An introduction to Homotopy Type Theory

Nicola Gambino

University of Palermo

Leicester, March 15th, 2013

Outline of the talk

- **Part I:** Type theory
- **Part II:** Homotopy type theory
- **Part III:** Voevodsky's Univalent Foundations

Part I:

Type theory

Motivation for type theory

Problem

- How can we write correct programs?

Standard approach

- Write the program
- Verify its correctness via semantics

Type-theoretic approach

- Write a correct-by-construction program

Verification via type-checking

Idea

- Use types to classify syntactic expressions and write specifications
- Use type-checking to prevent mistakes

Examples

- $\texttt{3} : \texttt{Nat}$
- $\texttt{cons}([3,4],[6,2]) : \texttt{List}(\texttt{Nat})$
- $[1,7,15,34] : \texttt{SortedList}(\texttt{Nat})$

× $[3,1,4,8] : \texttt{SortedList}(\texttt{Nat})$

Type theories

Goal

- Expressive type system
- Decidability of type checking

Idea

- Powerful mechanism for defining recursive data types, e.g.

$$\texttt{Nat}\,,\quad \texttt{List}(A)\,,\quad \texttt{Tree}(A)\,,\quad \ldots$$

- Dependent types, e.g.

$$\texttt{List}_n(A)\,,\quad \texttt{is\_sorted}(\ell)\,.$$

Martin-Löf type theories

Some forms of type:

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

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

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

We will only need the rules for identity types.

Identity types

Formation rule

$$\frac{A : \texttt{type} \qquad a : A \qquad b : A}{\texttt{Id}_A(a, b) : \texttt{type}}$$

For example, if $a : A$ then $\texttt{Id}_A(a, a) : \texttt{type}$

Introduction rule

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

Elimination rule

$$\frac{\begin{array}{l} p : \mathtt{Id}_A(a,b) \\ x\colon A\,,y\colon A\,,u\colon \mathtt{Id}_A(x,y) \vdash C(x,y,u)\colon \mathtt{type} \\ x\colon A \vdash c(x)\colon C(x,x,\mathtt{refl}(x)) \end{array}}{\mathtt{J}(a,b,p,c)\colon C(a,b,p)}$$

Idea

$$\frac{a = b \qquad \begin{array}{c}[x\colon A] \\ \vdots \\ C(x,x)\end{array}}{C(a,b)}$$

Similar to Lawvere's treatment of equality in categorical logic.

Computation rule

$$\frac{\begin{array}{l} a\colon A \\ x\colon A\,, y\colon A\,, u\colon \mathtt{Id}_A(x,y) \vdash C(x,y,u)\colon \mathtt{type} \\ x\colon A \vdash c(x)\colon C(x,x,\mathtt{refl}(x)) \end{array}}{\mathtt{J}(a,a,\mathtt{refl}(a),c) = c(a)\colon C(a,a,\mathtt{refl}(a))}$$

Idea

$$\dfrac{\dfrac{a\colon A}{a=a} \qquad \begin{array}{c}[x\colon A]\\ \vdots \\ C(x,x)\end{array}}{C(a,a)} \qquad\longrightarrow\qquad \begin{array}{c} a\colon A \\ \vdots \\ \vdots \\ C(a,a)\end{array}$$

Part II:

Homotopy type theory

Semantics of type theories

Problems

- Set-theoretical semantics validates also:

$$\frac{p : \mathtt{Id}_A(a, b)}{a = b : A} \qquad \frac{p : \mathtt{Id}_A(a, b)}{p = \mathtt{refl}(a) : \mathtt{Id}_A(a, b)}$$

  which makes type-checking undecidable.

- It is difficult to reason within type theories without good models.

Dictionary

Type theory	**Homotopy theory**
$A : \texttt{type}$	A space
$a : A$	$a \in A$
$x : A \vdash B(x) : \texttt{type}$	$B \to A$ fibration
$x : A, y : A \vdash \texttt{Id}_A(x, y)$	$A^{[0,1]} \to A \times A$
Inductive types	Homotopy-initial algebras

Some results

Theorem (Awodey and Warren)**.** The rules for identity types admit an interpretation in every category equipped with a weak factorisation system.

Theorem (Gambino and Garner)**.** The deduction rules for identity types determine a weak factorisation system on the syntactic category of a Martin-Löf type theory.

Theorem (Garner and van den Berg, Lumsdaine)**.** Every type of Martin-Löf type theory determines a weak ω-groupoid.

Theorem (Voevodsky)**.** Martin-Löf type theories have models in the category of simplicial sets that do not validate the reflection rule.

Part III:

Voevodsky's Univalent Foundations

“While working on the completion of the Bloch-Kato conjecture I have thought a lot about what to do next.

Eventually I became convinced that the most interesting and important directions in current mathematics are the ones related to the transition into a new era which will be characterized by the widespread use of automated tools for proof construction and verification.”

V. Voevodsky (2010)

Univalent Foundations

Overview

- Use the dictionary of Homotopy Type Theory to introduce topological notions in type theory
- Exploit these notions to develop mathematics in type theory
- Formalise the development in Coq/Agda.

Contractibility

Definition. A type X is **contractible** if the type

$$\mathtt{iscontr}(X) =_{\text{def}} (\Sigma x_0 : X)(\Pi x : X)\mathtt{Id}_X(x_0, x)$$

is inhabited.

Idea

- Existence and uniqueness

Note

- X contractible $\Leftrightarrow X \simeq \mathtt{Unit}$
- X contractible $\Rightarrow \mathtt{Id}_X(x, y)$ contractible for all $x, y : X$

The hierarchy of h-levels

Definition.

- A type X has level 0 if it is contractible.
- A type X has level $n+1$ if for all $x, y : X$, the type $\mathtt{Id}_X(x, y)$ has level n

Terminology.

- Types of h-level 1 are called h-propositions (logic)
- Types of h-level 2 are called h-sets (algebra)
- Types of h-level 3 are called h-groupoids (category theory)

Note. There is a $+2$ shift w.r.t. homotopy types.

Further developments

- Voevodsky's Univalence Axiom
- Calculations of fundamental groups of spheres
- Development of category theory
- ...