고전일차논리의 새로운 Coq 형식화와 건전성 및 완전성 정리의 증명

임기정
전남대학교 기계공학부
gijungdduk@naver.com

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

Kijeong Lim

Abstract

In this study, assuming the law of excluded middle, the soundness theorem of a Hilbert calculus with respect to Tarski's semantics and the completeness theorem for all countable first-order languages are formally proved. This formalisation provides named quantifiers and Leibniz equality. These features enable mathematicians to use it as a framework for verifying their proofs based on classical first-order logic.

1 Introduction

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. Thus, 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, the Software Foundation Laboratory formalized a theory of ordinal numbers, which is a purely mathematical concept, to state and/or prove the termination of a program in the CCR library [10] of the laboratory.

Most proof assistants do not adopt the usual first-order language used by working mathematicians (e.g., ZFC). Instead, they take dependent type theory as their foundation. Thus, translating even a simple proof directly into a proof assistant without an intermediate language can be error-prone and requires expertise in both foundational systems. Therefore, to facilitate the translation of mathematical theories into a proof assistant, an intermediate language is deemed necessary.

In this paper, the formalisation of a classical first-order logic and its proof rules in the setting of dependent type theory are proposed. To serve as an intermediate language, it closely resembles the traditional first-order language used by mathematicians, allowing many existing proofs to be integrated with minimal effort. This formalisation has been fully implemented in the Coq proof assistant. Hence, a proof written in our language is automatically translated into a proof term inside Coq.

Users are given both syntactic and semantic systems to obtain a consequence from a set of a possibly infinite number of assumptions. The syntactic entailment $\Gamma \vdash C$ means that C is provable assuming Γ, while the semantic entailment $\Gamma \vDash C$ means that every model that satisfies all of the members of Γ also satisfies C. Here, the word 'satisfy' is defined as the notion of "Tarski's definition of truth" in Section 3.2. The soundness theorem and the completeness theorem guarantee the two systems coincide with each other. Furthermore, to facilitate the translation of mathematical results, Leibniz equality $\doteq$ with its introduction and elimination rules is provided, and the quantifiers in the system under this study bind named variables. Thus, variables are not represented by de Bruijn indices.

Obviously, the idea of formalizing a first-order logic in dependent type theory is not new. To name a few:

Ilik (2010) [7] formalized in its first chapter the completeness theorem of classical natural deduction for Tarski's semantics, but the proof was incomplete.

Constable and Bickford (2014) [1] formalized the completeness theorem for intuitionistic logic rather than classical logic.

Herberlin, Kim and Lee (2017) [6] formalized the completeness theorem of the Gentzen-style sequent calculus LJT for Kripke's semantics instead of Tarski's semantics.

Forster, Kirst and Wehr (2021) [4] formalized the completeness theorem of classical natural deduction for Tarski's semantics. While their work is perfect, the setting based on de Bruijn index makes it difficult to use.

From (2022) [5] formalized the completeness theorem of a Hilbert calculus for Tarski's semantics while adopting de Bruijn index. However, there is a side condition.

However, there are no known works that give Leibniz equality and named quantifiers and allow users to assume an infinite number of axioms at the same time. Thus, the formalisation in this paper is the richest one.

2 Project overview

The formalisation with proofs in this paper is implemented as a Coq project in [8]. Here the organization of the project is briefly illustrated.

`theories/Logic/BasicFol.v` defines the syntax and semantics of first-order languages with substitutions and α-equivalence.

`theories/Logic/BasicFol2.v` has statements and proofs on augmented languages by adding Hekin constants.

`theories/Logic/HilbertFol.v` defines a Hilbert calculus. Moreover, it states and proves Theorem 2.

`theories/Logic/HilbertFol2.v` provides additional inference rules, includes theorems for the augmented languages, and contains Facts 4 and 5.

`theories/Logic/ClassicalFol.v` gives Theorems 3 and 6, which are the soundness theorem and the countable completeness theorem, respectively.

