# A New Coq Formalisation of Classical First-Order Logic with Proofs of the Soundness and Completeness Theorems

Kijeong Lim

Chonnam National University

Korea Logic Day
January 14, 2025

# Table of Contents

# Introduction

**The Motivation.**

- It has been a common practice in software verification to rely on proof assistants, such as Coq and Isabelle, to obtain formal proofs of program correctness. In these systems, programs are abstracted as mathematical entities, and their behaviour is denoted by certain mathematical properties.
- To reason about the behaviour of a program effectively using a proof assistant, there must be a well-developed theory about the mathematical properties being verified. For example, Software Foundations Lab formalised a theory of ordinal numbers, which is a purely mathematical concept, to state and/or prove the termination of a given program.
- However, most proof assistants do not adopt the usual first-order language used by working mathematicians. Instead, they take dependent type theories as their foundation. Therefore, to facilitate the translation of mathematical theories into a proof assistant, an intermediate language is deemed necessary.

# Introduction

**The Advantages of Embedding First-Order Logic into "Coq".**

- **CIC** serves as a very rich metalanguage. In contrast, the Fitch system has poor expressiveness, which limits its ability to state and/or prove meta-theorems. For example, it cannot directly state whether an object theory $\mathcal{T}$ of arithmetic admits the $\omega$-rule, whereas Coq can.

$$\frac{(\forall n \in \mathbb{N})(\mathcal{T} \vdash P(\bar{n}))}{\mathcal{T} \vdash \dot{\forall} x\, P(x)}\ \omega\text{-rule}$$

- One can leverage various libraries of the Coq community when formalising an embedded first-order theory. For instance, if a first-order theory $\mathcal{T}$ is complete—that is, $\mathcal{T}$ proves either an arbitrary proposition or its negation—the non-existence of a model for the negation of a proposition $\varphi$ implies that $\varphi$ is a theorem of $\mathcal{T}$, where the non-existence may be shown with the libraries in Coq.
- Furthermore, any sentence $\varphi$ proved in an embedded first-order theory $\mathcal{T}$ can be lifted to its corresponding theorem, which can be directly used in Coq. If $\mathfrak{M}$ is a model of $\mathcal{T}$, then it immediately becomes a theorem in Coq that $\varphi$ holds for $\mathfrak{M}$.

# Introduction

**Formalisation Overview.**

1. **Syntax.** How to use symbols to define logical expressions.
2. **Semantics.** What is the meaning of a logical expression.
3. **Deduction System.** What formulae are provable.
4. **Meta-theory.** The theory on first-order logic.

# Formalisation (Syntax)

A ***first-order language*** $L$ is represented as a record with the following fields:

- a **Set** $\mathcal{F}$ of ***function symbols***;
- a **Set** $\mathcal{C}$ of ***constant symbols***;
- a **Set** $\mathcal{R}$ of ***relation symbols***;
- a table mapping each $f \in \mathcal{F}$ to its arity $n_f \in \mathbb{N}$; and
- a table mapping each $R \in \mathcal{R}$ to its arity $n_R \in \mathbb{N}$.

# Formalisation (Syntax)

- An ***individual variable*** $x$ is of the form $v_i$ for a natural number $i$—i.e.,

$$x \ ::= \ v_i \quad \text{for} \quad i \in \mathbb{N}.$$

- An $L$***-term*** $t$ is defined inductively by

$$t \ ::= \ x \mid f\,\vec{t} \mid c$$

  where $f \in \mathcal{F}$, $c \in \mathcal{C}$, and $\vec{t}$ is a vector of $L$-terms.
- An $L$***-formula*** $\varphi$ is defined inductively by

$$\varphi \ ::= \ R\,\vec{t} \mid t_1 \doteq t_2 \mid \dot{\neg}\,\varphi_1 \mid \varphi_1 \mathrel{\dot{\to}} \varphi_2 \mid \dot{\forall} x\,\varphi_1$$

  where $R \in \mathcal{R}$.

**Notation.** For a first-order language $L$,

- the type of $L$-terms is denoted by $\mathtt{trm}\,L$; and
- the type of $L$-formulae is denoted by $\mathtt{frm}\,L$.

# Formalisation (Syntax)

- A simultaneous substitution $\sigma$ is defined as a map $\mathbb{N} \to \mathtt{trm}\, L$.
- $\iota := i \mapsto v_i.\ t/x; \sigma := i \mapsto \textbf{if}\ x = v_i\ \textbf{then}\ t\ \textbf{else}\ \sigma(i).\ t/x := t/x; \iota.$

# Formalisation (Syntax)

- A simultaneous substitution $\sigma$ is defined as a map $\mathbb{N} \to \mathtt{trm}\, L$.
- $\iota := i \mapsto v_i.\ t/x; \sigma := i \mapsto \textbf{if } x = v_i \textbf{ then } t \textbf{ else } \sigma(i).\ t/x := t/x; \iota.$
- The result of applying $\sigma$ to a syntactic object $X$ is denoted by $[\sigma]X$.
- $[\sigma](v_i) := \sigma(i).\ [\sigma](\dot{\forall} x\, \varphi_1) := \textbf{let } y := \chi(\sigma, \dot{\forall} x\, \varphi_1) \textbf{ in } \dot{\forall} y\, ([y/x; \sigma]\varphi_1).$
- $\chi(\sigma, \varphi) := \max\{\max(\mathrm{FV}(\sigma(i))) \mid v_i \in \mathrm{FV}(\varphi)\} + 1.$

