Type Theory and Homotopy
I. Constructions and Dependence

Alex Kavvos

Panhellenic Logic Symposium, 6–10 July 2022

University of Bristol

I. Intuitionism and Constructions

Intuitionism, Constructivism, and Type Theory

- Many different philosophies: Brouwerian intuitionism, Heyting arithmetic, Russian constructivism, Bishop-style mathematics, etc. (see Stanford Encyclopedia of Philosophy entries)
- One common feature:

  > To prove that a mathematical object exists
  > you must show how to construct it.

- In particular, the details of the construction matter.
- Modern algebra: the structure of an isomorphism matters.
- **Martin-Löf Type Theory** (MLTT) was created as a formalization of Bishop-style constructive mathematics.
- Less focus on **truth**, more focus on **proof**.
- The **law of the excluded middle** (LEM) $\phi \vee \neg\phi$ is rejected.

Constructions

Let $A, B, \ldots$ be sets.

$$\mathbf{0} \stackrel{\text{def}}{=} \emptyset \qquad \mathbf{1} \stackrel{\text{def}}{=} \{*\}$$

$$A \times B \stackrel{\text{def}}{=} \{(a, b) \mid a \in A \text{ and } b \in B\} \quad A \to B \stackrel{\text{def}}{=} \{f \mid f : A \to B\}$$

$$A + B \stackrel{\text{def}}{=} \{(1, a) \mid a \in A\} \cup \{(2, b) \mid b \in B\}$$

Let $\neg A \stackrel{\text{def}}{=} A \to \mathbf{0}$.

Example

- $(x, y) \mapsto x \in (A \times B) \to A$
- $x \mapsto (y \mapsto (x, y)) \in A \to (B \to A \times B)$
- $\lambda x. \lambda y. (x, y) \in A \to (B \to A \times B)$
- $\lambda(x, v). \begin{cases} (1, (x, b)) & \text{if } v = (1, b) \\ (2, (x, c)) & \text{if } v = (2, c) \end{cases} \in A \times (B + C) \to (A \times B) + (A \times C)$
- $\lambda a. \lambda f. f(a) \in A \to \neg\neg A$

Dependence

Let $(B_a)_{a \in A}$ be a **family** of sets.

$$(a : A) \times B_a \overset{\text{def}}{=} \sum_{a \in A} B_a \overset{\text{def}}{=} \{(a, b) \mid a \in A \text{ and } b \in B_a\}$$

$$(a : A) \to B_a \overset{\text{def}}{=} \prod_{a \in A} B_a \overset{\text{def}}{=} \left\{ f : A \to \bigcup_{a \in A} B_a \mid f(a) \in B_a \text{ for all } a \in A \right\}$$

Given a **constant** family of sets $(B)_{a \in A}$ we have

$$(a : A) \times B = A \times B \qquad (a : A) \to B = A \to B$$

Example

Let $P_n \overset{\text{def}}{=} \begin{cases} \{*\} & \text{if n is prime} \\ \emptyset & \text{otherwise} \end{cases}$

- $(11, *) \in (n : \mathbb{N}) \times P_n$, but $(4, *) \notin (n : \mathbb{N}) \times P_n$
- $\lambda n.\, \text{if } n \text{ is prime then } (1, *) \text{ else } (2, \text{id}_\emptyset) \in (n : \mathbb{N}) \to P_n + \neg P_n$

II. Martin-Löf Type Theory

Martin-Löf Type Theory (MLTT)

- Invented by Per Martin-Löf in the late 1960s.
- A formal theory in **natural deduction** style.
- Every term in the theory needs to have a **type**.
- There are **no propositions**, only types.
  Every term is a **construction** which proves its **type**.

types = predicates
terms = proofs

- ZFC: engine (first-order logic) + fuel (axioms)
  MLTT: "engine and fuel all in one" (Pieter Hofstra, 1975–2022)

Judgements

Six distinct kinds of **judgement**:

