Type theory as new constructive foundations for mathematics, logic, and computer science

Jad Issa

American University of Beirut

February 11, 2021

Sources

The following presentation heavily relies (non-exhaustively) on these resources:

- Proofs and Types, Jean-Yves Girard
- Homotopy Type Theory, The Univalent Foundations Program, Institute of Advanced Study
- Oregon Programming Languages Summer School 2012

Motivation

Low-level mathematics

- $a = b$ is supposed to mean a is **the exact same thing as** b.
- What about $4 \times 5 = 20$, $x^2 - y^2 = (x-y)(x+y)$ or $xy = yx$?
- $xy \not\equiv yx$; however, $xy = yx$.
- Similarly, $A \times B \neq B \times A$, but $A \times B \cong B \times A$.
- Distinction between proofs.
- Too obstructive for modern mathematics.
- Automatic proof verification.

Type Theory

Type Theory

Judgements vs propositions and types vs sets

Type theory	Set theory
Type	Set
Term	Element
$a : A$	$a \in A$
$a \equiv b$	$a = b$
Judgement	Proposition

The proposition that $0 \in \mathbb{N}$ can be argued to be true or false, however, the judgement that $0 : \mathbb{N}$ is definitionally true about 0 and asserted true.
Formally, $0 : \mathbb{N}$ is a different 'entity' than $0 : \mathbb{R}$, one could write $0_{\mathbb{N}}$ and $0_{\mathbb{R}}$.

Type Theory

All about functionality

Type theory is all about how elements of a type behave.

$$\frac{}{\Gamma \vdash 0 : \mathbb{N}} \ \mathbb{N}\mathcal{I}_0 \qquad\qquad \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash Sn : \mathbb{N}} \ \mathbb{N}\mathcal{I}_S$$

$$\frac{\Gamma \vdash C : \mathcal{U} \qquad \Gamma \vdash c_0 : C \qquad \Gamma, n : \mathbb{N}, c_n : C \vdash c_{n+1} : C}{\Gamma, n : \mathbb{N} \vdash c_n : C} \ \mathbb{N}\mathcal{E}$$

Type Theory

All about functionality

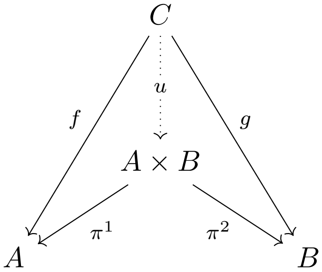
Type theory is all about how elements of a type behave.

$$\mathbb{N}\mathcal{I}_0 \quad 0 : \mathbb{N} \qquad\qquad \mathbb{N}\mathcal{I}_S \quad S : \mathbb{N} \to \mathbb{N}$$

$$\mathbb{N}\mathcal{E} \quad \mathsf{rec}_{\mathbb{N}} : \prod_{(C:\mathcal{U})} C \to (\mathbb{N} \to C \to C) \to (\mathbb{N} \to C)$$

Type Theory

Category theory



$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash (a, b) : A \times B} \times\mathcal{I}$$

$$\frac{\Gamma \vdash v : A \times B}{\Gamma \vdash \pi^1 v : A} \times\mathcal{E}_1 \qquad \frac{\Gamma \vdash v : A \times B}{\Gamma \vdash \pi^2 v : B} \times\mathcal{E}_2$$

Product

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash (a, b) : A \times B} \times\mathcal{I}$$

$$\frac{\Gamma \vdash v : A \times B}{\Gamma \vdash \pi^1 v : A} \times\mathcal{E}_1 \qquad \frac{\Gamma \vdash v : A \times B}{\Gamma \vdash \pi^2 v : B} \times\mathcal{E}_2$$

$$\begin{array}{lll} \times\mathcal{I} & (\cdot,\cdot) & : A \to B \to A \times B \\ \times\mathcal{E}_1 & \pi^1 & : A \times B \to A \\ \times\mathcal{E}_2 & \pi^2 & : A \times B \to B \end{array}$$

