

Foundations - type theory

November 6, 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
2.7	Natural numbers	4
2.8	Identity types	5

1 Deductive system

1.1 Contexts

Here are the basic principles for contexts in type theory.

Definition 0.1.1. We will note \cdot for the empty context

Axiom 0.1.1. empty derivation :

$$\text{rule_empty_derivation} : \epsilon \longrightarrow \text{empty_context ctx}$$

Axiom 0.1.2. context extension :

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

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

1.2 Judgemental equalities

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

Axiom 0.2.1. Judgmental reflexivity :

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

Axiom 0.2.2. Judgmental symmetry :

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

Axiom 0.2.3. Judgmental transitivity :

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

Axiom 0.2.4. Judgmental typing :

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

Axiom 0.2.5. Judgmental typing equiv :

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

1.3 Type universe

Axiom 0.3.1. Universe intro :

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

Axiom 0.3.2. Universe cumul :

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

2 Type formers

2.1 Function type

Axiom 1.1.1. Function formation :

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

Axiom 1.1.2. Function introduction :

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

Axiom 1.1.3. Function elimination :

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

Axiom 1.1.4. Function computation :

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

Axiom 1.1.5. Function uniqueness principle :

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

2.2 Dependent function type

Similar as non dependent functions.

Axiom 1.2.1. Dependent function formation :

$$\text{rule}_{\Pi-form} : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma, a : A \vdash B : \mathcal{U}_i \implies \Gamma \vdash \Pi_{x:A} B(x) : \mathcal{U}_i$$

Axiom 1.2.2. Dependent function introduction :

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

Axiom 1.2.3. Dependent function elimination :

$$\text{rule}_{\Pi-elim} : \Gamma \vdash f : \Pi_{x:A} B, \quad \Gamma \vdash a : A \implies \Gamma \vdash f(a) : B(a)$$

Axiom 1.2.4. Dependent function computation :

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

Axiom 1.2.5. Dependent function uniqueness principle :

$$\text{rule}_{\Pi-uniq} : \Gamma \vdash f : \Pi_{x:A} B \implies \Gamma \vdash f \equiv \lambda(x : A).f(x) : \Pi_{x:A} B$$

2.3 Dependent pair type

Axiom 1.3.1. Dependent pair formation :

$$\text{rule}_{\Sigma-form} : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma, a : A \vdash B : \mathcal{U}_i \implies \Gamma \vdash \Sigma_{x:A} B(x) : \mathcal{U}_i$$

Axiom 1.3.2. Dependent pair introduction :

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

Axiom 1.3.3. Dependent pair elimination :

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

Axiom 1.3.4. Dependent pair computation :

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

2.4 Coproduct type

Axiom 1.4.1. Coproduct formation :

$$\text{rule}_{\text{coproduct-form}} : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma \vdash B : \mathcal{U}_i \implies \Gamma \vdash A + B : \mathcal{U}_i$$

Axiom 1.4.2. Coproduct introduction left :

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

Axiom 1.4.3. Coproduct introduction right :

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

Axiom 1.4.4. Coproduct elimination :

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

Axiom 1.4.5. Coproduct computation left :

$$\text{rule_coproduct_comp1} : \Gamma, z : A + B \vdash C : \mathcal{U}_i, \quad \Gamma, a : A \vdash c : C(\text{inl}(a)), \quad \Gamma, b : B \vdash d : C(\text{inr}(b)), \quad \Gamma \vdash x : A \longrightarrow \Gamma \vdash \text{let } x = a, z = \text{inl}(a) \text{ in } \text{ind}(C, c, d, \text{inl}(a)) \equiv c(x) : C(z)$$

Axiom 1.4.6. Coproduct computation right :

$$\text{rule_coproduct_comp1} : \Gamma, z : A + B \vdash C : \mathcal{U}_i, \quad \Gamma, a : A \vdash c : C(\text{inl}(a)), \quad \Gamma, b : B \vdash d : C(\text{inr}(b)), \quad \Gamma \vdash y : B \longrightarrow \Gamma \vdash \text{let } y = b, z = \text{inr}(b) \text{ in } \text{ind}(C, c, d, \text{inr}(b)) \equiv d(y) : C(z)$$

2.5 Empty type

Axiom 1.5.1. Empty formation :

$$\text{rule_empty_form} : \Gamma \text{ ctx} \longrightarrow \Gamma \vdash 0 : \mathcal{U}_i$$

