Notes on Type Theory

Apiros3

First Version : Oct 16, 2025
Last Update : – –, 2025

Contents

1 Introduction

- Introduction to Homotopy Type Theory by Egbert Rijke

2 Judgements and Contexts

A mathematical argument consists of a sequence of deductive steps, which can be represented by inference rules of the form

$$\frac{\mathcal{H}_1 \quad \mathcal{H}_2 \quad \dots \quad \mathcal{H}_n}{\mathcal{C}}$$

Inference rules contain a finite list $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_n$ of judgement for the premises (hypothesis) and a single judgement $\mathcal{C}$ for the conclusion.

3 Dependent Type Theory

3.1 Basic Definitions

We outline the basic definitions of MLTT (Martin-Löf Type Theory).

Definition 3.1.1. *MLTT consists of four* ***judgements****:*

1. *A is a well-formed type in context Γ, written*

$$\Gamma \vdash A \text{ type}$$

2. *A and B are judgementally equal types in context Γ, written*

$$\Gamma \vdash A \doteq B \text{ type}$$

3. *a is an element of type A in context Γ, written*

$$\Gamma \vdash a : A$$

4. *a and b are judgementally equal elements of type A in context Γ, written*

$$\Gamma \vdash a \doteq b : A$$

Definition 3.1.2. *A* ***context*** *is a (potentially empty) finite list of variable declarations $x_1 : A_1, x_2 : A_2(x_1), \dots, x_n : A_n(x_1, \dots, x_{n-1})$ such that for each $1 \leq k \leq n$, we can derive the judgement*

$$x_1 : A_1, \dots, x_{k-1} : A_{k-1}(x_1, \dots, x_{k-2}) \vdash A_k(x_1, \dots, x_{k-1}) \text{ type}$$

using the inference rules of type theory. Variable names can be chosen freely so long as there is no overlap.

Definition 3.1.3. *Consider a type A in context Γ. A* ***family of types*** *over A in context Γ is a type $B(x)$ in context $\Gamma, x : A$. That is, when*

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

we say that B is a family of types over A in context Γ. Alternatively, $B(x)$ is a type ***indexed*** *by $x : A$ in context Γ.*

Definition 3.1.4. *Let B be a type family over A in context Γ. A **section** of the family B over A in context Γ is an element of type $B(x)$ in context $\Gamma, x : A$, as in the judgement*

$$\Gamma, x : A \vdash b(x) : B(x)$$

Then, we say that b is a section of the family B over A in context Γ. Alternatively, $b(x)$ is the element of type $B(x)$ indexed by $x : A$ in context Γ.

Definition 3.1.5. *The inference rules for the formation of contexts, types, and their elements are given by,*

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

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash A\ \text{type}} \qquad \frac{\Gamma \vdash a \doteq b : A}{\Gamma \vdash a : A} \qquad \frac{\Gamma \vdash a \doteq b : A}{\Gamma \vdash b : A}$$

Definition 3.1.6. *The inference rules that postulate judgemental equality as an equivalence relation is given as follows:*

$$\frac{\Gamma \vdash A\ \text{type}}{\Gamma \vdash A \doteq A\ \text{type}} \qquad \frac{\Gamma \vdash B \doteq A\ \text{type}}{\gamma \vdash A \doteq B\ \text{type}} \qquad \frac{\Gamma \vdash A \doteq B\ \text{type} \quad \Gamma \vdash B \doteq C\ \text{type}}{\Gamma \vdash A \doteq C\ \text{type}}$$

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash a \doteq a : A} \qquad \frac{\Gamma \vdash a \doteq b : A}{\Gamma \vdash b \doteq a : A} \qquad \frac{\Gamma \vdash a \doteq b : A \quad \Gamma \vdash b \doteq c : A}{\Gamma \vdash a \doteq c : A}$$