Γ ctx	Γ is a context
$\Gamma \vdash A$ type	A is a type in context Γ
$\Gamma \vdash M : A$	M is a term of type A in context Γ
$\Gamma \equiv \Delta$ ctx	Γ and Δ are **definitionally** equal contexts
$\Gamma \vdash A \equiv B$ type	A and B are **definitionally** equal types
$\Gamma \vdash M \equiv N : A$	M and N are **definitionally** equal terms

The equality judgements have rules that make them

- equivalence relations, e.g. $\dfrac{\Gamma \vdash A \equiv B \text{ type}}{\Gamma \vdash B \equiv A \text{ type}}$
- congruences, e.g.

$$\frac{\Gamma \vdash A_1 \equiv A_2 \text{ type} \qquad \Gamma, x : A_1 \vdash B_1 \equiv B_2 \text{ type}}{\Gamma \vdash (x : A_1) \to B_1 \equiv (x : A_2) \to B_2 \text{ type}}$$

Contexts, variables, conversion

A **context** is a list of variables and their types.

$$\frac{}{\cdot \ \mathsf{ctx}} \qquad\qquad \frac{\Gamma \ \mathsf{ctx} \qquad \Gamma \vdash A \ \mathsf{type}}{\Gamma, x : A \ \mathsf{ctx}}$$

Variables stand for terms.
If I have a variable I can use it as a term:

$$\frac{\Gamma, x : A, \Delta \ \mathsf{ctx}}{\Gamma, x : A, \Delta \vdash x : A}$$

We can always replace definitionally equals by equals.
The **type conversion** rule:

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash A \equiv B \ \mathsf{type}}{\Gamma \vdash M : B}$$

What is a type?

It is a classifier of terms.

Terms of a certain type have an **interface**: a specification of how they can be created and consumed.

Ingredients of a type

- a **formation** rule (when can I form this type?)
- an **introduction** rule (how do I make terms of this type?)
- an **elimination** rule (how do I use terms of this type?)
- a **computation** rule (how do I calculate with its elements?)
- a **uniqueness** rule (what do terms of this type look like?)

Sometimes computation rules are called β **rules**
and uniqueness rules η **rules**.

Dependent function types / Π types

formation
$$\frac{\Gamma, x : A \vdash B \text{ type}}{\Gamma \vdash (x : A) \to B \text{ type}}$$

introduction
$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.\, M : (x : A) \to B}$$

elimination
$$\frac{\Gamma \vdash M : (x : A) \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M(N) : B[N/x]}$$

computation
$$\frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash N : A}{\Gamma \vdash (\lambda x : A.\, M)(N) \equiv M[N/x] : B[N/x]}$$

uniqueness
$$\frac{\Gamma \vdash M : (x : A) \to B}{\Gamma \vdash M \equiv \lambda x : A.\, M(x) : (x : A) \to B}$$

Dependent sum types / Σ types

formation

$$\frac{\Gamma, x : A \vdash B \text{ type}}{\Gamma \vdash (x : A) \times B \text{ type}}$$

introduction

$$\frac{\Gamma, x : A \vdash B \text{ type} \qquad \Gamma \vdash M : A \qquad \Gamma \vdash N : B[M/x]}{\Gamma \vdash (M, N) : (x : A) \times B}$$

elimination

$$\frac{\Gamma \vdash M : (x : A) \times B}{\Gamma \vdash \mathrm{pr}_1(M) : A} \qquad \frac{\Gamma \vdash M : (x : A) \times B}{\Gamma \vdash \mathrm{pr}_2(M) : B[\mathrm{pr}_1(M)/x]}$$

computation

$$\frac{\Gamma, x : A \vdash B \text{ type} \qquad \Gamma \vdash M : A \qquad \Gamma \vdash N : B[M/x]}{\Gamma \vdash \mathrm{pr}_1((M, N)) \equiv M : A}$$

$$\frac{\Gamma, x : A \vdash B \text{ type} \qquad \Gamma \vdash M : A \qquad \Gamma \vdash N : B[M/x]}{\Gamma \vdash \mathrm{pr}_2((M, N)) \equiv N : B[M/x]}$$

uniqueness

$$\frac{\Gamma \vdash M : (x : A) \times B}{\Gamma \vdash M \equiv (\mathrm{pr}_1(M), \mathrm{pr}_2(M)) : (x : A) \times B}$$

Coproducts (disjoint unions)

form.

