A unified approach to Univalent Foundations and Homotopical Algebra

Nicola Gambino

University of Leeds

Vienna, TLCA-RTA 2014

References (I)

V. Voevodsky, **Univalent Foundations**, 2006-present

- Univalence axiom
- New approach to mathematics in type theory

D. Quillen, **Homotopical algebra**, 1967

- Notion of a model category
- Axiomatic development of homotopy theory

References (II)

The Univalent Foundations Program
Homotopy Type Theory
2013

M. Hovey
Model categories
AMS, 1998

Aim of the talk

- Step 1. Analysis of the dependent type theories considered in Univalent Foundations
- Step 2. New setting for the development of homotopical algebra
- Step 3. Homotopy-theoretic ideas feed back into dependent type theories

Step 2: work by Awodey, van den Berg and Garner, Joyal, Lumsdaine and Warren, Shulman, Voevodsky, ...

Plan of the talk

1. Review of dependent type theories

2. Homotopy-theoretic aspects of dependent type theories

3. Homotopy-initial natural numbers

1. Review of dependent type theories

Dependent type theories (I)

Key idea. We have types and their elements

$$A : \mathsf{type} \qquad a : A$$

and also **dependent types** and their elements

$$x : A \vdash B(x) : \mathsf{type} \qquad x : A \vdash b(x) : B(x)$$

Examples.

- $0 : \mathsf{Nat}$
- $[3, 14, 2] : \mathsf{List}(\mathsf{Nat})$
- $n : \mathsf{Nat} \vdash \mathsf{List}_n(\mathsf{Bool}) : \mathsf{type}$
- $x : A \vdash \mathsf{refl}(x) : \mathsf{Id}_A(x, x)$.

Dependent type theories (II)

In general, dependent types and their elements have the form

$$\Gamma \vdash A : \mathsf{type} \qquad \Gamma \vdash a : A$$

where Γ is a **context**, i.e. a sequence of variable declarations

$$\Big(x_0 : A_0\ , x_1 : A_1(x_0)\ ,\ \ldots\ ,\ x_n : A_n(x_0, \ldots, x_{n-1})\Big)$$

Examples.

- $n : \mathsf{Nat}\ , \ell : \mathsf{List}_n(\mathsf{Nat}) \vdash \mathsf{reverse}(\ell) : \mathsf{List}_n(\mathsf{Nat})$
- $x, y, z : A, u : \mathsf{Id}_A(x, y), v : \mathsf{Id}_A(y, z) \vdash \mathsf{trans}(u, v) : \mathsf{Id}_A(x, z)$.

We write $(\,)$ for the empty context.

Dependent type theories (III)

A dependent type theory has:

(1) Structural rules

(2) Rules for primitive types, e.g.

$$\mathsf{Empty}\,, \quad \mathsf{Unit}\,, \quad \mathsf{Bool}\,, \quad \mathsf{Nat}$$

(3) Rules for forming new types from old, e.g.

$$A \times B\,, \quad A \to B\,, \quad A + B\,,$$

$$\mathsf{Id}_A(a, b)\,, \quad (\Sigma x : A)B\,, \quad (\Pi x : A)B\,.$$

These rules have an abstract description (cf. typed λ-calculus).

The syntactic category

The **syntactic category** of a dependent type theory T has:

- Objects: contexts Γ, Δ, ...
- Morphisms: terms-in-context, e.g.

$$\Gamma \to (x\colon A) \qquad \Longleftrightarrow \quad (a)\,, \text{ where } \Gamma \vdash a\colon A$$

$$\Gamma \to (x\colon A, y\colon B(x)) \quad \Longleftrightarrow \quad (a,b)\,, \text{ where } \begin{cases} \Gamma \vdash a\colon A \\ \Gamma \vdash b\colon B(a) \end{cases}$$

Examples.

- A morphism $(x\colon A) \to (y\colon B)$ is a family $x\colon A \vdash f(x)\colon B$.
- A morphism $(\,) \to (x\colon A)$ is an element $a\colon A$.

Display maps

Definition. A **display map** is a morphism of the form

$$p_A \colon (\Gamma, x \colon A) \to (\Gamma)\,,$$

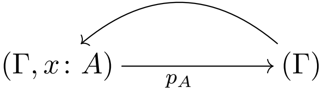
given by the list of the variables in Γ, where $\Gamma \vdash A \colon \mathsf{type}$.

Examples.