# Formalisation (Syntax)

- A simultaneous substitution $\sigma$ is defined as a map $\mathbb{N} \to \mathtt{trm}\, L$.
- $\iota := i \mapsto v_i.\ t/x; \sigma := i \mapsto \textbf{if } x = v_i \textbf{ then } t \textbf{ else } \sigma(i).\ t/x := t/x; \iota.$
- The result of applying $\sigma$ to a syntactic object $X$ is denoted by $[\sigma]X$.
- $[\sigma](v_i) := \sigma(i).\ [\sigma](\dot{\forall} x\, \varphi_1) := \textbf{let } y := \chi(\sigma, \dot{\forall} x\, \varphi_1) \textbf{ in } \dot{\forall} y\, ([y/x; \sigma]\varphi_1).$
- $\chi(\sigma, \varphi) := \max\{\max(\mathrm{FV}(\sigma(i))) \mid v_i \in \mathrm{FV}(\varphi)\} + 1.$
- $\alpha$-equivalence is defined inductively. The constructor for $\dot{\forall}$ is

$$\frac{[y/x_1]\varphi_1 \equiv_\alpha [y/x_2]\varphi_2}{\dot{\forall} x_1\, \varphi_1 \equiv_\alpha \dot{\forall} x_2\, \varphi_2}$$

  provided by $y \notin \mathrm{FV}(\dot{\forall} x_1\, \varphi_1) \wedge y \notin \mathrm{FV}(\dot{\forall} x_2\, \varphi_2)$.

# Formalisation (Syntax)

- A simultaneous substitution $\sigma$ is defined as a map $\mathbb{N} \to \mathtt{trm}\, L$.
- $\iota := i \mapsto v_i.\ t/x; \sigma := i \mapsto \textbf{if } x = v_i \textbf{ then } t \textbf{ else } \sigma(i).\ t/x := t/x; \iota.$
- The result of applying $\sigma$ to a syntactic object $X$ is denoted by $[\sigma]X$.
- $[\sigma](v_i) := \sigma(i).\ [\sigma](\dot\forall x\, \varphi_1) := \textbf{let } y := \chi(\sigma, \dot\forall x\, \varphi_1) \textbf{ in } \dot\forall y\, ([y/x; \sigma]\varphi_1).$
- $\chi(\sigma, \varphi) := \max\{\max(\mathrm{FV}(\sigma(i))) \mid v_i \in \mathrm{FV}(\varphi)\} + 1.$
- $\alpha$-equivalence is defined inductively. The constructor for $\dot\forall$ is

$$\frac{[y/x_1]\varphi_1 \equiv_\alpha [y/x_2]\varphi_2}{\dot\forall x_1\, \varphi_1 \equiv_\alpha \dot\forall x_2\, \varphi_2}$$

  provided by $y \notin \mathrm{FV}(\dot\forall x_1\, \varphi_1) \wedge y \notin \mathrm{FV}(\dot\forall x_2\, \varphi_2)$.
- A singleton substitution $\varphi[x := t]$ is also defined: $\varphi[x := t] \equiv_\alpha [t/x]\varphi$;

  whenever $y \neq x \wedge y \in \mathrm{FV}(t),\ (\dot\forall y\, \varphi)[x := t] = \dot\forall y'\, (\varphi[y := y'][x := t])$

  for some $y' \notin \{x\} \cup \mathrm{FV}(t) \cup \mathrm{FV}(\varphi)$; and $(\dot\forall x\, \varphi)[x := t] = (\dot\forall x\, \varphi)$.

# Formalisation (Semantics)

**Definition.** An *L-**structure*** $\mathfrak{A}$ is a record consisting of the following fields:

- a setoid $(A, \sim_A)$,
- interpretations $f^{\mathfrak{A}} : A^{n_f} \to A$ for each $f \in \mathcal{F}$,
- interpretations $c^{\mathfrak{A}} : A$ for each $c \in \mathcal{C}$, and
- interpretations $R^{\mathfrak{A}} : A^{n_R} \to \mathbf{Prop}$ for each $R \in \mathcal{R}$,

such that $\sim_A$ is compatible with all $f^{\mathfrak{A}}$ and $R^{\mathfrak{A}}$—i.e.,

$$\frac{a_1 \sim_A a'_1 \quad \cdots \quad a_{n_f} \sim_A a'_{n_f}}{f^{\mathfrak{A}}(a_1, \cdots, a_{n_f}) \sim_A f^{\mathfrak{A}}(a'_1, \cdots, a'_{n_f})} \qquad \frac{a_1 \sim_A a'_1 \quad \cdots \quad a_{n_R} \sim_A a'_{n_R}}{R^{\mathfrak{A}}(a_1, \cdots, a_{n_R}) \leftrightarrow R^{\mathfrak{A}}(a'_1, \cdots, a'_{n_R})}$$

—and $A$ is nonempty.
Denote the type $A$ by $|\mathfrak{A}|$, which is called the ***domain of discourse*** of $\mathfrak{A}$.

