Foundations - type theory

November 5, 2025

Contents

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 :
$rule\_empty\_derivation : \epsilon$ —— $empty\_context\ ctx$

Axiom 0.1.2. context extension :
$rule\_context\_extension : \Gamma \vdash A : \mathcal{U}_i, \quad x\ not\ in\ dom(\Gamma)$ —— $\Gamma,\ x : A\ ctx$

Axiom 0.1.3. variable rule : $rule\_variable : \Gamma\ ctx, \quad x : A\ in\ \Gamma$ —— $\Gamma \vdash x : A$

1.2 Judgemental equalities

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

Axiom 0.2.1. Judgmental reflexivity :
$rule\_judg\_refl : \Gamma \vdash a : A$ —— $\Gamma \vdash a \equiv a : A$

Axiom 0.2.2. Judgmental symmetry :
$rule\_judg\_sym : \Gamma \vdash a \equiv b : A$ —— $\Gamma \vdash b \equiv a : A$

Axiom 0.2.3. Judgmental transitivity :
$rule\_judg\_trans : \Gamma \vdash a \equiv b : A, \quad \Gamma \vdash b \equiv c : A$ —— $\Gamma \vdash a \equiv c : A$

Axiom 0.2.4. Judgmental typing :
$rule\_judg\_typing : \Gamma \vdash a : A, \quad \Gamma \vdash A \equiv B : \mathcal{U}_i$ —— $\Gamma \vdash a : B$

Axiom 0.2.5. Judgmental typing equiv :
$rule\_judg\_typing\_eq : \Gamma \vdash a \equiv b : A, \quad \Gamma \vdash A \equiv B : \mathcal{U}_i$ —— $\Gamma \vdash a \equiv b : B$

1.3 Type universe

Axiom 0.3.1. Universe intro :
$rule\_univ\_intro : \Gamma\ ctx$ —— $\Gamma \vdash \mathcal{U}_i : \mathcal{U}_{i+1}$

Axiom 0.3.2. Universe cumul :
$rule\_univ\_cumul : \Gamma \vdash A : \mathcal{U}_i$ —— $\Gamma \vdash A : \mathcal{U}_{i+1}$

2 Type formers

2.1 Function type

Axiom 1.1.1. Function formation :
$rule\_\lambda - form : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma, a : A \vdash B : \mathcal{U}_i$ —— $\Gamma \vdash A \to B : \mathcal{U}_i$

Axiom 1.1.2. Function introduction :
$rule\_\lambda - intro : \Gamma, a : A \vdash B : \mathcal{U}_i$ —— $\Gamma \vdash \lambda(x : A).b : A \to B$

Axiom 1.1.3. Function elimination :
$rule\_\lambda - elim : \Gamma \vdash f : A \to B, \quad \Gamma \vdash a : A$ —— $\Gamma \vdash f(a) : B$

Axiom 1.1.4. Function computation :
$rule\_\lambda - comp : \Gamma, x : A \vdash b : B, \quad \Gamma \vdash a : A$ —— $\Gamma \vdash \text{let } x = a \text{ in } \lambda(x : A).b : B$

Axiom 1.1.5. Function uniqueness principle :
$rule\_\lambda - uniq : \Gamma \vdash f : A \to B$ —— $\Gamma \vdash f \equiv \lambda(x : A).f(x) : A \to B$

2.2 Dependent function type

Similar as non dependent functions.

Axiom 1.2.1. Dependent function formation :
$rule\_\Pi - form : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma, a : A \vdash B : \mathcal{U}_i$ —— $\Gamma \vdash \Pi_{x:A} B(x) : \mathcal{U}_i$

Axiom 1.2.2. Dependent function introduction :
$rule\_\Pi - intro : \Gamma, a : A \vdash B : \mathcal{U}_i$ —— $\Gamma \vdash \lambda(x : A).b : \Pi_{x:A} B(x)$

Axiom 1.2.3. Dependent function elimination :
$rule\_\Pi - elim : \Gamma \vdash f : \Pi_{x:A} B, \quad \Gamma \vdash a : A$ —— $\Gamma \vdash f(a) : B(a)$