- $(x) \colon (x \colon A, y \colon B(x)) \to (x \colon A)$
- $(x, y) \colon (x \colon A, y \colon B(x), z \colon C(x, y)) \to (x \colon A, y \colon B(x))$

Dependent elements as sections

Remark. For a dependent type $\Gamma \vdash A\colon \mathsf{type}$, a section of p_A



is the same thing as a dependent element

$$\Gamma \vdash a\colon A$$

The section is given by the sequence $(\dots, a)$.

Note. For $\Gamma = (\,)$, we have just $a\colon A$, as before.

Substitution as pullback

For every

- display map $p_A \colon (\Gamma, x \colon A) \to \Gamma$
- context morphism $\sigma \colon \Delta \to \Gamma$

we have a pullback diagram

$$\begin{array}{ccc} (\Delta, x \colon A[\sigma]) & \longrightarrow & (\Gamma, x \colon A) \\ \downarrow & & \downarrow{\scriptstyle p_A} \\ \Delta & \underset{\sigma}{\longrightarrow} & \Gamma \end{array}$$

Example.

$$\begin{array}{ccc} (x \colon A, z \colon C(f(x))) & \longrightarrow & (y \colon B, z \colon C(y)) \\ \downarrow & & \downarrow{\scriptstyle p_B} \\ (x \colon A) & \underset{f(x)}{\longrightarrow} & (y \colon B) \end{array}$$

Basic axiomatic setting

Definition. A **category with projections** consists of

- a category $\mathbb{C}$ with a terminal object 1
- a class of maps $\mathcal{P}$, called **projections**

such that:

- $\mathcal{P}$ contains isomorphisms and is closed under composition,
- For every morphism $p\colon E \to A$ in $\mathcal{P}$ and $f\colon B \to A$, there is a pullback

  $$\begin{array}{ccc} F & \longrightarrow & E \\ {\scriptstyle q}\downarrow & & \downarrow{\scriptstyle p} \\ B & \xrightarrow[f]{} & A \end{array}$$

  with $q\colon F \to B$ in $\mathcal{P}$.
- Every map $A \to 1$ is in $\mathcal{P}$.

Examples of categories with projections

1. The syntactic category of a dependent type theory, with

$\mathcal{P} =$ closure of display maps under composition and isomorphisms

2. The category of Kan complexes (= 'spaces'), with

$$\mathcal{P} = \text{ Kan fibrations (= 'good projections')}$$

Σ-types and Π-types

- Σ-types are types of pairs:

$$\frac{\Gamma, x\colon A \vdash B(x)\colon \mathsf{type}}{\Gamma \vdash (\Sigma x\colon A)B(x)} \qquad \frac{\Gamma \vdash a\colon A \quad \Gamma \vdash b\colon B(a)\colon \mathsf{type}}{\Gamma \vdash \mathsf{pair}(a,b)\colon (\Sigma x\colon A)B(x)}$$

- Π-types are types of sections:

$$\frac{\Gamma, x\colon A \vdash B(x)\colon \mathsf{type}}{\Gamma \vdash (\Pi x\colon A)B(x)} \qquad \frac{\Gamma, x\colon A \vdash b(x)\colon B(x)\colon \mathsf{type}}{\Gamma \vdash (\lambda x\colon A)b(x)\colon (\Pi x\colon A)B(x)}$$

Strong versions of their rules correspond to existence of adjunctions to the pullback functor along $(\Gamma, x\colon A) \to (\Gamma)$.

For identity types, ideas of homotopical algebra are necessary.

2. Homotopical aspects of dependent type theories

Analogy

Type theory	**Homotopy theory**
$A\colon \mathsf{type}$	A space
$a\colon A$	point $a \in A$
$x\colon A \vdash B(x)\colon \mathsf{type}$	fibration $p\colon B \to A$
$x\colon A \vdash b(x)\colon B(x)$	section of $p\colon B \to A$
$x, y\colon A \vdash \mathsf{Id}_A(x, y)$	path space $A^I \to A \times A$

Id-types (I)

Formation rule.

$$\frac{A : \mathsf{type} \quad a : A \quad b : A}{\mathsf{Id}_A(a,b) : \mathsf{type}}$$

Introduction rule.

$$\frac{a : A}{\mathsf{refl}(a) : \mathsf{Id}_A(a,a)}$$

Idea. $p \in \mathsf{Id}_A(a,b) \Longleftrightarrow p$ is a proof that a equals b.