# Formalisation (Semantics)

***Tarski's definition of truth.*** Let $\rho : \mathbb{N} \to |\mathfrak{A}|$.

- $[\![v_i]\!]^{\mathfrak{A}}_{\rho} := \rho(i)$,
- $[\![f\ \vec{t}]\!]^{\mathfrak{A}}_{\rho} := f^{\mathfrak{A}}\ [\![\vec{t}]\!]^{\mathfrak{A}}_{\rho}$,
- $[\![c]\!]^{\mathfrak{A}}_{\rho} := c^{\mathfrak{A}}$,
- $[\![R\ \vec{t}]\!]^{\mathfrak{A}}_{\rho} := R^{\mathfrak{A}}\ [\![\vec{t}]\!]^{\mathfrak{A}}_{\rho}$,
- $[\![t_1 \doteq t_2]\!]^{\mathfrak{A}}_{\rho} := [\![t_1]\!]^{\mathfrak{A}}_{\rho} \sim_{|\mathfrak{A}|} [\![t_2]\!]^{\mathfrak{A}}_{\rho}$,
- $[\![\dot{\neg}\, \varphi_1]\!]^{\mathfrak{A}}_{\rho} := \neg\, [\![\varphi_1]\!]^{\mathfrak{A}}_{\rho}$,
- $[\![\varphi_1 \mathrel{\dot{\to}} \varphi_2]\!]^{\mathfrak{A}}_{\rho} := [\![\varphi_1]\!]^{\mathfrak{A}}_{\rho} \to [\![\varphi_2]\!]^{\mathfrak{A}}_{\rho}$,
- $[\![\dot{\forall} x\, \varphi_1]\!]^{\mathfrak{A}}_{\rho} := (\forall a \in |\mathfrak{A}|)\, [\![\varphi_1]\!]^{\mathfrak{A}}_{[a/x]\rho}$ where $[a/x]\rho := i \mapsto \begin{cases} a, & \text{if } x = v_i; \\ \rho(i), & \text{otherwise.} \end{cases}$

# Formalisation (Semantics)

**Notation.**

- We say $\rho$ ***satisfies*** $\varphi$ ***in*** $\mathfrak{A}$ when $[\![\varphi]\!]^{\mathfrak{A}}_{\rho}$ holds. Then write $\mathfrak{A} \models \varphi[\rho]$.
- For $\Gamma : \mathtt{frm}\, L \to \mathbf{Prop}$, write $\mathfrak{A} \models \Gamma[\rho]$ when $(\forall \varphi \in \Gamma)(\mathfrak{A} \models \varphi[\rho])$.
- We write $\Gamma \vDash C$ when, for any $L$-structure $\mathfrak{A}$ and any $\rho : \mathbb{N} \to |\mathfrak{A}|$,

$$\text{if } \mathfrak{A} \models \Gamma[\rho] \text{ then } \mathfrak{A} \models C[\rho].$$

**Details.** For a type $A$, the type of subsets of $A$ is defined as $A \to \mathbf{Prop}$—i.e.,

```
#[universes(polymorphic=yes)]
Definition ensemble@{u} (A : Type@{u}) : Type@{u} := A → Prop.
```

Thus, $\Gamma : \mathtt{frm}\, L \to \mathbf{Prop}$ indicates that $\Gamma$ is a set of $L$-formulae.

# Formalisation (Deduction System)

**Definition.** For $\Gamma : \mathtt{frm}\, L \to \mathbf{Prop}$ and $C : \mathtt{frm}\, L$, let the proposition

$$\Gamma \vdash C$$

mean that there exists $\vec{\varphi} : \mathtt{list}\,(\mathtt{frm}\, L)$ such that $\vec{\varphi} \subseteq \Gamma$ and $\mathtt{proof}\ \vec{\varphi}\ C$ is inhabited, where $\mathtt{list}\,(\mathtt{frm}\, L)$ is the type of finite sequences of $L$-formulae.

$$\textbf{Inductive}\ \mathtt{proof} : \mathtt{list}\,(\mathtt{frm}\, L) \to \mathtt{frm}\, L \to \mathbf{Set} := \cdots .$$

- Axm

$$\frac{}{\mathtt{proof}\ [p]\ p}$$

- Mp

$$\frac{\mathtt{proof}\ \vec{\varphi}_1\ (p \mathrel{\dot{\to}} q) \quad \mathtt{proof}\ \vec{\varphi}_2\ p}{\mathtt{proof}\ (\vec{\varphi}_1 +\!\!+ \vec{\varphi}_2)\ q}$$

- Gen

$$\frac{\mathtt{proof}\ \vec{\varphi}\ q}{\mathtt{proof}\ \vec{\varphi}\ (\dot{\forall} x\, q)}$$

  provided by $x \notin \mathrm{FV}(\vec{\varphi})$.

# Formalisation (Deduction System)

The axiom schema for propositional logic.

- `proof` $[]\ (p \dot{\to} (q \dot{\to} p))$
- `proof` $[]\ ((p \dot{\to} (q \dot{\to} r)) \dot{\to} ((p \dot{\to} q) \dot{\to} (p \dot{\to} r)))$
- `proof` $[]\ (((\dot{\neg} q) \dot{\to} (\dot{\neg} p)) \dot{\to} (p \dot{\to} q))$

# Formalisation (Deduction System)

The axiom schema for universal quantifier.

- `proof` [] $((\dot{\forall} x\, p) \dot{\to} [t/x]p)$
- `proof` [] $(p \dot{\to} (\dot{\forall} x\, p))$ provided by $x \notin \mathrm{FV}(p)$
- `proof` [] $((\dot{\forall} x\, (p \dot{\to} q)) \dot{\to} ((\dot{\forall} x\, p) \dot{\to} (\dot{\forall} x\, q)))$

# Formalisation (Deduction System)

The axioms for Leibniz equality.