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma \vdash A + B \text{ type}}$$

intro.

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathsf{inl}(M) : A + B} \quad \frac{\Gamma \vdash N : B}{\Gamma \vdash \mathsf{inr}(N) : A + B}$$

elim.

$$\frac{\begin{array}{c}\Gamma \vdash M : A + B \qquad \Gamma, c : A + B \vdash C \text{ type} \\ \Gamma, x : A \vdash P : C[\mathsf{inl}(x)/c] \qquad \Gamma, y : B \vdash Q : C[\mathsf{inr}(y)/c]\end{array}}{\Gamma \vdash \mathsf{case}_{[c.C]}(M; x.\, P; y.\, Q) : C[M/c]}$$

comp.

$$\frac{\begin{array}{c}\Gamma \vdash M : A + B \\ \Gamma, c : A + B \vdash C \text{ type} \qquad \Gamma, x : A \vdash P : C[\mathsf{inl}(x)/c] \\ \Gamma, y : B \vdash Q : C[\mathsf{inr}(y)/c] \qquad \Gamma \vdash E : A\end{array}}{\Gamma \vdash \mathsf{case}_{[c.C]}(\mathsf{inl}(E); x.\, P; y.\, Q) \equiv P[E/x] : C[\mathsf{inl}(E)/c]}$$

Natural numbers

form.

$$\frac{}{\Gamma \vdash \mathsf{Nat}\ \mathsf{type}}$$

intro.

$$\frac{}{\Gamma \vdash \mathsf{zero} : \mathsf{Nat}} \quad \frac{\Gamma \vdash N : \mathsf{Nat}}{\Gamma \vdash \mathsf{succ}(N) : \mathsf{Nat}}$$

elim.

$$\frac{\Gamma \vdash N : \mathsf{Nat} \qquad \Gamma, n : \mathsf{Nat} \vdash C\ \mathsf{type} \qquad \Gamma \vdash P : C[\mathsf{zero}/n] \qquad \Gamma, n : \mathsf{Nat}, c : C \vdash Q : C[\mathsf{succ}(n)/n]}{\Gamma \vdash \mathsf{rec}_{[c.C]}(N; P; n, c.\ Q) : C[N/n]}$$

comp.

$$\frac{\ldots}{\Gamma \vdash \mathsf{rec}_{[c.C]}(\mathsf{zero}; P; n, c.\ Q) \equiv P : C[\mathsf{zero}/n]}$$

$$\frac{\ldots}{\Gamma \vdash \mathsf{rec}_{[c.C]}(\mathsf{succ}(x); P; n, c.\ Q) \equiv Q[x, \mathsf{rec}(x; P; n, c.\ Q)/n, c] : C[\mathsf{succ}(x)/n]}$$

Metatheory (I)

Let $\mathcal{J}$ stand for either A type or $M : A$.

Theorem (Weakening)

The following rule is admissible: $\dfrac{\Gamma, \Delta \vdash \mathcal{J} \qquad \Gamma \vdash A \text{ type}}{\Gamma, x : A, \Delta \vdash \mathcal{J}}$

Theorem (Substitution / Cut)

The following rule is admissible: $\dfrac{\Gamma \vdash M : A \qquad \Gamma, x : A, \Delta \vdash \mathcal{J}}{\Gamma, \Delta[M/x] \vdash \mathcal{J}[M/x]}$

Theorem

There is a set-theoretic model of MLTT with Π, Σ, Nat, *and* $+$ *types.*

The model can also be constructed in CZF (constructive ZF).

Corollary: the theory is consistent (if the ambient metatheory is).

Metatheory (II)

Theorem (Canonicity)

Let $\vdash M : C$*. Then:*

- *if* $C \equiv A + B$ *then either* $\vdash M \equiv \mathsf{inl}(P) : A + B$ *for some* $\vdash P : A$ *or* $\vdash N \equiv \mathsf{inr}(Q) : A + B$ *for some* $\vdash Q : B$,
- *if* $C \equiv \mathsf{Nat}$ *then* $\vdash M \equiv succ^n(\mathsf{zero}) : \mathsf{Nat}$ *for some* $n \in \mathbb{N}$
- *if* $C \equiv (x : A) \times B$ *then* $\vdash M \equiv (P, Q) : (x : A) \times B$ *for some* $\vdash P : A$ *and* $\vdash Q : B[P/x]$

Moreover, finding the "canonical form" of such terms is computable.