3 Formalisation

3.1 Syntax

A ***first-order language*** L is represented as a record with the following fields: a set of ***function symbols***, a set of ***constant symbols***, a set of ***relation symbols***, a table mapping each function symbol to its arity, and a table mapping each relation symbol to its arity. Each ***individual variable*** x corresponds to a natural number:

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

An ***L-term*** t and an ***L-formula*** φ are defined inductively:

$$\begin{aligned} t \ &::= \ x \ \big| \ f\,\vec{t} \ \big| \ c \\ \varphi \ &::= \ R\,\vec{t} \ \big| \ t_1 \doteq t_2 \ \big| \ \dot{\neg}\varphi_1 \ \big| \ \varphi_1 \mathbin{\dot{\to}} \varphi_2 \ \big| \ \dot{\forall}x \ \varphi_1 \end{aligned}$$

where f is a function symbol of L, c is a constant symbol of L, R is a relation symbol of L, and $\vec{t}$ is a vector of L-terms. For a first-order language L, write $\texttt{trm}\,L$ for the type of L-terms, $\texttt{trms}\,L\,n$ for the type of vectors of L-terms of length n, and $\texttt{frm}\,L$ for the type of L-formulae.

Similarly to [2], a simultaneous substitution σ is defined as a map $\mathbb{N} \to \texttt{trm}\,L$, saying that $\sigma(i)$ is substituted for v_i. Additionally, α-equivalence is defined in the same way as [2].

3.2 Semantics

Let L be a first-order language. A ***structure*** $\mathfrak{A}$ of L consists of a setoid $(A, \sim_A)$, interpretations $f^{\mathfrak{A}} : A^{n_f} \to A$ for each function symbol f of L, interpretations $c^{\mathfrak{A}} : A$ for each constant symbol c of L, and interpretations $R^{\mathfrak{A}} : A^{n_R} \to \mathbf{Prop}$ for each relation symbol R of L such that A is inhabited and $\sim_A$ is congruent with all interpretations of $\mathfrak{A}$, where n_f and n_R are the arities of f and R, respectively. Denote the type A by $|\mathfrak{A}|$, which is called the ***domain of discourse*** of $\mathfrak{A}$.

Let $\mathfrak{A}$ be a structure of L and $\rho : \mathbb{N} \to |\mathfrak{A}|$ be given. An interpretation $[\![t]\!]^{\mathfrak{A}}_{\rho} : |\mathfrak{A}|$ of an L-term t is defined recursively in a natural way, starting with $[\![v_i]\!]^{\mathfrak{A}}_{\rho} := \rho\,(i)$. An interpretation $[\![\varphi]\!]^{\mathfrak{A}}_{\rho} : \mathbf{Prop}$ of an L-formula φ is defined recursively in a natural way, with Leibniz equality $\doteq$ interpreted as $\sim_{|\mathfrak{A}|}$. We say that $(\mathfrak{A}, \rho)$ ***satisfies*** φ if $[\![\varphi]\!]^{\mathfrak{A}}_{\rho}$ is inhabited.

Definition 1. Given a first-order language L, an L-formula C is said to be a ***semantic consequence*** of a set Γ of L-formulae if and only if for every structure $\mathfrak{A}$ of L and every $\rho : \mathbb{N} \to |\mathfrak{A}|$, if $(\mathfrak{A}, \rho)$ satisfies all of the members of Γ, then it also satisfies C. Then write $\Gamma \vDash C$.

3.3 Deduction system

The deduction system in this system is defined à la Hilbert in a manner that is similar to that used in the thesis of Russell O'Connor [9]. For a (possibly infinite) set Γ of L-formulae and an L-formula C, write $\Gamma \vdash C$ to denote that there is a finite list $\vec{\varphi} \subseteq \Gamma$ such that $\texttt{proof}\,\vec{\varphi}\,C$ is inhabited. Here, $\texttt{proof} : \texttt{list}\,(\texttt{frm}\,L) \to \texttt{frm}\,L \to \mathbf{Set}$ is an inductive family of types representing proofs with constructors including:

$$\textsc{Axm}\ \frac{}{\texttt{proof}\ [p]\ p} \qquad \textsc{Mp}\ \frac{\texttt{proof}\ \vec{\varphi}_1\ (p \mathbin{\dot{\to}} q) \quad \texttt{proof}\ \vec{\varphi}_2\ p}{\texttt{proof}\ (\vec{\varphi}_1 \mathbin{+\!\!+} \vec{\varphi}_2)\ q}$$