Product

$$\frac{\Gamma \vdash A : \mathcal{U} \qquad \Gamma \vdash B : \mathcal{U}}{\Gamma \vdash A \times B : \mathcal{U}} \; \mathcal{U}\mathcal{I}_\times$$

$$\frac{\Gamma \vdash A : \mathcal{U} \qquad \Gamma \vdash B : \mathcal{U} \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash (a, b) : A \times B} \; \times\mathcal{I}$$

$$\frac{\Gamma \vdash A : \mathcal{U} \qquad \Gamma \vdash B : \mathcal{U} \qquad \Gamma \vdash v : A \times B}{\Gamma \vdash \pi^1 v : A} \; \times\mathcal{E}_1$$

$$\frac{\Gamma \vdash A : \mathcal{U} \qquad \Gamma \vdash B : \mathcal{U} \qquad \Gamma \vdash v : A \times B}{\Gamma \vdash \pi^2 v : B} \; \times\mathcal{E}_2$$

$$\times\mathcal{I} \quad (\cdot, \cdot) : \prod_{A,B:\mathcal{U}} A \to B \to A \times B$$

$$\times\mathcal{E}_1 \quad \pi^1 : \prod_{A,B:\mathcal{U}} A \times B \to A$$

$$\times\mathcal{E}_2 \quad \pi^2 : \prod_{A,B:\mathcal{U}} A \times B \to B$$

Product

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash (a, b) : A \times B} \times\mathcal{I}$$

$$\frac{\Gamma \vdash v : A \times B}{\Gamma \vdash \pi^1 v : A} \times\mathcal{E}_1 \qquad \frac{\Gamma \vdash v : A \times B}{\Gamma \vdash \pi^2 v : B} \times\mathcal{E}_2$$

$$\begin{array}{rll} \times\mathcal{I} & (\cdot, \cdot) & : A \to B \to A \times B \\ \times\mathcal{E}_1 & \pi^1 & : A \times B \to A \\ \times\mathcal{E}_2 & \pi^2 & : A \times B \to B \end{array}$$

Coproduct

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{inl}\ a : A + B} +\mathcal{I}_l \qquad \frac{\Gamma \vdash b : B}{\Gamma \vdash \mathsf{inr}\ b : A + B} +\mathcal{I}_r$$

$$\frac{\Gamma, a : A \vdash c : C \qquad \Gamma, b : B \vdash c : C}{\Gamma, v : A + B \vdash c : C} +\mathcal{E}$$

$$\begin{array}{rll} +\mathcal{I}_l & \mathsf{inl} & : A \to A + B \\ +\mathcal{I}_r & \mathsf{inr} & : B \to A + B \\ +\mathcal{E} & \mathsf{case} & : (A \to C) \to (B \to C) \to (A + B \to C) \end{array}$$

Exponents

$$\frac{\Gamma, a : A \vdash b : B}{\Gamma \vdash (\lambda a.b) : A \to B} \to \mathcal{I} \text{ (λ-abstraction)}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash f : A \to B}{\Gamma \vdash f a : B} \to \mathcal{E} \text{ (evaluation)}$$

A universe $\mathcal{U}$ is a type of (small) types. We thus internalize types.

"A is a type" becomes a judgment $A : \mathcal{U}$.

To avoid Russel's paradox, we can make a tower of universes as such.

$$\mathcal{U}_0 : \mathcal{U}_1 : \mathcal{U}_2 : \cdots$$

Type Theory

Type families/dependent types

A type family (or dependent type) B over a type A is a type-valued function parametrized over the elements of A.

$$\begin{aligned} B : A &\to \mathcal{U} \\ a &\mapsto B(a) : \mathcal{U} \end{aligned}$$

Dependent functions / universals

$$\frac{\Gamma, a : A \vdash B(a) : \mathcal{U} \qquad \Gamma, a : A \vdash b : B(a)}{\Gamma \vdash (\lambda a.b) : \prod_{a:A} B(a)} \prod \mathcal{I}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash f : \prod_{a:A} B(a)}{\Gamma \vdash fa : B(a)} \prod \mathcal{E}$$

