An Invitation to Homotopy Type Theory

Tingxiang Zou

Type Theory

• Formal Systems:
Church's simply typed lambda calculus (1940);
Martin Löf's dependent type theory (1971-1984)

• Origin: Russell's theory of types
The world is organised by **types**, each entity/**term** is assigned to certain type. (e.g. $n : \mathbb{N}; \quad f : \mathbb{N} \to \mathbb{N}$)
We have some basic types and terms to start with, and build new ones from rules. (e.g. $A \times B; \quad f(n) : \mathbb{N}$)

• Four basic kinds of judgements:

A type; $\qquad a : A$;

$A = B$; $\qquad a = b : A$.

• Each judgement is warranted by a suitable (possibly empty) **context**, which is a variable declaration: $x_1 : A_1, \cdots, x_n : A_n$.

• $x_1 : A_1, \cdots, x_n : A_n \vdash a : A$

Martin Löf's Type Theory

- Types are **dependent**: $x : A \ \vdash \ B(x)$ type

Contexts: $x_1 : A_1, x_2 : A_2(x_1), \cdots, x_n : A_n(x_1, \cdots, x_{n-1})$, if for each i, $x_1 : A_1, \cdots, x_i : A_i(x_1, \cdots, x_{i-1}) \ \vdash \ A_{i+1}$ type.

- Dependent Product (Π-type):

Type Formation: $$\frac{\vdash \ A \text{ type}; \quad x : A \ \vdash \ B(x) \text{ type}}{\vdash \Pi_{x:A} B(x) \text{ type}};$$

Term Introduction: $$\frac{x : A \ \vdash \ b(x) : B(x)}{\vdash \lambda x.b(x) : \Pi_{x:A} B(x)};$$

Term Elimination: $$\frac{\vdash \ f : \Pi_{x:A} B(x); \quad \vdash a : A}{\vdash Ap(f, a) : B(a)};$$

Computation Rule: $$\frac{x : A \ \vdash \ b(x) : B(x); \quad \vdash a : A}{\vdash Ap(\lambda x.b(x), a) = b(a) : B(a)};$$

How to read $a : A$?

• A is a set, a is an element of A.
Type constructions corresponds to set constructions.

• A is a proposition, a is a proof/construction/witness of proposition A. $a : A$ implies A is true.
Type constructions corresponds to the construction of formulas.
$A \times B \rightsquigarrow A \wedge B$; $\Pi_{x:A} B(x) \rightsquigarrow \forall x^A B(x)$; $\Sigma_{x:A} B(x) \rightsquigarrow \exists x^A B(x)$.
A proposition is nothing but a collection of proofs, term introduction rules states what are accepted as proofs.
Howard: The typed lambda calculus corresponded to intuitionistic natural deduction. Martin Löf extends this correspondence to predicate logic.

• A is a problem, a is a program/algorithm solving this problem.
Foundation of a programming language (Coq, Agda).

• Curry-Howard correspondence:
Proofs-as-programs; Propositions-as-types

Various Faces of Type Theory

- Foundation of Mathematics
- Intuitionistic logic, Constructive mathematics
- Programming languages
- A is a space and a is a point of A.

The motivation is from interpreting a special kind of types, the **identity types** in Martin Löf's type theory.

- New area of research: Homotopy Type Theory.

Identity types

• Definitional equality: $a = b : A$; (judgement)
• Propositional equality: (type, proposition)

Type Formation: $$\frac{\vdash\ A \text{ type}}{x : A,\ \ y : A \vdash Id_A(x, y) \text{ type}};$$

Term Introduction: $$x : A \ \vdash\ r_A(x) : Id_A(x, x);$$

• Definitional equality implies propositional ones.
• **Extensional identity types**: propositional equality implies definitional one, i.e., $p : Id_A(a, b)\ \vdash\ a = b : A$.
Under types-as-sets view, two elements are equal, if they are extensionally equal.
• **Intensional identity types**: $p : Id_A(a, b)\ \vdash\ a = b : A$ is not valid.
• Type theory with intentional identity types preserves nice computational property (type checking is decidable).