$$\textsc{Gen}\ \frac{\texttt{proof}\ \vec{\varphi}\ p}{\texttt{proof}\ \vec{\varphi}\ (\dot{\forall}x\ p)}\ x \notin \mathrm{FV}(\vec{\varphi})$$

$$\textsc{Eqn\_Fun}\ \frac{}{\texttt{proof}\ []\ (\texttt{Fun\_eqAxm}\,f)} \qquad \textsc{Eqn\_Rel}\ \frac{}{\texttt{proof}\ []\ (\texttt{Rel\_eqAxm}\,R)}$$

Axm is required to show $\varphi \in \Gamma \implies \Gamma \vdash \varphi$. Since Axm and Mp are modified in such a way, it requires the list of axioms to be in the correct order for the proof. Gen has a proviso, which is a trade-off for eliminating the side condition of the Deduction Theorem. Eqn_Fun and Eqn_Rel state that $\doteq$ is congruent with f and R, respectively. The constructors for $\doteq$ are just axioms, not axiom schemata. Thus, there are no parameters that can be adjusted except for f and/or R.

4 Meta-theory

First of all, the following theorem is fundamental:

Theorem 2 (Deduction)**.** For any L-formulae H, C, and any set Γ of L-formulae,

$$\Gamma \vdash H \mathbin{\dot{\to}} C \iff \{H\} \cup \Gamma \vdash C\ .$$

Then, one can prove the soundness theorem by induction on `proof` with $\texttt{classic} : \forall P : \mathbf{Prop}, P \vee \neg P$.

Theorem 3 (Soundness)**.** For any L-formula C and any set Γ of L-formulae, $\Gamma \vdash C$ implies $\Gamma \vDash C$.

Furthermore, we need to prove the properties of $\doteq$. One can show that $\doteq$ is reflexive, symmetric, and transitive, and that it is congruent with f and R. Finally, one can obtain:

Fact 4. For any individual variable x, any L-terms t_1, t_2, any L-formula φ, and any set Γ of L-formulae,

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

We now aim to prove the completeness theorem for an arbitrary countable first-order language L. Define an extension L' of L by adding ***Henkin constants***, where each Henkin constant $\bar{c}$ corresponds to a natural number. Denote the embedding of an L-formula φ into the language L' by $\upharpoonleft\varphi$. For a set Γ of L-formulae, define $\upharpoonleft\Gamma := \{\upharpoonleft\varphi \mid \varphi \in \Gamma\}$. Observe:

Fact 5. For any L-formula φ and any set Γ of L-formulae,

$$\upharpoonleft\Gamma \vdash \upharpoonleft\varphi \iff \Gamma \vdash \varphi.$$

Meanwhile, define a sequence $\langle \theta_n \rangle_{n\in\mathbb{N}}$ of L'-formulae, which will be called the sequence of ***Henkin axioms***, and define a sequence $\langle \bar{c}_n \rangle_{n\in\mathbb{N}}$ of Henkin constants, similarly to [3].

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 φ_n is an L'-formula. For $n \in \mathbb{N}$,

$$\theta_n :\equiv ([\bar{c}_n/x_n]\varphi_n) \mathrel{\dot{\to}} (\dot{\forall} x_n\ \varphi_n),$$

where $\bar{c}_n$ is the first of the new Henkin constant symbols not occurring in φ_n or θ_k for any $k < n$. Once the Henkin axioms are defined in that way, for any set Γ of L-formulae,

$$\Gamma \vdash \dot{\bot} \iff \{\theta_n \mid n \in \mathbb{N}\} \cup \upharpoonleft\Gamma \vdash \dot{\bot},$$

where $\dot{\bot} :\equiv \dot{\neg}(\dot{\forall} v_0\ (v_0 \doteq v_0))$. Finally, one can prove:

Theorem 6 (Countable Completeness)**.** For any L-formula b and any (possibly infinite) set X of L-formulae,

$$X \vDash b \implies X \vdash b.$$

Proof. Assume $X \nvdash b$. Take $\Gamma := \{\dot{\neg} b\} \cup X$. Then, $\Gamma \nvdash \dot{\bot}$. Hence, one can construct a maximally consistent set Δ of L'-formulae such that $\{\theta_n \mid n \in \mathbb{N}\} \cup \upharpoonleft\Gamma \subseteq \Delta$ and

$$\varphi \in \Delta \iff \Delta \vdash \varphi$$

for an arbitrary L'-formula φ. Now, define a structure $\mathfrak{A}$ of L' as follows: its domain of discourse is $\texttt{trm}\, L'$, where the equivalence relation on $|\mathfrak{A}|$ is defined by

$$t_1 \sim t_2 :\iff \Delta \vdash t_1 \doteq t_2.$$

The interpretations are given by $f^{\mathfrak{A}} := \vec{t} \mapsto f\ \vec{t}$, $c^{\mathfrak{A}} := c$, and $R^{\mathfrak{A}} := \vec{t} \mapsto \Delta \vdash R\ \vec{t}$. Then, for an arbitrary L'-formula φ, setting $\rho := i \mapsto v_i$, by induction on the depth of φ, one can show $\varphi \in \Delta \Leftrightarrow [\![\varphi]\!]^{\mathfrak{A}}_{\rho}$ with the axiom `classic`. By restricting $\mathfrak{A}$ to the original language L, one can derive $\Gamma \nvDash \dot{\bot}$, which contradicts the assumption $X \vDash b$. Therefore, $X \vdash b$. □

5 Conclusion

In this paper, a full formalisation of a first-order logic along with its semantics, deduction system, and the soundness and completeness theorems in the Coq proof assistant was presented. It was confirmed that only the law of excluded middle is used for the formalisation.

There are possibilities for future works. First, 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. Furthermore, the completeness theorem for first-order languages with cardinalities greater than $\aleph_0$ will also be proved.

Acknowledgements

I would like to thank Ph.D. Sewon Park for advice on the introduction and guidance on LaTeX, and Prof. Kwanghoon Choi for providing feedback on the introduction. I am also grateful to Soonwon Moon for a valuable insight on the proof of Fact 5, and to Chihyo Roh for correcting grammar.

References

[1] Robert Constable and Mark Bickford. Intuitionistic completeness of first-order logic. *Annals of Pure and Applied Logic*, 165(1):164–198, 2014. The Constructive in Logic and Applications.

[2] Ernesto Copello, Nora Szasz, and Álvaro Tasistro. Formal metatheory of the lambda calculus using stoughton's substitution. *Theoretical Computer Science*, 685:65–82, 2017. Logical and Semantic Frameworks with Applications.

[3] Herbert B. Enderton. *A Mathematical Introduction to Logic.* Academic Press, 2001.

[4] Yannick Forster, Dominik Kirst, and Dominik Wehr. Completeness theorems for first-order logic analysed in constructive type theory: Extended version. *Journal of Logic and Computation*, 31(1):112–151, January 2021. Saarland University, Saarland Informatics Campus, Saarbrücken, Germany. Institute for Logic, Language and Computation, University of Amsterdam, The Netherlands.

[5] Asta Halkjær From. A succinct formalization of the completeness of first-order logic. In *Proceedings of the 27th International Conference on Types for Proofs and Programs (TYPES 2021)*, volume 239 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

[6] Hugo Herberlin, SunYoung Kim, and Gyesik Lee. Formalizing the meta-theory of first-order predicate logic. *Journal of the Korean Mathematical Society*, 54(5):1521–1536, September 2017. 대한수학회.

[7] Danko Ilik. *Constructive Completeness Proofs and Delimited Control. (Preuves constructives de complétude et contrôle délimité).* PhD thesis, École Polytechnique, Palaiseau, France, 2010.

[8] KiJeong Lim. Fol-archived. `https://github.com/KiJeong-Lim/Fol-archived`.

[9] Russell SS O'Connor. *Incompleteness & completeness: formalizing logic and analysis in type theory.* Sl: sn, 2009.

[10] Youngju Song, Minki Cho, Dongjae Lee, Chung-Kil Hur, Michael Sammler, and Derek Dreyer. Conditional contextual refinement. *Proc. ACM Program. Lang.*, 7(POPL), January 2023.