Theorem (Normalization)

Given Γ, M, N *and* A*, it is decidable whether* $\Gamma \vdash M \equiv N : A$.

Theorem (Decidability)

Given Γ*, and any judgement* $\mathcal{J}$*, it is decidable whether* $\Gamma \vdash \mathcal{J}$.

These properties give MLTT its computational flavour.

III. Examples

Propositional constructions

Types are propositions. Terms are proofs.

Define:

$$\wedge \stackrel{\text{def}}{=} \times \qquad\qquad \vee \stackrel{\text{def}}{=} +$$

Given $\vdash A, B$ type we have

- $\vdash \lambda x.\, \lambda y.\, x : A \to B \to A$
- $\vdash \lambda x.\, \lambda y.\, (x, y) : A \to B \to A \wedge B$
- $\vdash \lambda p.\, (\mathrm{pr}_2(p), \mathrm{pr}_1(p)) : A \wedge B \to B \wedge A$
- $\vdash \lambda u.\, \mathrm{case}(u; x.\, \mathrm{inr}(x); y.\, \mathrm{inl}(y)) : A \vee B \to B \vee A$

Theorem (Curry-Howard correspondence)

All intuitionistically valid formulas/types are inhabited.

Addition

Let $\Gamma \stackrel{\text{def}}{=} x : \mathsf{Nat}, y : \mathsf{Nat}$.

$$\dfrac{\dfrac{}{\Gamma \vdash x : \mathsf{Nat}} \qquad \dfrac{}{\Gamma \vdash y : \mathsf{Nat}} \qquad \dfrac{\dfrac{}{\Gamma, n : \mathsf{Nat}, c : \mathsf{Nat} \vdash n : \mathsf{Nat}}}{\Gamma, n : \mathsf{Nat}, c : \mathsf{Nat} \vdash \mathsf{succ}(n) : \mathsf{Nat}}}{\Gamma \vdash \mathsf{rec}_{[\_.\mathsf{Nat}]}(x; y; n, c.\, \mathsf{succ}(c)) : \mathsf{Nat}}$$

So we can define

$$\vdash \mathsf{add} = \lambda x.\, \lambda y.\, \mathsf{rec}(x; y; n, c.\, \mathsf{succ}(n)) : \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat}$$

and compute

$$y : \mathsf{Nat} \vdash \mathsf{add}(\mathsf{zero})(y) \equiv y : \mathsf{Nat}$$
$$y : \mathsf{Nat} \vdash \mathsf{add}(\mathsf{succ}(\mathsf{zero}))(y) \equiv \mathsf{succ}(y) : \mathsf{Nat}$$

and so on.

A familiar construction (I)

Let $\Gamma \vdash A, B$ type, and $x : A, y : B \vdash R(x, y)$ type. Then

$$\dfrac{\dfrac{\dfrac{}{x : A, y : B \vdash R(x, y) \text{ type}}}{x : A \vdash (y : B) \times R(x, y) \text{ type}}}{\vdash (x : A) \to (y : B) \times R(x, y) \text{ type}}$$

This is essentially $\forall x : A.\ \exists y : B.\ R(x, y)$.

Similarly, recalling that $A \to B \stackrel{\text{def}}{=} (x : A) \to B$, we have

$$\dfrac{\dfrac{\vdots}{f : A \to B, x : A \vdash R(x, f(x)) \text{ type}}}{\vdash (f : A \to B) \times ((x : A) \to R(x, f(x))) \text{ type}}$$

This is essentially $\exists f : A \to B.\ \forall x : A.\ R(x, f(x))$.

