Type theory and the logic of toposes

Xuanrui Qi [1]

[1]Graduate School of Mathematics, Nagoya University

July 21, 2021

Synopsis

Correspondence between type theory and category theory

- Simply-typed λ-calculus $\rightleftharpoons$ cartesian closed categories
- Extensional Martin-Löf type theory $\rightarrow$ presheaf categoies
- Extensional calculus of constructions $\rightleftharpoons$ elementary toposes
- Intensional type theory $\rightleftharpoons$ $(\infty, 1)$-Grothendieck toposes

Some category theory

exponential (Y^X, ev) = "internal hom". In **Set**: $Y^X = X \to Y$

subobject classifier (Ω, true), "classifies" monomorphisms. In **Set**: 2-element set.

cartesian closed category (CCC) category w/ all finite products and exponentials

(elementary) topos category w/ all finite limits & colimits, exponentials, and a subobject classifer

Some category theory

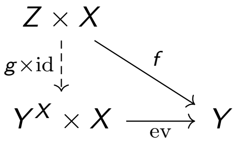

Figure: UP for exponentials

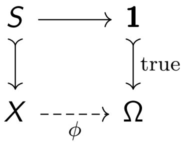

Figure: the subobject classifier pullback diagram

What is the simply typed λ-calculus (STLC)?

a simple computational language for logic

Expressions (terms) and types:

function type $(A \to B)$ functions $\lambda x.a$, can be applied $f\ a$ (or $f(a)$)

product type $(A \times B)$ pairs (a, b), canonical projections fst, snd

singleton type $(\mathbf{1})$ single inhabitant tt

reduction notion of computation, "applying functions"

$\beta\eta$-equivalence computational notion of equivalence, observational equivalence/equality

$$(\lambda x.x\ y)(\lambda x.x) \to (x\ y)[x := (\lambda x'.x')]$$
$$\to (\lambda x'.x')y \to x'[x' := y] \to y$$

Typing rules

Typing rules: $\Gamma \vdash M : \tau$, "under Γ (context = mapping from variables to types), M has type τ"

$$\textsc{Typ-Abs}\quad \frac{\Gamma; x : \tau \vdash M : \tau'}{\Gamma \vdash \lambda(x : \tau).M : \tau \to \tau'}$$

$$\textsc{Typ-App}\quad \frac{\Gamma \vdash M : \tau \to \tau' \qquad \Gamma \vdash N : \tau}{\Gamma \vdash M\ N : \tau'}$$

The Curry-Howard correspondence

STLC considered as logic: types = proposition, terms = proof.

The Curry-Howard correspondence

STLC considered as logic: types = proposition, terms = proof.

Proof as computation: prove B from A = computational procedure to produce proof of B from proof of A.

Specifically: STLC = constructive propositional logic. $\to\, = \implies$, $\times = \wedge$, proves same propositions.

STLC and CCCs (1): syntactic category

Construct category from STLC:

- objects = types
- morphisms = functions $(\lambda x.M(x))$
- composition of morphisms = composition of functions
- identity = identity function $(\lambda x.x)$

This is a CCC!

STLC and CCCs (2): the other way round

"Interpret" STLC in a category: assign object to type, contexts to products of types, terms to morphisms, etc.

Theorem: STLC can be interpreted in any CCC!

Construction: recursion on typing derivation, application as evaluation map, λ as exponential, product as categorical product, etc.

STLC and CCCs (2): the other way round

"Interpret" STLC in a category: assign object to type, contexts to products of types, terms to morphisms, etc.

Theorem: STLC can be interpreted in any CCC!

Construction: recursion on typing derivation, application as evaluation map, λ as exponential, product as categorical product, etc.

Important property: soundness. $\beta\eta$-equivalent terms have same interpretation.

STLC and CCCs (3): the equivalence

Construction: to every CCC, can construct language w/ types from objects, terms from morphisms, etc.

STLC and CCCs (3): the equivalence

Construction: to every CCC, can construct language w/ types from objects, terms from morphisms, etc.