Id-types (II)

Elimination rule.

$$\frac{x, y : A, u : \mathsf{Id}_A(x, y) \vdash E(x, y, u) : \mathsf{type} \quad x : A \vdash d(x) : E(x, x, \mathsf{refl}(x))}{x, y : A, u : \mathsf{Id}_A(x, y) \vdash \mathsf{J}(x, y, y, d) : E(x, y, u)}$$

Computation rule.

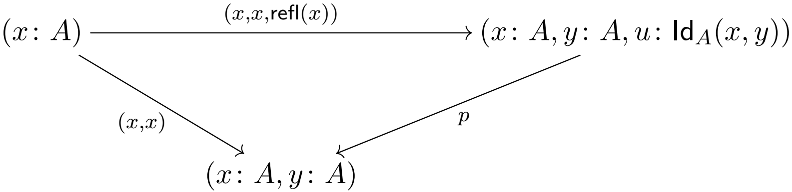
$$\frac{x, y : A, u : \mathsf{Id}_A(x, y) \vdash E(x, y, u) : \mathsf{type} \quad x : A \vdash d(x) : E(x, \mathsf{refl}(x))}{x : A \vdash \mathsf{J}(x, x, \mathsf{refl}(x), d) = d(x) : E(x, x, \mathsf{refl}(x))}$$

Identity types in the syntactic category (I)

Formation rule. A display map

$$p\colon (x\colon A, y\colon A, u\colon \mathsf{Id}_A(x,y)) \to (x\colon A, y\colon A)$$

Introduction rule. A factorisation



Identity types in the syntactic category (II)

Elimination and computation rule. A diagonal filler for diagrams

$$\begin{array}{ccc}
(x\colon A) & \xrightarrow{(x,x,\mathsf{refl}(x),d(x))} & (x,y\colon A, u\colon \mathsf{Id}(x,y), z\colon E(x,y,u)) \\
{\scriptstyle (x,x,\mathsf{refl}(x))}\big\downarrow & & \big\downarrow{\scriptstyle p_E} \\
(x,y\colon A, u\colon \mathsf{Id}(x,y)) & \xrightarrow[=]{} & (x,y\colon A, u\colon \mathsf{Id}(x,y))
\end{array}$$

Anodyne maps

Let $(\mathbb{C}, \mathcal{P})$ be a category with a class of projections.

Definition. We say that $i \colon X \to Y$ is an **anodyne map** if it has the left lifting property with respect to every projection, i.e. every diagram

$$\begin{array}{ccc} X & \longrightarrow & E \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle p} \\ Y & \longrightarrow & B \end{array}$$

with $p \colon E \to B$ in $\mathcal{P}$, has a diagonal filler.

Example. The morphism

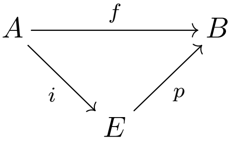
$$(x, x, \mathsf{refl}(x)) \colon (x \colon A) \to (x, y \colon A, u \colon \mathsf{Id}(x, y))$$

is anodyne.

Homotopical categories with projections

Definition. We say that a category with a class of projections $(\mathbb{C}, \mathcal{P})$ is **homotopical** if

- every map factors as an anodyne map followed by a projection:

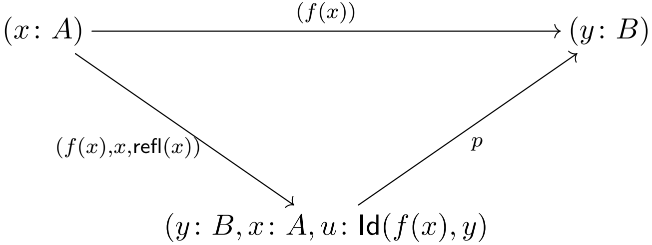


- the pullback of an anodyne map along a projection is anodyne.

Examples

1. The syntactic category of a dependent type theory with identity types is homotopical (Gambino and Garner).

For example:



2. The category of Kan complexes is homotopical.

Relation to homotopical algebra

Homotopical categories with projections are a weakening of many structures considered in homotopical algebra.