Dependent pairs / existentials

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B(a)}{\Gamma \vdash (a, b) : \sum_{x:A} B(x)} \sum\mathcal{I}$$

$$\frac{\Gamma \vdash v : \sum_{x:A} B(x)}{\Gamma \vdash \pi^1 v : A} \sum\mathcal{E}_1 \qquad \frac{\Gamma \vdash v : \sum_{x:A} B(x)}{\Gamma \vdash \pi^2 v : B(\pi^1(x))} \sum\mathcal{E}_2$$

$$\begin{array}{lll} \sum\mathcal{I} & (\cdot, \cdot) & : \prod_{x:A} \left(B(x) \to \sum_{x:A} B(x) \right) \\ \sum\mathcal{E}_1 & \pi^1 & : \left(\sum_{x:A} B(x) \right) \to A \\ \sum\mathcal{E}_2 & \pi^2 & : \prod_{v:\sum_{x:A} B(x)} B(\pi^1(v)) \end{array}$$

Curry-Howard correspondence

Curry-Howard Correspondence

Basics

Logic	Type theory
Proposition	Type[1]
n-ary predicate	n-ary dependent type
Proof	Element
$A \wedge B$	$A \times B$
$A \vee B$	$A + B$
$A \implies B$	$A \to B$ or B^A
$\neg A$	$A \to \bot$
$\forall x \in A, C(x)$	$\prod_{x:A} C(x)$
$\exists x \in A, C(x)$	$\sum_{x:A} C(x)$
Contradiction	$\bot$
Tautology	$\top$

[1] Booleans are the type **2** and so called 'mere propositions' are special types, but in general, the correspondence is with general types

There is no proof of the law of the excluded middle ($P \vee \neg P$) for general types, however, it is not refuted (thus, consistent with (intuitionistic) type theory), $\neg\neg(P \vee \neg P)$ is provable.

Proof.

We want an element of type $((P + (P \to \bot)) \to \bot) \to \bot$. This is a function whose argument is a function $f : P + (P \to \bot) \to \bot$. We want to construct an element of $\bot$ from f. Note: $f \circ \mathsf{inl} : P \to \bot$. $f \circ \mathsf{inr} : (P \to \bot) \to \bot$. So, $(f \circ \mathsf{inr})(f \circ \mathsf{inl}) : \bot$, thus the whole proof is the function

$$(\lambda f.(f \circ \mathsf{inr})(f \circ \mathsf{inl})) : \neg\neg(P \vee \neg P)$$

□

Group axioms

$$\mathsf{Grp} :\equiv \sum_{A:\mathcal{U}} \sum_{f:A\to A\to A} \sum_{e:A} \prod_{x,y,z:A} (f(e,x)=x)\times (f(x,e)=x)\times (f(x,f(y,z))=f(f(x,y),z))\times \sum_{x':A}(f(x,x')=e)\times (f(x',x)=e)$$

Every group here consists of the classical tuple (A, f, e) along with proofs of the axioms.

Quantitative logic

A proof can only be used **once** in constructing any other proof and cannot be ‘destroyed’.

We remove weakening and contraction.

$$\text{WR}\,\frac{\Gamma \vdash \Sigma}{\Gamma \vdash \Sigma, A} \qquad \frac{\Gamma \vdash \Sigma}{\Gamma, A \vdash \Sigma}\,\text{WL}$$

$$\text{CR}\,\frac{\Gamma \vdash \Sigma, A, A}{\Gamma \vdash \Sigma, A} \qquad \frac{\Gamma, A, A \vdash \Sigma}{\Gamma, A \vdash \Sigma}\,\text{CL}$$

Quantitative logic

Quantitative type theory in programming

- Qubits cannot be duplicated (no-cloning) $\implies$ linear types.
- Converging toward providing proofs alongside programs.

```
map : !n (a -> b) -> !1 (Vector a n) -> (Vector b n)
map(f, [])                 = []
map(f, [x0, ...xi...]) = [f(x0), ...map(f, xi)...]
```

Idris 2

Homotopy Type Theory

Internalize equality and generalize it. First introduced by Per Martin-Löf. Inductive type generated by reflexivity.

$$\frac{\Gamma \vdash x : A, y : A}{\Gamma \vdash \mathsf{refl}_x : x = x} =_A \mathcal{I}$$

$$\frac{\Gamma, x, y : A, p : x = y \vdash C(x, y, p) : \mathcal{U} \qquad \Gamma, x : A \vdash c : C(x, x, \mathsf{refl}_x)}{\Gamma, x, y : A, p : x = y \vdash c : C(x, y, p)} =_A \mathcal{E}$$

Internalize equality and generalize it. First introduced by Per Martin-Löf. Inductive type generated by reflexivity.