Axiom 1.2.4. Dependent function computation :
$rule\_\Pi - comp : \Gamma, x : A \vdash b : B, \quad \Gamma \vdash a : A$ —— $\Gamma \vdash \text{let } x = a \text{ in } \lambda(x : A).b : B(x)$

Axiom 1.2.5. Dependent function uniqueness principle :
$rule\_\Pi - uniq : \Gamma \vdash f : \Pi_{x:A} B$ —— $\Gamma \vdash f \equiv \lambda(x : A).f(x) : \Pi_{x:A} B$

2.3 Dependent pair type

Axiom 1.3.1. Dependent pair formation :
$rule\_\Sigma - form : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma, a : A \vdash B : \mathcal{U}_i$ —— $\Gamma \vdash \Sigma_{x:A} B(x) : \mathcal{U}_i$

Axiom 1.3.2. Dependent pair introduction :
$rule\_\Sigma - intro : \Gamma, a : A \vdash B : \mathcal{U}_i, \quad \Gamma \vdash a : A, \quad \Gamma \vdash b : B(x)$ —— $\Gamma \vdash pair(a, b) : \Sigma_{y:A} B(y)$

Axiom 1.3.3. Dependent pair elimination :
$rule\_\Sigma - elim : \Gamma, z : \Sigma_{x:A} B(x) \vdash C : \mathcal{U}_i, \quad \Gamma, a : A, b : B \vdash g : C(pair(a, b)), \quad \Gamma \vdash p : \Sigma_{y:A} B(y)$ ——$\Gamma \vdash ind(C, g, p) : C(z)$

Axiom 1.3.4. Dependent pair computation :
$rule\_\Sigma - comp : \Gamma, z : \Sigma_{x:A} B(x) \vdash C : \mathcal{U}_i, \quad \Gamma, a : A, b : B \vdash g : C(z), \quad \Gamma \vdash a : A, \quad \Gamma \vdash b : B(x)$ —— $\Gamma \vdash \text{let } x = a, y = b, z = pair(a, b) \text{ in } ind(C, g, pair(a, b)) \equiv g(x, y) : C(z)$

2.4 Coproduct type

Axiom 1.4.1. Coproduct formation :
$rule\_coproduct - form : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma \vdash B : \mathcal{U}_i$ —— $\Gamma \vdash A + B : \mathcal{U}_i$

Axiom 1.4.2. Coproduct introduction left :
$rule\_coproduct - intro1 : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma \vdash B : \mathcal{U}_i, \quad \Gamma \vdash a : A$ —— $\Gamma \vdash inl(a) : A + B$

Axiom 1.4.3. Coproduct introduction right :
$rule\_coproduct - intro2 : \Gamma \vdash A : \mathcal{U}_i, \quad \Gamma \vdash B : \mathcal{U}_i, \quad \Gamma \vdash b : B$ —— $\Gamma \vdash inr(b) : A + B$

Axiom 1.4.4. Coproduct elimination :
$rule\_coproduct - elim : \Gamma, z : A + B \vdash C : \mathcal{U}_i, \quad \Gamma, a : A \vdash c : C(inl(a)), \quad \Gamma, b : B \vdash d : C(inr(b)), \quad \Gamma \vdash e : A + B$ —— $\Gamma \vdash ind(C, c(a), d(b), e) : C(e)$

Axiom 1.4.5. Coproduct computation left :
$rule\_coproduct - comp1 : \Gamma, z : A + B \vdash C : \mathcal{U}_i, \quad \Gamma, a : A \vdash c : C(inl(a)), \quad \Gamma, b : B \vdash d : C(inr(b)), \quad \Gamma \vdash x : A$ —— $\Gamma \vdash \text{let } x = a, z = inl(a) \text{ in } ind(C, c, d, inl(a)) \equiv c(x) : C(z)$

Axiom 1.4.6. Coproduct computation right :
$rule\_coproduct - comp1 : \Gamma, z : A + B \vdash C : \mathcal{U}_i, \quad \Gamma, a : A \vdash c : C(inl(a)), \quad \Gamma, b : B \vdash d : C(inr(b)), \quad \Gamma \vdash y : B$ —— $\Gamma \vdash \text{let } y = b, z = inr(b) \text{ in } ind(C, c, d, inr(b)) \equiv d(y) : C(z)$