This language is STLC!

STLC and CCCs (3): the equivalence

Construction: to every CCC, can construct language w/ types from objects, terms from morphisms, etc.

This language is STLC!

Result: "equivalence" between STLC and CCCs.

Martin-Löf type theory

Problem with STLC: very weak, no quantifiers, impractical as a logic.

Solution: **Martin-Löf type theory** (MLTT)
$\Pi(x : A).B$ (dependent function) and $\Sigma(x : A).B$ (dependent product), extended version of $\to$ and $\times$ to allow type depending on terms.

Curry-Howard: $\Pi = \forall$, $\Sigma = \exists$, equivalent to constructive higher-order logic.

Close to "natural" mathematical language w/ unrestricted quantification!

The identity type

New type: $a =_A b$, a type/proposition encoding equality of a and b (propositional equality/equivalence).

- Curry-Howard: the proposition of logical equality
- Equivalent to observational/$\beta\eta$-equivalence

The identity type

New type: $a =_A b$, a type/proposition encoding equality of a and b (propositional equality/equivalence).

- Curry-Howard: the proposition of logical equality
- Equivalent to observational/$\beta\eta$-equivalence

MLTT also has definitional/intrinsic notion of equality $\Gamma \vdash a \equiv b : A$, used e.g. to decide if two things should be considered equal in proof-checking.

Universes

Problem: need "type of types" (e.g., to quantify over types in Π or Σ)

Naïve solution: $A : \mathcal{U}$, $\mathcal{U} : \mathcal{U}$ (DANGER!, Russell's paradox)

Proper solution: hierarchy of universes $\mathcal{U}_0$, $\mathcal{U}_1$, ..., such that $\mathcal{U}_i : \mathcal{U}_{i+1}$
If $\Gamma \vdash A$ type$_i$ then $\Gamma \vdash A : \mathcal{U}_i$

Category with families (CwF)

A categorical structure close to the syntax to type theory; "scaffolding" for semantics.

- objects = contexts
- functors $\mathrm{Ty}(-) : \mathcal{C} \to \mathbf{Set}$, $\mathrm{Tm}(\Gamma; A) : \mathbf{Set}$
- for $\Gamma : \mathcal{C}$ and $A \in \mathrm{Ty}(\Gamma)$, an "extension" $\Gamma.A : \mathcal{C}$

Close to syntax, so construct CwF structure on $\mathcal{C}$ = get a model of MLTT in $\mathcal{C}$

Interlude: Grothendieck construction

presheaf on $\mathcal{C}$ = functor $\mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ (example: Yoneda functor $\mathrm{Hom}(-, C)$)

Grothendieck construction $\int_{\mathcal{C}} F$ on presheaf $F : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ = "category of elements" of F. More precisely, pairs (X, p), where $X : \mathcal{C}$, $p \in F(X)$.

Interlude: Grothendieck construction

presheaf on $\mathcal{C}$ = functor $\mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ (example: Yoneda functor $\mathrm{Hom}(-, C)$)

Grothendieck construction $\int_{\mathcal{C}} F$ on presheaf $F : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ = "category of elements" of F. More precisely, pairs (X, p), where $X : \mathcal{C}$, $p \in F(X)$.

"Structured, categorized data to one unstructured big table"

The presheaf model

Theorem: MLTT has a model in any presheaf category, i.e. $\mathbf{PSh}(\mathcal{C}) = [\mathcal{C}^{\mathrm{op}}; \mathbf{Set}]$ where $\mathcal{C}$ small

Construct CwF from any presheaf category:

- context Γ = presheaf Γ
- $\mathrm{Ty}(\Gamma) = \mathbf{PSh}(\int_{\mathcal{C}} \Gamma)$
- each type is a presheaf i.e. a family of sets parametrized by $I : \mathcal{C}$
- other constructions

The presheaf model: Π, Σ, =

Presheaf model can interpret Π, Σ, =:

- Π interpreted by family of functions interpreted by morphisms $f : J \to I$ in $\mathcal{C}$
- abstractly: right adjoint to pullback/base change functor
- Σ interpreted by categorical products in CwF (which exist in any category of presheaves)
- $=$ interpred by equalizers in CwF (which exist in any category of presheaves)

Bonus: MLTT in any topos

$\mathbf{PSh}(\mathcal{C})$ is a topos.
Can construct CwF in any elementary topos $\mathcal{E}$!

