

Foundations - type theory

November 5, 2025

Contents

1	Deductive system	1
1.1	Contexts	1
1.2	Judgemental equalities	1
1.3	Type universe	2
2	Type formers	2
2.1	Function type	2
2.2	Dependent function type	2
2.3	Dependent pair type	3

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 judgmental 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 :

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

Axiom 1.2.2. Dependent function introduction :

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

Axiom 1.2.3. Dependent function elimination :

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

Axiom 1.2.4. Dependent function computation :

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

Axiom 1.2.5. Dependent function uniqueness principle :

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

2.3 Dependent pair type

Axiom 1.3.1. Dependent pair formation :

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

Axiom 1.3.2. Dependent pair introduction :

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

Axiom 1.3.3. Dependent pair elimination :

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

Axiom 1.3.4. Dependent pair computation :

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