- $\texttt{proof}\ []\ (v_0 \doteq v_0)$
- $\texttt{proof}\ []\ ((v_0 \doteq v_1) \dot{\to} (v_1 \doteq v_0))$
- $\texttt{proof}\ []\ ((v_0 \doteq v_1) \dot{\to} ((v_1 \doteq v_2) \dot{\to} (v_0 \doteq v_2)))$
- $\texttt{proof}\ []\ (\texttt{Fun\_eqAxm}\, f)$ for each $F \in \mathcal{F}$
- $\texttt{proof}\ []\ (\texttt{Rel\_eqAxm}\, R)$ for each $R \in \mathcal{R}$

where

$$\begin{aligned}
\texttt{Fun_eqAxm}\, f \ :=\ & ((v_{2n_f-2} \doteq v_{2n_f-1}) \dot{\to} (\cdots \dot{\to} ((v_0 \doteq v_1) \dot{\to} \\
& (f(v_{2n_f-2}, \cdots, v_0) \doteq f(v_{2n_f-1}, \cdots, v_1))))), \\
\texttt{Rel_eqAxm}\, R \ :=\ & ((v_{2n_R-2} \doteq v_{2n_R-1}) \dot{\to} (\cdots \dot{\to} ((v_0 \doteq v_1) \dot{\to} \\
& (R(v_{2n_R-2}, \cdots, v_0) \dot{\to} R(v_{2n_R-1}, \cdots, v_1))))).
\end{aligned}$$

# Formalisation (Meta-theory)

**Theorem.** The Deduction Theorem.
*For any set* $\Gamma$ *of* $L$*-formulae and any* $L$*-formulae* $A$, $B$,

$$\Gamma \vdash A \dot{\to} B \leftrightarrow \{A\} \cup \Gamma \vdash B.$$

# Formalisation (Meta-theory)

**Theorem.** The Deduction Theorem.
*For any set* $\Gamma$ *of* $L$*-formulae and any* $L$*-formulae* $A$, $B$,

$$\Gamma \vdash A \dot{\to} B \leftrightarrow \{A\} \cup \Gamma \vdash B.$$

Proof.

($\Rightarrow$) Apply Mp and Axm.

($\Leftarrow$) There is a finite list $\vec{\varphi}$ of $L$-formulae with PF : `proof` $\vec{\varphi}$ $B$ such that $\vec{\varphi} \subseteq \{A\} \cup \Gamma$. It is sufficient to show $\vec{\varphi} \cap \Gamma \vdash A \dot{\to} B$. It can be proved by induction on PF. The most difficult case is Gen, but observing that

$$A \in \vec{\varphi} \vee \vec{\varphi} \subseteq \Gamma,$$

we can close the case.

□

# Formalisation (Meta-theory)

**Theorem.** The Soundness Theorem.
*For any set* $\Gamma$ *of* $L$*-formulae and any* $L$*-formula* $C$*,*

$$\Gamma \vdash C \rightarrow \Gamma \vDash C.$$

# Formalisation (Meta-theory)

**Theorem.** The Soundness Theorem.
*For any set* $\Gamma$ *of* $L$*-formulae and any* $L$*-formula* $C$*,*

$$\Gamma \vdash C \to \Gamma \models C.$$

Proof.

$\texttt{proof}\ \vec{\varphi}\ C$ is inhabited for some $\vec{\varphi} \subseteq \Gamma$. Now, by induction on $\texttt{proof}\ \vec{\varphi}\ C$. The law of excluded middle is assumed to show the theorem, because of the following axiom scheme:

$$\texttt{proof}\ []\ (((\dot{\neg}\, q) \mathrel{\dot{\to}} (\dot{\neg}\, p)) \mathrel{\dot{\to}} (p \mathrel{\dot{\to}} q)).$$

□

# Formalisation (Meta-theory)

**Lemma.** *For any set* $\Gamma$ *of* $L$*-formulae and any* $L$*-formulae* $\varphi_1$, $\varphi_2$,

$$\frac{\Gamma \vdash \varphi_1 \quad \varphi_1 \equiv_\alpha \varphi_2}{\Gamma \vdash \varphi_2}$$

# Formalisation (Meta-theory)

**Lemma.** *For any set* $\Gamma$ *of L-formulae and any L-formulae* $\varphi_1$, $\varphi_2$,

$$\frac{\Gamma \vdash \varphi_1 \quad \varphi_1 \equiv_\alpha \varphi_2}{\Gamma \vdash \varphi_2}$$

Proof.

It is enough to show

$$\varphi_1 \equiv_\alpha \varphi_2 \to (\{\varphi_1\} \vdash \varphi_2 \wedge \{\varphi_2\} \vdash \varphi_1).$$

Now, by strong induction on the height of $\varphi_1$ and destructing $\varphi_1 \equiv_\alpha \varphi_2$. □

# Formalisation (Meta-theory)

**Fact.** *All of the following rules are admissible:*

$$\frac{}{\Gamma \vdash t \doteq t} \qquad \frac{\Gamma \vdash t \doteq t'}{\Gamma \vdash t' \doteq t} \qquad \frac{\Gamma \vdash t \doteq t' \quad \Gamma \vdash t' \doteq t''}{\Gamma \vdash t \doteq t''}$$

$$\frac{\Gamma \vdash t_1 \doteq t'_1 \quad \cdots \quad \Gamma \vdash t_{n_f} \doteq t'_{n_f}}{\Gamma \vdash f(t_1, \cdots, t_{n_f}) \doteq f(t'_1, \cdots, t'_{n_f})} \qquad \frac{\Gamma \vdash t_1 \doteq t'_1 \quad \cdots \quad \Gamma \vdash t_{n_R} \doteq t'_{n_R}}{\Gamma \vdash R(t_1, \cdots, t_{n_R}) \dot{\to} R(t'_1, \cdots, t'_{n_R})}$$

$$\frac{\Gamma \vdash t_1 \doteq t_2 \quad \Gamma \vdash [t_1/x]\varphi}{\Gamma \vdash [t_2/x]\varphi}$$

# Formalisation (Meta-theory)

**Fact.** *All of the following rules are admissible:*

$$\frac{}{\Gamma \vdash t \doteq t} \qquad \frac{\Gamma \vdash t \doteq t'}{\Gamma \vdash t' \doteq t} \qquad \frac{\Gamma \vdash t \doteq t' \quad \Gamma \vdash t' \doteq t''}{\Gamma \vdash t \doteq t''}$$

$$\frac{\Gamma \vdash t_1 \doteq t_1' \quad \cdots \quad \Gamma \vdash t_{n_f} \doteq t_{n_f}'}{\Gamma \vdash f(t_1, \cdots, t_{n_f}) \doteq f(t_1', \cdots, t_{n_f}')} \qquad \frac{\Gamma \vdash t_1 \doteq t_1' \quad \cdots \quad \Gamma \vdash t_{n_R} \doteq t_{n_R}'}{\Gamma \vdash R(t_1, \cdots, t_{n_R}) \mathrel{\dot{\to}} R(t_1', \cdots, t_{n_R}')}$$

$$\frac{\Gamma \vdash t_1 \doteq t_2 \quad \Gamma \vdash [t_1/x]\varphi}{\Gamma \vdash [t_2/x]\varphi}$$

Proof.