Axiom 1.5.2. Empty elimination :

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

2.6 Unit type

Axiom 1.6.1. Unit formation :

$$\text{rule_unit_form} : \Gamma \text{ ctx} \longrightarrow \Gamma \vdash 1 : \mathcal{U}_i$$

Axiom 1.6.2. Unit introduction :

$$\text{rule_unit_intro} : \Gamma \text{ ctx} \longrightarrow \Gamma \vdash \star : 1$$

Axiom 1.6.3. Unit elimination :

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

Axiom 1.6.4. Unit computation :

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

2.7 Natural numbers

Axiom 1.7.1. Natural numbers formation :

$$\text{rule_N_form} : \Gamma \text{ ctx} \longrightarrow \Gamma \vdash \mathbb{N} : \mathcal{U}_i$$

Axiom 1.7.2. Natural numbers introduction : $0 :$

$$\text{rule_N_intro_0} : \Gamma \text{ ctx} \longrightarrow \Gamma \vdash 0 : \mathbb{N}$$

Axiom 1.7.3. Natural numbers introduction : $\text{succ} :$

$$\text{rule_N_intro_succ} : \Gamma \vdash n : \mathbb{N} \longrightarrow \Gamma \vdash \text{succ}(n) : \mathbb{N}$$

Axiom 1.7.4. Natural numbers elimination :

$$\text{rule_N_elim} : \Gamma, x : \mathbb{N} \vdash C : \mathcal{U}_i, \quad \Gamma \vdash c_0 : C(0), \quad \Gamma, x : \mathbb{N}, y : C \vdash c_s : C(\text{succ}(x)), \quad \Gamma \vdash n : \mathbb{N} \implies \Gamma \vdash \text{ind}(C, c_0, c_s, n) : C(n)$$

Axiom 1.7.5. Natural numbers computation : 0 :

$$\text{rule_N_comp_0} : \Gamma, x : \mathbb{N} \vdash C : \mathcal{U}_i, \quad \Gamma \vdash c_0 : C(0), \quad \Gamma, x : \mathbb{N}, y : C \vdash c_s : C(\text{succ}(x)) \implies \Gamma \vdash \text{let } x = 0 \text{ in } \text{ind}(C, c_0, c_s, 0) \equiv c_0 : C(x)$$

Axiom 1.7.6. Natural numbers computation : succ :

$$\text{rule_N_comp_succ} : \Gamma, x : \mathbb{N} \vdash C : \mathcal{U}_i, \quad \Gamma \vdash c_0 : C(0), \quad \Gamma, x : \mathbb{N}, y : C \vdash c_s : C(\text{succ}(x)), \quad \Gamma \vdash n : \mathbb{N} \implies \Gamma \vdash \text{let } x = n, x' = \text{succ}(n), y = \text{ind}(C, c_0, c_s, n) \text{ in } \text{ind}(C, c_0, c_s, \text{succ}(n)) \equiv c_s(x, y) : C(x')$$

2.8 Identity types

Axiom 1.8.1. Identity type formation :

$$\text{rule_id_form} : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma \vdash a : A, \quad \Gamma \vdash b : A \implies \Gamma \vdash a =_A b : \mathcal{U}_i$$

Axiom 1.8.2. Identity type introduction :

$$\text{rule_id_intro} : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma \vdash a : A \implies \Gamma \vdash \text{refl}_a : a =_A a$$

Axiom 1.8.3. Identity type elimination :

$$\text{rule_id_elim} : \Gamma, a : A, b : A, p : a =_A b \vdash C : \mathcal{U}_i, \quad \Gamma, z : A \vdash c : C(z, z, \text{refl}_z), \quad \Gamma \vdash a : A, \quad \Gamma \vdash b : A, \quad \Gamma \vdash p' : a =_A b \implies \Gamma \vdash \text{ind}(C, z, a, b, p') : C(a, b, p')$$

Axiom 1.8.4. Identity type computation :

$$\text{rule_id_comp} : \Gamma, a : A, b : A, p : a =_A b \vdash C : \mathcal{U}_i, \quad \Gamma, z : A \vdash c : C(z, z, \text{refl}_z), \quad \Gamma \vdash a : A \implies \Gamma \vdash \text{let } x = a, y = a, z = a, p = \text{refl}_a \text{ in } \text{ind}(C, z, a, a, \text{refl}_a) \equiv c(z) : C(x, y, p)$$