$$\frac{\Gamma \vdash x : A, y : A}{\Gamma \vdash \mathsf{refl}_x : x = x} =_A \mathcal{I}$$

$$\frac{\color{red}{\Gamma, x, y : A, p : x = y \vdash C(x, y, p) : \mathcal{U}} \qquad \Gamma, x : A \vdash c : C(x, x, \mathsf{refl}_x)}{\Gamma, x, y : A, p : x = y \vdash c : C(x, y, p)} =_A \mathcal{E}$$

Homotopy Type Theory

Identity types - symmetry/inverses

Symmetry here is a **function** of proofs of equality.

$$(\cdot)^{-1} : x = y \to y = x$$

$$\mathsf{sym} : \prod_{A:\mathcal{U}} \prod_{x,y:A} x = y \to y = x$$

By induction, assume $x \equiv y$ and $p \equiv \mathsf{refl}_x$.

$$(\mathsf{refl}_x)^{-1} :\equiv \mathsf{sym}(A, x, x, \mathsf{refl}_x) :\equiv \mathsf{refl}_x$$

In analogy with induction on natural numbers, we have **completely** defined $(\cdot)^{-1}$.

Homotopy Type Theory

Identity types - transitivity/concatenation

Transitivity is also a (binary) function.

$$\cdot : x = y \to y = z \to x = z$$

Double induction:

$$\mathsf{refl}_x \cdot \mathsf{refl}_x :\equiv \mathsf{refl}_x$$

Single induction:

$$p \cdot \mathsf{refl}_x :\equiv p$$

$$\mathsf{refl}_x \cdot q :\equiv q$$

Is this the case?

$$\prod_{A:\mathcal{U}} \prod_{x,y:A} \prod_{p:x=y} p = \mathsf{refl}_x$$

Homotopy Type Theory

Identity types - Non-trivial paths

Is this the case?

$$\prod_{A:\mathcal{U}} \prod_{x,y:A} \prod_{p:x=y} p = \mathsf{refl}_x$$

Not well-typed!

Is this the case?

$$\prod_{A:\mathcal{U}} \prod_{x,y:A} \prod_{p:x=y} p = \mathsf{refl}_x$$

Not well-typed!

Is this the case?

$$\prod_{A:\mathcal{U}} \prod_{x:A} \prod_{p:x=x} p = \mathsf{refl}_x$$

Homotopy Type Theory

Identity types - Non-trivial paths

Is this the case?

$$\prod_{A:\mathcal{U}} \prod_{x,y:A} \prod_{p:x=y} p = \mathsf{refl}_x$$

Not well-typed!

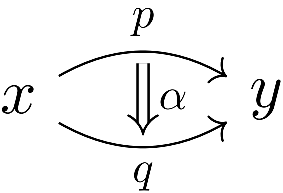
Is this the case?

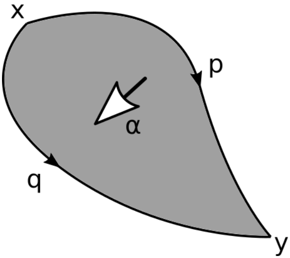
$$\prod_{A:\mathcal{U}} \prod_{x:A} \prod_{p:x=x} p = \mathsf{refl}_x$$

Not a **binary** relation!

Homotopy Type Theory

Paths and homotopies, ∞-groupoids





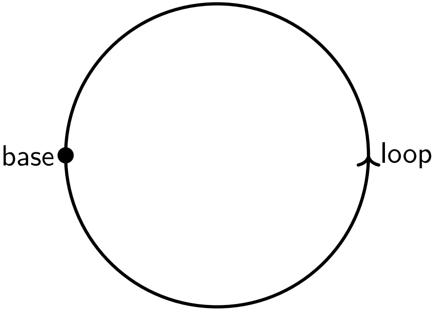
$$p : x =_A y$$

$$q : x =_A y$$

$$\alpha : p =_{x=y} q$$

Homotopy Type Theory

Higher inductive types



$$\mathsf{base} : \mathbb{S}^1$$

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

(Formal) equivalence (i.e. isomorphism) is equivalent to identity

$$\mathsf{ua} : \prod_{A,B:\mathcal{U}} (A \cong B) \cong (A =_{\mathcal{U}} B)$$

Thank you!