Note that $\emptyset \vdash (\dot{\forall} v_{n-1}\, (\dot{\forall} v_{n-2} \cdots (\dot{\forall} v_0\, \psi))) \mathrel{\dot{\to}} [i \mapsto \textbf{if } i < n \textbf{ then } \sigma(i) \textbf{ else } v_i]\psi$ holds for any $n \in \mathbb{N}$, any $\sigma : \mathbb{N} \to \mathtt{trm}\, L$, and any $L$-formula $\psi$. The last rule follows from

$$\frac{\Gamma \vdash t_1 \doteq t_2}{(\Gamma \vdash (\varphi[x := t_1]) \mathrel{\dot{\to}} (\varphi[x := t_2])) \wedge (\Gamma \vdash (\varphi[x := t_2]) \mathrel{\dot{\to}} (\varphi[x := t_1]))}$$

which can be shown by strong induction on the height of $\varphi$. □

# Formalisation (Meta-theory)

**Lemma.** The Substitution Lemma.
*For any set* $\Gamma$ *of* $L$*-formulae, any* $L$*-formula* $\varphi$*, and any* $\sigma : \mathbb{N} \to \mathtt{trm}\, L$*,*

$$\Gamma \vdash \varphi \to [\sigma]\Gamma \vdash [\sigma]\varphi.$$

# Formalisation (Meta-theory)

**Lemma.** The Substitution Lemma.
*For any set* $\Gamma$ *of* $L$*-formulae, any* $L$*-formula* $\varphi$*, and any* $\sigma : \mathbb{N} \to \mathtt{trm}\, L$*,*

$$\Gamma \vdash \varphi \to [\sigma]\Gamma \vdash [\sigma]\varphi.$$

Proof.

There is a list $\vec{\psi}$ such that $\mathtt{proof}\ \vec{\psi}\ \varphi$ is inhabited and $\vec{\psi} \subseteq \Gamma$. Now, induction on $\vec{\psi}$. One can prove $\emptyset \vdash \varphi \to \emptyset \vdash [\sigma]\varphi$ by induction on $\mathtt{proof}\ []\ \varphi$. □

# Formalisation (Meta-theory)

**Lemma.** The Enumeration Lemma.

*If $\mathcal{F}$, $\mathcal{C}$, and $\mathcal{R}$ are countable, then* $\mathtt{trm}\, L$ *and* $\mathtt{frm}\, L$ *are enumerable.*

To prove the Countable Completeness Theorem, we now assume that $L$ is an arbitrary first-order language whose sets of function, constant, and relation symbols are countable.

**Details.** Let $X$ : **Type**. $X$ is said to be ***countable*** when there exists an injection $X \rightarrowtail \mathbb{N}$. $X$ is said to be ***enumerable*** when there exists a surjection $\mathbb{N} \twoheadrightarrow X$, which is called an ***enumeration*** of $X$. Note that $X$ is countable if and only if it is enumerable or empty.

# Formalisation (Meta-theory)

**Lemma.** The Enumeration Lemma.

*If $\mathcal{F}$, $\mathcal{C}$, and $\mathcal{R}$ are countable, then* $\mathtt{trm}\, L$ *and* $\mathtt{frm}\, L$ *are enumerable.*

Proof.

Using the Cantor pairing function $\mathtt{cp} : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$, one can construct functions that return abstract syntax trees of $\mathtt{trm}\, L$ and $\mathtt{frm}\, L$, respectively, with heights that are less than or equal to the second parameter, using the first parameter as the seed for AST generation. Now, using $\mathtt{cp}$ again, it is possible to construct enumerations of $\mathtt{trm}\, L$ and $\mathtt{frm}\, L$. □

To prove the Countable Completeness Theorem, we now assume that $L$ is an arbitrary first-order language whose sets of function, constant, and relation symbols are countable.

**Details.** Let $X$ : **Type**. $X$ is said to be ***countable*** when there exists an injection $X \rightarrowtail \mathbb{N}$. $X$ is said to be ***enumerable*** when there exists a surjection $\mathbb{N} \twoheadrightarrow X$, which is called an ***enumeration*** of $X$. Note that $X$ is countable if and only if it is enumerable or empty.

# Formalisation (Meta-theory)

**Definition.** Each ***Henkin constant symbol*** $\bar{c}$ is defined to be a natural number. Furthermore, $L'$ is defined to be the augmented language obtained by adding Henkin constant symbols to $L$.
**Fact.** $\mathtt{frm}\, L'$ *is also enumerable.*

**Notation.** For an $L$-formula $\varphi$, the embedding $\varphi$ into $L'$ is denoted by $\lceil\varphi$.
**Fact.** *For any $L$-formulae $\varphi$ and $\psi$,*

$$\varphi \equiv_\alpha \psi \leftrightarrow \lceil\varphi \equiv_\alpha \lceil\psi.$$

**Details.** The set of constant symbols of $L'$ can be thought of as $\mathcal{C} \uplus \mathbb{N}$, while the other sets of symbols are the same as those of $L$.