In particular:

- Minimal assumptions on $\mathbb{C}$
  (no completeness and cocompleteness)
- Just path objects, not cylinder objects
  (assumed in a model category)
- No functoriality of the factorisation
  (often assumed in the theory of model categories)

Note. Strengthenings by adding

- Π-types and function extensionality
- higher inductive types (Lumsdaine and Shulman)
- Univalence axiom

The Univalence axiom

Fix a type universe $\mathsf{U} \colon \mathsf{type}$.

$$A \colon \mathsf{U} \iff A \text{ is a `small type'}$$

For each $A, B \colon \mathsf{U}$, we have:

- $\mathsf{Id}_{\mathsf{U}}(A, B)$
- the type $\mathsf{Equiv}(A, B)$ of equivalences $f \colon A \to B$.
- a function
$$\mathsf{Id}_{\mathsf{U}}(A, B) \to \mathsf{Equiv}(A, B)$$

Univalence Axiom. The function $\mathsf{Id}_{\mathsf{U}}(A, B) \to \mathsf{Equiv}(A, B)$ is an equivalence.

3. Homotopy-initial natural numbers

The type of natural numbers (I)

Formation rule.

$$\mathsf{Nat} : \mathsf{type}$$

Introduction rules.

$$0 : \mathsf{Nat} \qquad\qquad \frac{n : \mathsf{Nat}}{\mathsf{succ}(n) : \mathsf{Nat}}$$

The type of natural numbers (II)

Elimination rule.

$$\frac{x : \mathsf{Nat} \vdash E(x) : \mathsf{type} \quad d : E(0) \quad x : \mathsf{Nat}, y : E(x) \vdash e(x,y) : E(\mathsf{succ}(x))}{x : \mathsf{Nat} \vdash \mathsf{natrec}(x,d,e) : E(x)}$$

Computation rules.

$$\frac{x : \mathsf{Nat} \vdash E(x) : \mathsf{type} \quad d : E(0) \quad x : \mathsf{Nat}, y : E(x) \vdash e(x,y) : E(\mathsf{succ}(x))}{\mathsf{natrec}(0,d,e) = d : E(0)}$$

$$\frac{x : \mathsf{Nat} \vdash E(x) : \mathsf{type} \quad d : E(0) \quad x : \mathsf{Nat}, y : E(x) \vdash e(x,y) : E(\mathsf{succ}(x))}{x : \mathsf{Nat} \vdash \mathsf{natrec}(\mathsf{succ}(x),d,e) = e(x, \mathsf{natrec}(u,d,e)) : E(\mathsf{succ}(x))}$$

Homotopy-invariance

Note. If $f\colon A \to \mathsf{Nat}$ is an equivalence, then A will satisfy

- the introduction rules for Nat,
- the elimination rules for Nat,
- the computation rules, modified by having propositional equalities in the conclusion.

We call such a type **inductive**.

Successor algebras

Definition.

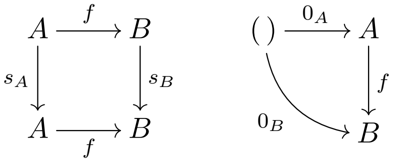
- A **successor algebra** is a tuple $(A, s_A, 0_A)$, where A is a type, $s_A \colon A \to A$ and $0_A \colon A$.
- A **morphism** of successor algebras

$$(f, \bar{f}_s, \bar{f}_0) \colon (A, s_A, 0_A) \to (B, s_B, 0_B)$$

is a function $f \colon A \to B$ together with

$$\bar{f}_s \colon \mathsf{Id}(s_B \circ f, f \circ s_A)\,, \quad \bar{f}_0 \colon \mathsf{Id}(f(0_A), 0_B)\,.$$

These are proofs that the diagrams commute:



Homotopy-initial successor algebras

Note. For successor algebras A and B, we can form the type

$$\mathsf{SuccAlg}[A, B] =_{\text{def}} (\Sigma f \colon A \to B)\big(\mathsf{Id}(s_B \circ f, f \circ s_A) \times \mathsf{Id}(f(0_A), 0_B)\big)$$

of successor algebra morphisms from A to B.

Definition. A successor algebra A is **homotopy-initial** if for any successor algebra B the type $\mathsf{SuccAlg}[A, B]$ is contractible, i.e. it has a unique element up to propositional equality.

Note.

- What is required is uniqueness of tuples.
- Homotopy-theoretic variant of initiality.

A characterisation

Theorem (Awodey, Gambino, Sojakova) For a successor algebra A, the following are equivalent:

1. A is equivalent to Nat
2. A is inductive
3. A is homotopy-initial.

Note.

- Special case of general result on W-types.
- Result can be internalized.

Conclusion

The interplay between dependent type theory and homotopy theory:

- suggests a new, refined axiomatic setting for developing homotopical algebra, yet to be fully explored.
- provides new, topologically-inspired, intuition for working with dependent type theories.