- context Γ = object $\Gamma : \mathcal{E}$
- same: "right adjoint to pullback functor", product, equalizer
- bonus: a universe of propositions

The universe of propositions

Elementary toposes support, in addition to $\mathcal{U}_i$, a universe Prop of *logical propositions*

- Prop is impredicative: quantifications over Prop still in Prop, so no universe levels needed
- Prop is proof-irrelevant: $\vdash p \equiv q : P$ if $P : \mathrm{Prop}$ (any two proofs of same proposition considered equal)
- The subobject classifier provides semantics for Prop
- MLTT + Prop = calculus of constructions (CoC); elementary topos = CoC!

Intensional vs extensional type theory

Two different notions of equality: definitional/intrinsic equality (definitional $\equiv$), observational/extrinsic equality ($\equiv_{\beta\eta}$, propositional equality $=_A$)

Generally: definitional equality $\subset$ observational equality! There are terms observationally equal but not definitionally equal.

Solution: either accept (intensional type theory), or fix by making observationally equal terms definitionally equal ("reflection rule"):

$$\textsc{Eq-Ref}\quad \frac{\Gamma \vdash p : x =_A y}{\Gamma \vdash x \equiv y : A}$$

Extensional type theory

Type theory + reflection rule = extensional type theory!
Consequences:

- uniqueness of identity proofs (UIP): if $p, q : a =_A b$ then $p =_{a =_A b} q$
- deciding typing (given Γ, a and A, check if $\Gamma \vdash a : A$) becomes undecidable
- fits our intuition about equality, but computationally bad

Extensional type theory

Type theory + reflection rule = extensional type theory!
Consequences:

- uniqueness of identity proofs (UIP): if $p, q : a =_A b$ then $p =_{a =_A b} q$
- deciding typing (given Γ, a and A, check if $\Gamma \vdash a : A$) becomes undecidable
- fits our intuition about equality, but computationally bad

Presheaf categories (and toposes in general) support only extensional type theory! (because equalizers are unique)

Interlude: ∞-categories

- Many different definitions
- One possible definition ($(\infty, 1)$-category): a simplicial set (presheaf on simplex category Δ) satisfying certain conditions
- Roughly speaking: morphisms, 2-morphisms between morphisms, 3-morphisms between 2-morphisms, etc.
- Can also define ∞ version of groupoid (∞-groupoid, ∞-category where all morphisms are (weak) equivalences)

Homotopy type theory

In intensional TT, types A are groupoids (identities $a =_A b$ are the invertible morphisms)

Identities of identities $p =_{a=_A b} q$, etc., exist, so types are ∞-groupoids!
(Grothendieck's homotopy hypothesis: ∞-groupoid $\equiv$ topological space?)

Homotopy type theory

In intensional TT, types A are groupoids (identities $a =_A b$ are the invertible morphisms)

Identities of identities $p =_{a=_A b} q$, etc., exist, so types are ∞-groupoids!
(Grothendieck's homotopy hypothesis: ∞-groupoid $\equiv$ topological space?)

Homotopy type theory (HoTT) = viewpoint of intensional TT, types have inherent higher structure. This allows powerful univalence axiom (equivalence $\approx$ equality), which is incompatible with UIP.

Models of intensional type theory

Since types have inherent homotopy structure, must take this structure into consideration.

- simplicial set/cubical set (based on presheaf model) or Quillen-style model category: presentations of ∞-categories, so inherently support homotopy structure
- any Lurie-style $(\infty, 1)$-Grothendieck topos supports intensional TT w/ univalence (Shulman, 2019)