# Formalisation (Meta-theory)

**Definition.** Each ***Henkin constant symbol*** $\bar{c}$ is defined to be a natural number. Furthermore, $L'$ is defined to be the augmented language obtained by adding Henkin constant symbols to $L$.
**Fact.** $\mathtt{frm}\, L'$ *is also enumerable.*

Proof.

$\mathcal{C} \uplus \mathbb{N}$ is countable. Now, apply the Enumeration Lemma. □

**Notation.** For an $L$-formula $\varphi$, the embedding $\varphi$ into $L'$ is denoted by $\lceil \varphi$.
**Fact.** *For any $L$-formulae $\varphi$ and $\psi$,*

$$\varphi \equiv_\alpha \psi \leftrightarrow \lceil \varphi \equiv_\alpha \lceil \psi.$$

Proof.

Both sides can be proved by induction. □

**Details.** The set of constant symbols of $L'$ can be thought of as $\mathcal{C} \uplus \mathbb{N}$, while the other sets of symbols are the same as those of $L$.

# Formalisation (Meta-theory)

**Fact.** *For any set* $\Gamma$ *of* $L$*-formulae and any* $L$*-formula* $\varphi$*,*

$$\neg\Gamma \vdash \neg\varphi \leftrightarrow \Gamma \vdash \varphi.$$

# Formalisation (Meta-theory)

**Fact.** *For any set* $\Gamma$ *of* $L$*-formulae and any* $L$*-formula* $\varphi$*,*

$$\upharpoonright\Gamma \vdash \upharpoonright\varphi \leftrightarrow \Gamma \vdash \varphi.$$

Proof.

($\Leftarrow$) By induction on `proof`.

($\Rightarrow$) By the Deduction Theorem, it is sufficient to show $\emptyset \vdash \upharpoonright\varphi \to \emptyset \vdash \varphi$.
Let $\mathbf{shift}\,(X) := [v_i \mapsto v_{i+1}, \overline{c} \mapsto v_0]\,X$—i.e.,

$$\mathbf{shift}\,(v_i) := v_{i+1},\ \mathbf{shift}\,(\overline{c}) := v_0,\ \text{and}\ \mathbf{shift}\,(\dot{\forall} v_i\,\psi) := \dot{\forall} v_{i+1}\,(\mathbf{shift}\,(\psi)).$$

Then, by induction on `proof` $[]\ A$, the following can be shown:

$$(\forall A : \mathtt{frm}\,L')(\emptyset \vdash A \to \emptyset \vdash \mathbf{shift}\,(A)).$$

Now, noting $\upharpoonright([i \mapsto v_{i-1}](\mathbf{shift}\,(\upharpoonright\varphi))) \equiv_\alpha \upharpoonright\varphi$, it is possible to derive

$$\emptyset \vdash \upharpoonright\varphi \implies \emptyset \vdash \mathbf{shift}\,(\upharpoonright\varphi) \implies \emptyset \vdash [i \mapsto v_{i-1}](\mathbf{shift}\,(\upharpoonright\varphi)) \implies \emptyset \vdash \varphi.$$

□

# Formalisation (Meta-theory)

**Definition.** We are going to define a sequence $\langle \theta_n \rangle_{n\in\mathbb{N}}$ of $L'$-formulae, which will be called the sequence of ***Henkin axioms***.
Let $\langle (x_n, \varphi_n) \rangle_{n\in\mathbb{N}}$ be a fixed enumeration of pairs, where $x_n$ is an individual variable and $\varphi_n$ is an $L'$-formula. For $n \in \mathbb{N}$, define

$$\theta_n := ([\overline{c}_n/x_n]\varphi_n) \dot{\to} (\dot{\forall} x_n \, \varphi_n),$$

where $\overline{c}_n$ is the first of the Henkin constant symbols not occurring in $\varphi_n$ or $\theta_k$ for any $k < n$.
**Details.** To refer to the Henkin axioms $\theta_k$ for $k < n$,

*memoisation* was employed.

That is, I defined a sequence

$$\langle ((\theta_{n-1}, \theta_{n-2}, \cdots, \theta_0), (\overline{c}_{n-1}, \overline{c}_{n-2}, \cdots, \overline{c}_0)) \rangle_{n\in\mathbb{N}}$$

of pairs consisting of a vector of Henkin axioms and a vector of Henkin constants.

# Formalisation (Meta-theory)

**Fact**. *Define* $\dot{\bot} := \dot{\neg}(\dot{\forall} v_0 \, (v_0 \doteq v_0))$. *Then, for any set* $\Gamma$ *of* $L$*-formulae,*

$$\Gamma \vdash \dot{\bot} \leftrightarrow \{\theta_n \mid n \in \mathbb{N}\} \cup \urcorner\Gamma \vdash \dot{\bot} \, .$$

# Formalisation (Meta-theory)

**Fact**. *Define* $\dot{\bot} := \dot{\neg}(\dot{\forall} v_0\,(v_0 \doteq v_0))$. *Then, for any set* $\Gamma$ *of* $L$*-formulae,*

$$\Gamma \vdash \dot{\bot} \leftrightarrow \{\theta_n \mid n \in \mathbb{N}\} \cup \urcorner\Gamma \vdash \dot{\bot}\,.$$

Proof.

($\Rightarrow$) $\Gamma \vdash \dot{\bot} \implies \urcorner\Gamma \vdash \dot{\bot} \implies \{\theta_n \mid n \in \mathbb{N}\} \cup \urcorner\Gamma \vdash \dot{\bot}$.