4 Universes

A

5 Propositional Truncation

6 Algebra 'Zoo'

6.0.1 Algebra

The computation for the types of algebras is given by the standard interpretation into the `Set` universe. The operations are defined inductively with notation $-^A$, overloaded for `Con`, `Ty`, and `Tm` as follows:

$$\begin{aligned}
-^A &: \mathtt{Ty} \to \mathtt{Set} \to \mathtt{Set} \\
\iota^A &\equiv X \\
(\iota \to A)^A \quad X &\equiv X \to A^A\ X
\end{aligned}$$

$$\begin{aligned}
(-)^A &: \mathrm{Con} \to \mathbf{Set} \to \mathbf{Set} \\
\Gamma^A X &\equiv \{A : \mathrm{Ty}\} \to \mathrm{Var}\,\Gamma\,A \to A^A X \\
(-)^A &: \mathrm{Tm}\,\Gamma\,A \to \{X : \mathbf{Set}\} \to \Gamma^A X \to A^A X \\
(\mathrm{var}\,x)^A\,\gamma &\equiv \gamma\,x \\
(\mathrm{app}\ t\ u)^A\,\gamma &\equiv t^A\,\gamma\,(u^A\,\gamma) \\
(-)^A &: \mathrm{Sub}\,\Gamma\,\Delta \to \{X : \mathbf{Set}\} \to \Gamma^A X \to \Delta^A X \\
\sigma^A\,\gamma\,x &\equiv (\sigma\,x)^A\,\gamma
\end{aligned}$$

7 Other (move later)

7.1 Intrinsic vs Extrinsic

Intrinsic constructions give sanity lemmas for free, by only allowing well-typed terms to be constructed. Here we give an intrinsic encoding of the lambda calculus such that only well-typed terms can be constructed. The typing rules become the constructors of the term datatype. We use De Bruijn indices (as we don't care about the implementation here).

The object types are given as

```
data Ty : Set where
    ι     : Ty
    _⇒_ : Ty → Ty → Ty
```

where we can define optional types as needed.

A context is a list of types that looks like

```
data Ctx : Set where
    ϵ   : Ctx
    _▷_ : Ctx → Ty → Ctx

data Var : Ctx → Ty → Set where
    vz : ∀ {Γ A}   → Var (Γ ▷ A) A              -- top
    vs : ∀ {Γ A B} → Var Γ A → Var (Γ ▷ B) A    -- weaken
```

Then `Var Γ A` is inhabited exactly when `A` occurs in Γ at the right index.

Then intrinsic terms are given by

```
data Tm : Ctx → Ty → Set where
    var : ∀ {Γ A} → Var Γ A → Tm Γ A
    lam : ∀ {Γ A B} → Tm (Γ ▷ A) B → Tm Γ (A ⇒ B)
    app : ∀ {Γ A B} → Tm Γ (A ⇒ B) → Tm Γ A → Tm Γ B
```

7.2 Theory of Signatures

Definition 7.2.1. *The **theory of signatures** or **ToS** is a small type theory equipped with:*

- *An empty base type ι*
- *A first-order function type $\iota \to -$ (with application but no λ-abstraction)*

The full syntax of this theory can be specified inductively as follows:

$$
\begin{array}{lcl}
\mathrm{Ty} & : & \mathrm{Set} \\
\iota & : & \mathrm{Ty} \\
\iota \to \_ & : & \mathrm{Ty} \to \mathrm{Ty}
\end{array}
\qquad
\begin{array}{lcl}
\mathrm{Var} & : & \mathrm{Con} \to \mathrm{Ty} \to \mathrm{Set} \\
\mathrm{vz} & : & \mathrm{Var}\,(\Gamma \triangleright A)\,A \\
\mathrm{vs} & : & \mathrm{Var}\,\Gamma\,A \to \mathrm{Var}\,(\Gamma \triangleright B)\,A
\end{array}
$$

$$
\begin{array}{lcl}
\mathrm{Con} & : & \mathrm{Set} \\
\bullet & : & \mathrm{Con} \\
\_ \triangleright \_ & : & \mathrm{Con} \to \mathrm{Ty} \to \mathrm{Con}
\end{array}
\qquad
\begin{array}{lcl}
\mathrm{Tm} & : & \mathrm{Con} \to \mathrm{Ty} \to \mathrm{Set} \\
\mathrm{var} & : & \mathrm{Var}\,\Gamma\,A \to \mathrm{Tm}\,\Gamma\,A \\
\mathrm{app} & : & \mathrm{Tm}\,\Gamma\,(\iota \to A) \to \mathrm{Tm}\,\Gamma\,\iota \to \mathrm{Tm}\,\Gamma\,A
\end{array}
$$