A familiar construction (II)

Let $\Gamma \vdash A, B$ type, and $x : A, y : B \vdash R(x, y)$ type. Then

$$\vdash (x : A) \to (y : B) \times R(x, y) \text{ type}$$

This is essentially $\forall x : A.\ \exists y : B.\ R(x, y)$.
Similarly, recalling that $A \to B \stackrel{\text{def}}{=} (x : A) \to B$, we have

$$\vdash (f : A \to B) \times ((x : A) \to R(x, f(x))) \text{ type}$$

This is essentially $\exists f : A \to B.\ \forall x : A.\ R(x, f(x))$.

$$\Gamma \vdash ? : ((x : A) \to (y : B) \times R(x, y)) \\ \to ((f : A \to B) \times ((x : A) \to R(x, f(x))))$$

A familiar construction (II)

Let $\Gamma \vdash A, B$ type, and $x : A, y : B \vdash R(x, y)$ type. Then

$$\vdash (x : A) \to (y : B) \times R(x, y) \text{ type}$$

This is essentially $\forall x : A.\ \exists y : B.\ R(x, y)$.
Similarly, recalling that $A \to B \overset{\text{def}}{=} (x : A) \to B$, we have

$$\vdash (f : A \to B) \times ((x : A) \to R(x, f(x))) \text{ type}$$

This is essentially $\exists f : A \to B.\ \forall x : A.\ R(x, f(x))$.

$$\Gamma \vdash ? : ((x : A) \to (y : B) \times R(x, y)) \\ \to ((f : A \to B) \times ((x : A) \to R(x, f(x))))$$

Indeed, this is the **type-theoretic "axiom" of choice**:

$$\Gamma \vdash \lambda g.\,(\lambda x.\,\mathrm{pr}_1(g(x)), \lambda x.\,\mathrm{pr}_2(g(x))) : ((x : A) \to (y : B) \times R(x, y)) \\ \to ((f : A \to B) \times ((x : A) \to R(x, f(x))))$$

The type-theoretic “axiom” of choice

Let $\Gamma \vdash A, B$ type, and $x : A, y : B \vdash R(x, y)$ type. Then

$$\Gamma \vdash \lambda g.\,(\lambda x.\,\mathrm{pr}_1(g(x)), \lambda x.\,\mathrm{pr}_2(g(x))) : ((x : A) \to (y : B) \times R(x, y)) \\ \to ((f : A \to B) \times ((x : A) \to R(x, f(x))))$$

Suppose $g : (x : A) \to (y : B) \times R(x, y)$. Then clearly

$$f_g \overset{\text{def}}{=} \lambda x : A.\,\underbrace{\mathrm{pr}_1(\overbrace{g(x)}^{(y:B)\times R(x,y)})}_{B} : A \to B$$

$$h_g \overset{\text{def}}{=} \lambda x : A.\,\underbrace{\mathrm{pr}_2(\overbrace{g(x)}^{(y:B)\times R(x,y)})}_{R(x,\mathrm{pr}_1(g(x)))} : (x : A) \to R(x, \mathrm{pr}_1(g(x)))$$

But $f(x) \equiv \mathrm{pr}_1(g(x))$, so this type is equal to $(x : A) \to R(x, f(x))$.
Hence $\lambda g.\,(f_g, h_g)$ has the right type.

Summary

- MLTT is a formal theory of **constructions** and **dependence**.
- It has very good metatheoretic and computational properties.
- It is inherently "constructive" (for some sense of the word).

Tomorrow: equality as a proposition/type.

References

Martin Hofmann, *Syntax and Semantics of Dependent Types*, Semantics and Logics of Computation (Andrew M. Pitts and P. Dybjer, eds.), Cambridge University Press, 1997, pp. 79–130.

Per Martin-Löf, *An Intuitionistic Theory of Types: Predicative Part*, Logic Colloquium '73 (H. E. Rose and J. C. Shepherdson, eds.), Studies in Logic and the Foundations of Mathematics, no. 80, Elsevier, Bristol, 1975, pp. 73–118.

———, *Constructive mathematics and computer programming*, Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences **312** (1984), no. 1522, 501–518.

Bengt Nordström, Kent Petersson, and Jan M. Smith, *Programming in Martin-Löf's Type Theory: an Introduction*, Oxford University Press, 1990.