($\Leftarrow$) Let $\Gamma_n := \{\theta_k \mid k < n\} \cup \urcorner\Gamma$. Then $\{\theta_n \mid n \in \mathbb{N}\} \cup \urcorner\Gamma \vdash \dot{\bot} \leftrightarrow (\exists n \in \mathbb{N})(\Gamma_n \vdash \dot{\bot})$. Thus, it is enough to show $(\forall n \in \mathbb{N})(\Gamma_n \vdash \dot{\bot} \to \urcorner\Gamma \vdash \dot{\bot})$. This follows from

$$\begin{aligned}
\Gamma_{n+1} \vdash \dot{\bot} &\implies \Gamma_n \vdash \dot{\neg}\,\theta_n \implies (\Gamma_n \vdash [\overline{c}_n/x_n]\varphi_n) \wedge (\Gamma_n \vdash \dot{\neg}(\dot{\forall} x\,\varphi_n)) \\
&\implies (\vec{\psi} \vdash [\overline{c}_n/x_n]\varphi_n) \wedge (\Gamma_n \vdash \dot{\neg}(\dot{\forall} x\,\varphi_n)) \\
&\implies ([y/\overline{c}_n]\vec{\psi} \vdash [y/\overline{c}_n][\overline{c}_n/x_n]\varphi_n) \wedge (\Gamma_n \vdash \dot{\neg}(\dot{\forall} x_n\,\varphi_n)) \\
&\implies (\vec{\psi} \vdash \dot{\forall} x_n\,\varphi_n) \wedge (\Gamma_n \vdash \dot{\neg}(\dot{\forall} x_n\,\varphi_n)) \implies \Gamma_n \vdash \dot{\bot},
\end{aligned}$$

where $\vec{\psi} \subseteq \Gamma_n$, $\vec{\psi} \vdash [\overline{c}_n/x_n]\varphi_n$, and $y \notin \mathrm{FV}(\vec{\psi} +\!\!+ [\dot{\forall} x_n\,\varphi_n; [\overline{c}_n/x_n]\varphi_n])$.

□

# Formalisation (Meta-theory)

**Definitions.** Let $\mathcal{L}$ be a first-order language.

(a) For a set $\Gamma$ of $\mathcal{L}$-formulae, define $\mathsf{Th}_{\mathcal{L}}(\Gamma) : \mathtt{frm}\,\mathcal{L} \to \mathbf{Prop}$ by

$$\mathsf{Th}_{\mathcal{L}}(\Gamma) := \varphi \mapsto \Gamma \vdash \varphi.$$

Note that $\mathsf{Th}_{\mathcal{L}}$ is a closure operator on $\mathtt{frm}\,\mathcal{L}$.

(b) A set $\Gamma$ of $\mathcal{L}$-formulae is said to be ***consistent*** when $\Gamma \nvdash \dot{\bot}$.

(c) A set $\Delta$ of $\mathcal{L}$-formulae is said to be ***maximally consistent*** when, for any set $\Delta'$ of $\mathcal{L}$-formulae with $\Delta \subseteq \Delta'$, $\Delta' \nvdash \dot{\bot}$ if and only if $\Delta = \Delta'$.

Assuming that a consistent set $\Gamma$ of $L$-formulae is given, let us construct a maximally consistent set $\Delta$ of $L'$-formulae such that

$$\{\theta_n \mid n \in \mathbb{N}\} \cup \neg\Gamma \subseteq \Delta.$$

# Formalisation (Meta-theory)

Let $\langle \psi_n \rangle_{n \in \mathbb{N}}$ be a fixed enumeration of $L'$-formulae. Define $\langle \Delta_n \rangle_{n \in \mathbb{N}}$ by

- $\Delta_0 := \mathsf{Th}_{L'}(\{\theta_n \mid n \in \mathbb{N}\} \cup \neg\Gamma)$,
- $\Delta_{n+1} := \begin{cases} \mathsf{Th}_{L'}(\{\psi_n\} \cup \Delta_n), & \text{if } \{\psi_n\} \cup \Delta_n \vdash \dot{\bot} \leftrightarrow \Delta_n \vdash \dot{\bot}; \\ \mathsf{Th}_{L'}(\Delta_n), & \text{otherwise.} \end{cases}$

Now, take $\Delta := \bigcup_{n \in \mathbb{N}} \Delta_n$. Then, $\Delta$ is maximally consistent. Hence,

$$\varphi \in \Delta \leftrightarrow \Delta \vdash \varphi$$

for any $L'$-formula $\varphi$. Furthermore, even without the law of excluded middle,

- for any $L'$-formula $p$, $(\dot{\neg}\, p) \notin \Delta \to p \in \Delta$;
- for any $L'$-formula $p$, $(\dot{\neg}\, p) \in \Delta \leftrightarrow p \notin \Delta$;
- for any $L'$-formulae $p$ and $q$, $(p \mathrel{\dot{\to}} q) \in \Delta \leftrightarrow (p \in \Delta \to q \in \Delta)$; and
- for any $L'$-formula $p$ and any individual variable $x$,

$$(\dot{\forall} x\, p) \in \Delta \leftrightarrow (\forall t : \mathtt{trm}\, L')([t/x]p \in \Delta).$$

# Formalisation (Meta-theory)

**Theorem.** The Model Existence Theorem.
*Define an $L'$-structure $\mathfrak{A}$ by*

- $|\mathfrak{A}| := \mathtt{trm}\, L'$,
- $t_1 \sim_{|\mathfrak{A}|} t_2 := \Delta \vdash t_1 \doteq t_2$,
- $f^{\mathfrak{A}} := \vec{t} \mapsto f\, \vec{t}$,
- $c^{\mathfrak{A}} := c$,
- $R^{\mathfrak{A}} := \vec{t} \mapsto \Delta \vdash R\, \vec{t}$,

