\_iso}_{a,b} : (a \rightsquigarrow b) &\to \mathrm{Iso}(a, b) \\ \mathrm{refl}(a) &\mapsto \mathrm{id}_a \end{aligned}$$

We call the category $\mathcal{C}$ **univalent** if, for any objects $a, b : \mathcal{C}_0$,

$$\mathrm{id\_to\_iso}_{a,b} : (a \rightsquigarrow b) \to \mathrm{Iso}(a, b)$$

is an isomorphism of types.

- In a univalent category, isomorphic objects are equal.
- *"$\mathcal{C}$ is univalent"* is a proposition, written $\mathrm{isUniv}(\mathcal{C})$.

Examples of univalent categories

- SET (follows from the Univalence Axiom)
- categories of algebraic structures (groups, rings,...)
  - made precise by the **Structure Identity Principle** (Coquand, Aczel)
- full subcategories of univalent categories
- functor category $\mathcal{D}^{\mathcal{C}}$, if $\mathcal{D}$ is univalent (see below)

What about categories as objects?

Definition (Functor)

A **functor** F from $\mathcal{C}$ to $\mathcal{D}$ is given by

- a map $F_0 : \mathcal{C}_0 \to \mathcal{D}_0$
- for any $a, a' : \mathcal{C}_0$, a map $F_{a,a'} : \mathcal{C}(a, a') \to \mathcal{D}(Fa, Fa')$
- preserving identity and composition

A functor F is an **isomorphism of categories** if

- F_0 is an isomorphism of types and
- $F_{a,a'}$ is an isomorphism of types (a bijection) for any a, a'.

$$\mathcal{C} \cong \mathcal{D} := \sum_{F:\mathcal{C}\to\mathcal{D}} \mathrm{isIsoOfCats}(F)$$

Natural transformations

Definition (Natural transformation)

Let $F, G : \mathcal{C} \to \mathcal{D}$ be functors. A **natural transformation** $\alpha : F \to G$ is given by

- for any $a : \mathcal{C}_0$ a morphism $\alpha_a : \mathcal{D}(Fa, Ga)$ s.t.
- for any $f : \mathcal{C}(a, b)$, $Gf \circ \alpha_a \rightsquigarrow \alpha_b \circ Ff$

The type of natural transformations $F \to G$ is a **set**.

Definition (Functor category $\mathcal{D}^{\mathcal{C}}$)

- objects: functors from $\mathcal{C}$ to $\mathcal{D}$
- morphisms from F to G: natural transformations

A natural transformation α is an isomorphism iff each α_a is.

Equivalence of categories

Definition (Left Adjoint, Equivalence of Categories)

A functor $F : \mathcal{C} \to \mathcal{D}$ is a **left adjoint** if there are

- $G : \mathcal{D} \to \mathcal{C}$
- $\eta : 1_{\mathcal{C}} \to GF$
- $\epsilon : FG \to 1_{\mathcal{D}}$
- + higher coherence data.

A left adjoint F is an **equivalence of categories** if η and ϵ are isomorphisms.

"F is an equivalence" is a proposition.

$$\mathcal{C} \simeq \mathcal{D} := \sum_{F:\mathcal{C}\to\mathcal{D}} \mathrm{isEquivOfCats}(F)$$

1 kind of sameness for univalent categories

Equality	$\mathcal{C} \rightsquigarrow \mathcal{D}$
Isomorphism	$\mathcal{C} \cong \mathcal{D}$
Equivalence	$\mathcal{C} \simeq \mathcal{D}$

Theorem

*For **univalent** categories $\mathcal{C}$ and $\mathcal{D}$, these three are equivalent as types.*

In particular, we can substitute a univalent category with an equivalent one.

Rezk completion

- *"Being univalent"* is a proposition

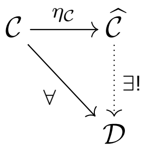
⇝ Inclusion from univalent categories to categories

Theorem

The inclusion of univalent categories into categories has a left adjoint (in bicategorical sense),

$$\mathcal{C} \mapsto \widehat{\mathcal{C}} \qquad \textit{\textbf{Rezk completion}} \textit{ of } \mathcal{C}$$

That is, any functor $F : \mathcal{C} \to \mathcal{D}$ with $\mathcal{D}$ univalent factors uniquely via $\eta_{\mathcal{C}} : \mathcal{C} \to \widehat{\mathcal{C}}$:



η is unit of adjunction

Special case of Rezk completion: Quotienting

Specialise: category $\rightsquigarrow$ groupoid $\rightsquigarrow$ equivalence relation

Theorem

Univalent Foundations admits quotients

- More direct construction of set-level quotients by Voevodsky: “type of equivalence classes”

Formalization and reference

Formalization in Coq

- Rezk completion formalized
- approx. 4000 lines of code
- based on Voevodsky's library "*Foundations*"

⇝ `github.com/benediktahrens/rezk_completion`

References

- preprint with same title `arxiv.org/abs/1303.0584`
- C. Rezk, *A model for the homotopy theory of homotopy theory*, 2001