Homotopical Type Theory

Eaton Liu

Magdalene College

Type theory

Type theory: a formal system for developing mathematics
Primitives:

- *types*: $\vdash A$, e.g. $\mathbb{N}$, $\mathbb{R}$, S^1;
- *terms*: $\vdash a : A$, e.g. $1 : \mathbb{N}$, $1 : \mathbb{R}$, $\mathrm{base} : S^1$;
- *family of types*: $x : A \vdash B(x)$, $x : A, y : B(x) \vdash C(x, y)$, e.g. $n : \mathbb{N} \vdash \mathbb{R}^n$, $n : \mathbb{N} \vdash \mathrm{isPrime}(n)$;
- *family of terms*: $x : A \vdash b_x : B(x)$, $x : A, y : B(x) \vdash c_{x,y} : C(x, y)$, e.g. $n : \mathbb{N} \vdash (0, \ldots, 0) : \mathbb{R}^n$.

Constructions:

- *product types:* $A \times B$, *sum types:* $A + B$, *function types:* $A \to B$;
- *dependent product types:* $\prod_{x:A} B(x)$, *dependent sum types:* $\sum_{x:A} B(x)$;
- *identity types:* $x, y : A \vdash x =_A y$;

equipped with rules to construct their terms (introduction), deconstruct their terms (elimination), and compute with them (computation).

Propositions as types

How to prove the proposition $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$?

Proof.

Assume $p \to (q \to r)$, then assume $p \to q$, and finally assume p. We get q by applying $p \to q$ to p, and we conclude r by applying $p \to (q \to r)$ to p then q. □

Same as constructing a function $(r^q)^p \to (r^p)^{(q^p)}$!
$\lambda f.\lambda g.\lambda x.\ (f\ \ x)\ (g\ \ x) : (p \to (q \to r)) \to (p \to q) \to (p \to r)$.

Type theory	Logic
Type A	Proposition A
Term $a : A$	Proof of proposition A
$A \times B$	Conjunction $A \wedge B$
$A + B$	Disjunction $A \vee B$
$A \to B$	Implication $A \to B$
$\prod_{x:A} B(x)$	Universal quantification $\forall x \in A, B(x)$
$\sum_{x:A} B(x)$	Existential quantification $\exists x \in A, B(x)$

Homotopical interpretation

Types as spaces, terms as points.

Type theory	Homotopy theory
$A \to B$	Space of continuous maps $A \to B$
$A \times B$	Product space $A \times B$
$A + B$	Disjoint union $A + B$
Type family $x : A \vdash B(x)$	Fibration over A
Term family $x : A \vdash b_x : B(x)$	Section of the fibration
$\prod_{x:A} B(x)$	Space of sections
$\sum_{x:A} B(x)$	Total space of fibration

Identity types

Rules

Formation rule: $x, y : A \vdash x =_A y$;
Introduction rule: $x : A \vdash \mathsf{refl}_x : x =_A x$;
Elimination rule (path induction): To give a term family $x, y : A, p : x =_A y \vdash b(x, y, p) : B(x, y, p)$, suffices to assume $y \equiv x$ and $p \equiv \mathsf{refl}_x$ (i.e. giving $b' : B(x, x, \mathsf{refl}_x)$).
Computation rule: The function constructed by the elimination rule satisfies $b(x, x, \mathsf{refl}_x) \equiv b'$.

Logical interpretation: the *proposition* that x and y are equal;
Homotopical interpretation: the *space of paths* from x to y in A.
Terms $p : x =_A y$ are *paths* from x to y.
$\mathsf{refl}_x : x =_A x$ is the *constant path* at x.
Can form iterated identity types $p =_{x =_A y} q$ (homotopies between paths), $r =_{p =_{x =_A y} q} s$ (homotopies between homotopies), etc.

Path induction

Concatenation/Transitivity

Given $x, y, z : A$, $p : x =_A y$, $q : y =_A z$, there is a path $p \cdot q : x =_A z$.

Proof/Construction.

By path induction on p, assume $y \equiv x$ and $p \equiv \mathsf{refl}_x$. Then have $q : x =_A z$, so by path induction on q, assume $z \equiv x$ and $q \equiv \mathsf{refl}_x$. Then $\mathsf{refl}_x : x =_A x$ has the desired type.
Pattern-matching definition: $\mathsf{refl}_x \cdot \mathsf{refl}_x :\equiv \mathsf{refl}_x$. □

Similarly, can prove symmetry/inverses, so identity types are "equivalence relations". Concatenation also satisfies associativity, left and right unit laws, and inverse laws up to higher paths.

Warning

$K : \prod_{p : x =_A x} p =_{x =_A x} \mathsf{refl}_x$
$K(\mathsf{refl}_x) :\equiv \mathsf{refl}_{\mathsf{refl}_x}$

Truncations

Problem with logical interpretation: disjunctions and existentials may have non-identical proofs (encode too much data).
A type A is *contractible* if

$$\sum_{a:A} \prod_{x:A} (a =_A x).$$

Logical interpretation: A is a singleton.
Homotopical interpretation: there is a point $a : A$ with a *homotopy* $\mathrm{const}_a \sim \mathrm{id}_A$.
A type A is a *proposition* if for all $x, y : A$, $x =_A y$ is contractible (proof irrelevance).
A type A is a *set* if for all $x, y : A$ and $p, q : x =_A y$, we have $r : p =_{x =_A y} q$ (uniqueness of identity proofs).
$\mathbb{N}$ is a set; $\mathrm{isContr}(A), \mathrm{isProp}(A), \mathrm{isSet}(A)$ are propositions.
For any type A, can define its *propositional truncation* $\|A\|$ with a map $\eta : A \to \|A\|$ that is universal for maps from A to a proposition B.
Now define $A \vee B :\equiv \|A + B\|$, $\exists(x : A), B(x) :\equiv \|\Sigma(x : A), B(x)\|$.

Higher inductive types

Inductive types: types A "freely generated" by some constructors $X \to A$.

Example ($\mathbb{N}$)

- $0 : \mathbb{N}$
- $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$

Higher inductive types: allow *path constructors*.

Example (S^1)

- $\mathsf{base} : S^1$
- $\mathsf{loop} : \mathsf{base} =_{S^1} \mathsf{base}$

Example (T^2)

- $b : T^2$
- $p : b =_{T^2} b$
- $q : b =_{T^2} b$
- $t : p \cdot q =_{b =_{T^2} b} q \cdot p$

Higher inductive types (HITs)

Example (S^2)

- $\mathsf{base} : S^2$
- $\mathsf{surf} : \mathsf{refl}_{\mathsf{base}} =_{\mathsf{base}=_{S^2}\mathsf{base}} \mathsf{refl}_{\mathsf{base}}$

Can compute their homotopy groups, show $T^2 \cong S^1 \times S^1$, etc. internally. HITs are not just cell complexes!

Example (Propositional truncation $\|A\|$)

- $\eta : A \to \|A\|$
- $p : (x, y : \|A\|) \to x =_A y$

More generally, can define n-truncations that erase all homotopy data above dimension n, giving rise to the *Postinikov tower*.
HITs can also be used to define set-quotients or quotient types, which allow us to construct *(homotopy) colimits*.