*and $\rho := i \mapsto v_i$. Then, for any $L'$-formula $\varphi$,*

$$\varphi \in \Delta \leftrightarrow [\![\varphi]\!]^{\mathfrak{A}}_{\rho}.$$

*Therefore, $\mathfrak{A} \models \Delta[\rho]$ and $\mathfrak{A} \models (\urcorner\Gamma)[\rho]$.*

# Formalisation (Meta-theory)

**Theorem.** The Model Existence Theorem.
*Define an $L'$-structure $\mathfrak{A}$ by*

- $|\mathfrak{A}| := \mathtt{trm}\, L'$,
- $t_1 \sim_{|\mathfrak{A}|} t_2 := \Delta \vdash t_1 \doteq t_2$,
- $f^{\mathfrak{A}} := \vec{t} \mapsto f\, \vec{t}$,
- $c^{\mathfrak{A}} := c$,
- $R^{\mathfrak{A}} := \vec{t} \mapsto \Delta \vdash R\, \vec{t}$,

*and $\rho := i \mapsto v_i$. Then, for any $L'$-formula $\varphi$,*

$$\varphi \in \Delta \leftrightarrow [\![\varphi]\!]^{\mathfrak{A}}_{\rho}.$$

*Therefore, $\mathfrak{A} \models \Delta[\rho]$ and $\mathfrak{A} \models (\urcorner\Gamma)[\rho]$.*

Proof.

By strong induction on the height of $\varphi$. This theorem can be proved without the law of excluded middle. □

# Formalisation (Meta-theory)

**Theorem.** The Countable Completeness Theorem.
*Let $X$ be a set of $L$-formulae and let $b$ be an $L$-formula such that $X \models b$. Then $X \vdash b$.*

# Formalisation (Meta-theory)

**Theorem.** The Countable Completeness Theorem.
*Let $X$ be a set of $L$-formulae and let $b$ be an $L$-formula such that $X \vDash b$. Then $X \vdash b$.*

Proof.

Put $\Gamma := \{\dot{\neg} b\} \cup X$. Assume $X \nvdash b$. Then $\Gamma \nvdash \dot{\bot}$. Restricting the $L'$-structure $\mathfrak{A}$ obtained by the Model Existence Theorem to $L$ yields $\Gamma \nvDash \dot{\bot}$, which contradicts the assumption $X \vDash b$. Therefore, we can conclude that the assumption $X \nvdash b$ is false and finally obtain $X \vdash b$. □

# Comparison

**Comparison with Ilik (2010).**
He formalised in the first chapter of his thesis the completeness theorem of classical natural deduction, but the proof was incomplete.

# Comparison

**Comparison with Herberlin, Kim, and Lee (2017).**
They formalised the Weak Completeness Theorem of the Gentzen-style sequent calculus LJT for Kripke's semantics instead of Tarski's semantics.

# Comparison

**Comparison with Forster, Kirst, and Wehr (2021).**
They formalised the completeness theorem of classical natural deduction. However, their setting, based on de Bruijn index, makes it difficult to use as a framework.

# Comparison

**Comparison with From (2022).**
She formalised the completeness theorem of a Hilbert calculus while using de Bruijn index. However, there is a side condition. The main theorem of the study is

$$\emptyset \vdash \varphi \leftrightarrow \emptyset \vDash \varphi.$$

# Comparison

**Comparison with Herberlin and Ilik (2024).**
They formalised the completeness theorem for classical first-order languages *not* equipped with Leibniz equality. They also modified Henkin's method.

# Conclusion

**The Main Result.** I formalised classical first-order logic *equipped with* Leibniz equality using `Coq 8.18.0`, *assuming only the law of excluded middle.*

A Coq Script for Checking Theorem Statements and Used Axioms.

```
Check @HilbertCalculus_sound.
Print Assumptions HilbertCalculus_sound.
Check @HilbertCalculus_complete.
Print Assumptions HilbertCalculus_complete.
```

# Conclusion

**The Main Result.** I formalised classical first-order logic *equipped with* Leibniz equality using `Coq 8.18.0`, *assuming only the law of excluded middle.*

```
@HilbertCalculus_sound
     : forall (L : language) (Gamma : ensemble (frm L)) (C : frm L),
       Gamma ⊢ C -> Gamma ⊨ C
Axioms:
classic : forall P : Prop, P \/ ~ P
@HilbertCalculus_complete
     : forall L : language,
       isCountable (function_symbols L) ->
       isCountable (constant_symbols L) ->
       isCountable (relation_symbols L) ->
       forall (X : ensemble (frm L)) (b : frm L), X ⊨ b -> X ⊢ b
Axioms:
classic : forall P : Prop, P \/ ~ P
lim@K1-20230101CTCH:~/portfolio/Fol-archived$
```

Figure: The result from the script.

# Conclusion

**The Contributions.**

- The fact that, for any set $\Gamma$ of $L$-formulae and any $L$-formula $\varphi$,

$$\nmid \Gamma \vdash \nmid \varphi \leftrightarrow \Gamma \vdash \varphi,$$

  had not been formalised in any former studies.
- My formal proof of the Countable Completeness Theorem closely follows the approach presented in Enderton's mathematical text, ensuring its fidelity to orthodox methodology.
- While there are many studies that formalise natural deduction, Hilbert calculi have received comparatively less attention. This makes my formalisation a significant contribution.

# Conclusion

**Future Work.**

- The practical applicability of the framework such as PA and ZFC will be explored.
- Since both systems have axiom schemata, a tool will be made for handling meta-variables.
- The Completeness Theorem for first-order languages with cardinalities greater than $\aleph_0$ will be formally proved as well.

*Thank you for listening!*

E-mail: gijungdduk@naver.com

GitHub: github.com/KiJeong-Lim/Fol-archived