Homotopy Theory

- **Path:** A path in space X is a continuous function $f : [0,1] \to X$.
- **Homotopy:** A homotopy between two continuous functions $f, g : X \to Y$ is a continuous function $H : X \times [0,1] \to Y$ such that for all $x \in X$, $H(x,0) = f(x)$, $H(x,1) = g(x)$.
- **Path homotopy:** Given two paths f, g from x to y in X, a path homotopy is a homotopy H from f to g, such that $H(0,t) = x$ and $H(1,t) = y$ for all t.
- **Homotopy equivalence:** A continuous function $f : X \to Y$ is a homotopy equivalence if there is a continuous function $g : Y \to X$ such that both $f \circ g$ and $g \circ f$ are homotopic to identity functions. We call X, Y are homotopy equivalent or of the same **homotopy type**.
- **Homotopy group:** For a space X with a fixed base point b, we define $\pi_n(X,b)$ to be the group of homotopy classes of maps $g : [0,1]^n \to X$ from the n-cube to X that take the boundary of the n-cube to the base point b.

Homotopy theory and Type theory

- Identity types are path spaces.

$p : Id_A(a, b)$ is a path from a to b, and if $p, q : Id_A(a, b)$, then $h : Id_{Id_A(a,b)}(p, q)$ is a path homotopy from p to q.

Not necessary $p : Id_A(a, b) \ \vdash \ a = b : A$.

Transport

Suppose P is a dependent type over A and $p : Id_A(x, y)$. Then there is a function $p_* : P(x) \to P(y)$.

Path Lifting Property

Suppose we have $u : P(x)$ for some $x : A$, then for any $p : Id_A(x, y)$, we have a term $\mathrm{lift}(u, p) : Id_{\Sigma_{x:A} P(x)}((x, u), (y, p_*(u)))$, such that $p_1(\mathrm{lift}(u, p)) = p$.

- Dependent types are fibrations;
  Terms are continuous sections of fibrations;

- Martin Löf's intentional type theory can be seen as logic for homotopy theory. (e.g. Homotopy, Contactable)

Univalence Axiom

•**Universe**: We have a hierarchy of universes

$$U_0 : U_1 : U_2 : \cdots ,$$

each universe U_i is a term of the next universe U_{i+1}.
Universes are cumulative: if $A : U_i$, then $A : U_{i+1}$.
Judgement A type is $A : U_i$ for some i, we write $A : U$.
• We can talk about spaces now, $Id_U(A, B)$
• For any type A, B, we have the type $(A \simeq B)$ of equivalences from A to B (e.g. functions $f : A \to B$ which has both left and right homotopical inverse).
• **Univalence Axiom:**(Vladimir Voevodsky)
For any $A, B : U$, $\quad Id_U(A, B) \simeq (A \simeq B)$.
• Identity is equivalent to equivalence.
• In particular there is a term $ua : (A \simeq B) \to Id_U(A, B)$, which witnesses the proposition:
if A, B are equivalent, then they are equal.

h-levels

• **h-levels:** A type A is of h-level 0 if it is contractible.
A type A is of h-level $n+1$ if, for all terms a and b of type A, the type $Id_A(a,b)$ is of h-level n.
• **Homotopy n-types:** We say that a space X for which all $\pi_k(X,a)$ with $k > n$ are trivial is a homotopy n-type.

h-level	corresponding space up to equivalence
0	the contractible space 1
1	the space 1 and the empty space 0
2	sets
3	the homotopy 1-types (groupoids)
$\dots$	$\dots$
n	the homotopy $(n-2)$-types
$\dots$	$\dots$

• **Univalnet Perspective:**
logic: homotopy types of level 1;
Set-theoretic mathematics: homotopy types of level 2;
Categorical-theoretic mathematics: homotopy types of level 3...

Univalent Foundation program

• Features:(Voevodsky)
Can be used both for constructive and non-constructive mathematics;
Naturally included axiomatizing of categorical thinking;
Can be conveniently formalised using dependent type systems;
The whole foundation is based on a direct formalization/axiomatizing of the world of homotopy types instead of the world of sets.

• Do mathematics in this type theory with the proof assistant Coq!

• A lot of homotopy theory can be done in Coq, e.g. the proof $\pi_n(S^n) \simeq \mathbb{Z}$. People are trying on some of the other modern mathematics under this approach.

• "One of Voevodsky's goals is that in a not too distant future, mathematicians will be able to verify the correctness of their own papers by working within the system of univalent foundations formalised in a proof assistant, and doing so will become natural even for pure mathematicians."

Conclusion

- Course: (1st April-22nd May, 2015)
  Benno van den Berg: Homotopy Type Theory

Thank You !