

Foundations - type theory

November 5, 2025

Contents

1	Deductive system	1
1.1	Contexts	1
1.2	Judgemental equalities	2
1.3	Type universe	2
2	Type formers	2
2.1	Function type	2
2.2	Dependent function type	3
2.3	Dependent pair type	3
2.4	Coproduct type	3
2.5	Empty type	4
2.6	Unit type	4

1 Deductive system

1.1 Contexts

Here are the basic principles for contexts in type theory.

Definition 0.1.1. We will note $.$ for the empty context

Axiom 0.1.1. empty derivation :

$rule_empty_derivation : \epsilon \longrightarrow empty_context\ ctx$

Axiom 0.1.2. context extension :

$rule_context_extension : \Gamma \vdash A : \mathcal{U}_i, \quad x \text{ not in } dom(\Gamma) \longrightarrow \Gamma, x : A\ ctx$

Axiom 0.1.3. variable rule : $rule_variable : \Gamma\ ctx, \quad x : A \text{ in } \Gamma \longrightarrow \Gamma \vdash x : A$

1.2 Judgemental equalities

We suppose that judgmental equality is an equivalence relation. Namely:

Axiom 0.2.1. Judgmental reflexivity :

$$rule_judg_refl : \Gamma \vdash a : A \text{ --- } \Gamma \vdash a \equiv a : A$$

Axiom 0.2.2. Judgmental symmetry :

$$rule_judg_sym : \Gamma \vdash a \equiv b : A \text{ --- } \Gamma \vdash b \equiv a : A$$

Axiom 0.2.3. Judgmental transitivity :

$$rule_judg_trans : \Gamma \vdash a \equiv b : A, \quad \Gamma \vdash b \equiv c : A \text{ --- } \Gamma \vdash a \equiv c : A$$

Axiom 0.2.4. Judgmental typing :

$$rule_judg_typing : \Gamma \vdash a : A, \quad \Gamma \vdash A \equiv B : \mathcal{U}_i \text{ --- } \Gamma \vdash a : B$$

Axiom 0.2.5. Judgmental typing equiv :

$$rule_judg_typing_eq : \Gamma \vdash a \equiv b : A, \quad \Gamma \vdash A \equiv B : \mathcal{U}_i \text{ --- } \Gamma \vdash a \equiv b : B$$

1.3 Type universe

Axiom 0.3.1. Universe intro :

$$rule_univ_intro : \Gamma \text{ ctx } \text{ --- } \Gamma \vdash \mathcal{U}_i : \mathcal{U}_{i+1}$$

Axiom 0.3.2. Universe cumul :

$$rule_univ_cumul : \Gamma \vdash A : \mathcal{U}_i \text{ --- } \Gamma \vdash A : \mathcal{U}_{i+1}$$

2 Type formers

2.1 Function type

Axiom 1.1.1. Function formation :

$$rule_lambda - form : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma, a : A \vdash B : \mathcal{U}_i \text{ --- } \Gamma \vdash A \rightarrow B : \mathcal{U}_i$$

Axiom 1.1.2. Function introduction :

$$rule_lambda - intro : \Gamma, a : A \vdash B : \mathcal{U}_i \text{ --- } \Gamma \vdash \lambda(x : A).b : A \rightarrow B$$

Axiom 1.1.3. Function elimination :

$$rule_lambda - elim : \Gamma \vdash f : A \rightarrow B, \quad \Gamma \vdash a : A \text{ --- } \Gamma \vdash f(a) : B$$

Axiom 1.1.4. Function computation :

$$rule_lambda - comp : \Gamma, x : A \vdash b : B, \quad \Gamma \vdash a : A \text{ --- } \Gamma \vdash \text{let } x = a \text{ in } \lambda(x : A).b : B$$

Axiom 1.1.5. Function uniqueness principle :

$$rule_lambda - uniq : \Gamma \vdash f : A \rightarrow B \text{ --- } \Gamma \vdash f \equiv \lambda(x : A).f(x) : A \rightarrow B$$

2.2 Dependent function type

Similar as non dependent functions.

Axiom 1.2.1. Dependent function formation :

$$rule_{\Pi} - form : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma, a : A \vdash B : \mathcal{U}_i \longrightarrow \Gamma \vdash \Pi_{x:A} B(x) : \mathcal{U}_i$$

Axiom 1.2.2. Dependent function introduction :

$$rule_{\Pi} - intro : \Gamma, a : A \vdash B : \mathcal{U}_i \longrightarrow \Gamma \vdash \lambda(x : A).b : \Pi_{x:A} B(x)$$

Axiom 1.2.3. Dependent function elimination :

$$rule_{\Pi} - elim : \Gamma \vdash f : \Pi_{x:A} B, \quad \Gamma \vdash a : A \longrightarrow \Gamma \vdash f(a) : B(a)$$

Axiom 1.2.4. Dependent function computation :

$$rule_{\Pi} - comp : \Gamma, x : A \vdash b : B, \quad \Gamma \vdash a : A \longrightarrow \Gamma \vdash \text{let } x = a \text{ in } \lambda(x : A).b : B(x)$$

Axiom 1.2.5. Dependent function uniqueness principle :

$$rule_{\Pi} - uniq : \Gamma \vdash f : \Pi_{x:A} B \longrightarrow \Gamma \vdash f \equiv \lambda(x : A).f(x) : \Pi_{x:A} B$$

2.3 Dependent pair type

Axiom 1.3.1. Dependent pair formation :

$$rule_{\Sigma} - form : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma, a : A \vdash B : \mathcal{U}_i \longrightarrow \Gamma \vdash \Sigma_{x:A} B(x) : \mathcal{U}_i$$

Axiom 1.3.2. Dependent pair introduction :

$$rule_{\Sigma} - intro : \Gamma, a : A \vdash B : \mathcal{U}_i, \quad \Gamma \vdash a : A, \quad \Gamma \vdash b : B(x) \longrightarrow \Gamma \vdash \text{pair}(a, b) : \Sigma_{y:A} B(y)$$

Axiom 1.3.3. Dependent pair elimination :

$$rule_{\Sigma} - elim : \Gamma, z : \Sigma_{x:A} B(x) \vdash C : \mathcal{U}_i, \quad \Gamma, a : A, b : B \vdash g : C(\text{pair}(a, b)), \quad \Gamma \vdash p : \Sigma_{y:A} B(y) \longrightarrow \Gamma \vdash \text{ind}(C, g, p) : C(z)$$

Axiom 1.3.4. Dependent pair computation :

$$rule_{\Sigma} - comp : \Gamma, z : \Sigma_{x:A} B(x) \vdash C : \mathcal{U}_i, \quad \Gamma, a : A, b : B \vdash g : C(z), \quad \Gamma \vdash a : A, \quad \Gamma \vdash b : B(x) \longrightarrow \Gamma \vdash \text{let } x = a, y = b, z = \text{pair}(a, b) \text{ in } \text{ind}(C, g, \text{pair}(a, b)) \equiv g(x, y) : C(z)$$

2.4 Coproduct type

Axiom 1.4.1. Coproduct formation :

$$rule_{\text{coproduct}} - form : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma \vdash B : \mathcal{U}_i \longrightarrow \Gamma \vdash A + B : \mathcal{U}_i$$

Axiom 1.4.2. Coproduct introduction left :

$$rule_{\text{coproduct}} - intro1 : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma \vdash B : \mathcal{U}_i, \quad \Gamma \vdash a : A \longrightarrow \Gamma \vdash \text{inl}(a) : A + B$$

Axiom 1.4.3. Coproduct introduction right :

$$rule_{\text{coproduct}} - intro2 : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma \vdash B : \mathcal{U}_i, \quad \Gamma \vdash b : B \longrightarrow \Gamma \vdash \text{inr}(b) : A + B$$

Axiom 1.4.4. Coproduct elimination :

$rule_coproduct - elim : \Gamma, z : A + B \vdash C : \mathcal{U}_i, \quad \Gamma, a : A \vdash c : C(inl(a)), \quad \Gamma, b : B \vdash d : C(inr(b)), \quad \Gamma \vdash e : A + B \longrightarrow \Gamma \vdash ind(C, c(a), d(b), e) : C(e)$

Axiom 1.4.5. Coproduct computation left :

$rule_coproduct - comp1 : \Gamma, z : A + B \vdash C : \mathcal{U}_i, \quad \Gamma, a : A \vdash c : C(inl(a)), \quad \Gamma, b : B \vdash d : C(inr(b)), \quad \Gamma \vdash x : A \longrightarrow \Gamma \vdash let\ x = a, z = inl(a)\ in\ ind(C, c, d, inl(a)) \equiv c(x) : C(z)$

Axiom 1.4.6. Coproduct computation right :

$rule_coproduct - comp1 : \Gamma, z : A + B \vdash C : \mathcal{U}_i, \quad \Gamma, a : A \vdash c : C(inl(a)), \quad \Gamma, b : B \vdash d : C(inr(b)), \quad \Gamma \vdash y : B \longrightarrow \Gamma \vdash let\ y = b, z = inr(b)\ in\ ind(C, c, d, inr(b)) \equiv d(y) : C(z)$

2.5 Empty type

Axiom 1.5.1. Empty formation :

$rule_empty - form : \Gamma\ ctx \longrightarrow \Gamma \vdash 0 : \mathcal{U}_i$

Axiom 1.5.2. Empty elimination :

$rule_empty - elim : \Gamma, x : 0 \vdash C : \mathcal{U}_i, \quad \Gamma \vdash a : 0 \longrightarrow \Gamma \vdash absurd\ a : C(a)$

2.6 Unit type

Axiom 1.6.1. Unit formation :

$rule_unit - form : \Gamma\ ctx \longrightarrow \Gamma \vdash 1 : \mathcal{U}_i$

Axiom 1.6.2. Unit introduction :

$rule_unit - intro : \Gamma\ ctx \longrightarrow \Gamma \vdash \star : 1$

Axiom 1.6.3. Unit elimination :

$rule_unit - elim : \Gamma, x : 1 \vdash C : \mathcal{U}_i, \quad \Gamma \vdash c : C(\star), \quad \Gamma \vdash a : 1 \longrightarrow \Gamma \vdash ind(C, c, a) : C(a)$

Axiom 1.6.4. Unit computation :

$rule_unit - comp : \Gamma, x : 1 \vdash C : \mathcal{U}_i, \quad \Gamma \vdash c : C(\star) \longrightarrow \Gamma \vdash let\ x = \star\ in\ ind(C, c, \star) \equiv c : C(x)$