

A TRUE THEOREM LEAN CAN STATE BUT CAN NEVER VERIFY

WILLIAM D. BANKS

ABSTRACT. We give an expository account of a mathematical result that the Lean proof assistant can formally *state* but, by Gödel’s second incompleteness theorem, can never *verify*, and which it also does not refute, namely, *the consistency of Lean’s own underlying type theory*. Working in ZFC extended by the existence of infinitely many inaccessible cardinals, we sketch the construction of a set-theoretic model of Lean’s Calculus of Inductive Constructions (with its universe hierarchy, quotient types, and choice), outline the proof of a soundness lemma, and conclude that no derivation of **False** exists. The argument is the model-theoretic consistency proof carried out in full by Carneiro, building on Werner’s set-theoretic interpretation of the same calculus; all substantive results are due to these authors, and this paper claims no new mathematics. We explain why this theorem lies outside Lean’s own reach, note that Gödel’s second incompleteness theorem has itself been verified by Lean’s kernel for first-order arithmetic, and discuss why every consistency proof ultimately rests on trust rather than proof.

1. INTRODUCTION AND STATEMENT OF RESULTS

1.1. Background. Lean [3] is a proof assistant, a computer program whose small, fully specified kernel mechanically checks every step of a formal mathematical proof. Lean’s logical rulebook is a dependent type theory, namely, the *Calculus of Inductive Constructions* (CIC) with a countable hierarchy of non-cumulative universes, definitional proof irrelevance, quotient types, and an axiom of definite description/choice (`Classical.choice`) [1].

The trustworthiness of such a system rests on a single property, which is of course *consistency*. In Lean’s language there is a distinguished proposition **False**, an inductive type that has no constructors; it is the canonical contradiction. Because the underlying logic validates *ex falso quodlibet*, a single derivation of **False** would make every proposition derivable and render the certification stamp utterly meaningless. Consistency is the assertion that this never happens.

1.2. Results. This paper presents that assertion as a theorem and proves it. The proof is not ours, rather it is the model-theoretic consistency proof carried out in detail by Carneiro [1], building on the tradition of set-theoretic models of type theory going back to Werner [7]. Our aim is a self-contained exposition of its architecture, followed by a careful discussion of its peculiar epistemic status. The theorem below is a true statement *about* Lean that Lean itself can formalize, can never prove, and does not disprove. Our main result is the following.

Date: September 13, 2026.

2020 Mathematics Subject Classification. 03B35, 03F40, 03B38, 03E55.

Key words and phrases. Lean, proof assistants, consistency, incompleteness, inaccessible cardinals, type theory.

THEOREM 1.1 (Consistency of Lean; Carneiro [1]). *Work in ZFC^+ , which is ZFC extended by the existence of infinitely many inaccessible cardinals. Then Lean’s type theory does not derive **False**; in other words, $\text{Con}(\text{Lean})$ holds.*

The hypothesis is essentially optimal (Remark 2.1), and Theorem 1.1 is proved in Section 4 by constructing a set-theoretic model of Lean’s type theory and establishing soundness. The theorem’s significance is framed by two further facts: Lean itself can never prove $\text{Con}(\text{Lean})$ (Theorem 3.1, proved in Section 3), and Lean does not refute it either (Theorem 4.2, proved in Section 4 alongside the main theorem). Together these show that $\text{Con}(\text{Lean})$ is independent of Lean. Section 5 discusses what has and has not been proved, and what has been machine checked.

1.3. Conventions. Precise definitions of the object theory, of the arithmetized sentence $\text{Con}(\text{Lean})$, and of the metatheory ZFC^+ are given in Section 2. Throughout, we use $\llbracket \cdot \rrbracket$ to denote the set-theoretic interpretation constructed in Section 4, V_α denotes the cumulative hierarchy, and $\vdash$ denotes kernel-level derivability in Lean.

2. PRELIMINARIES

2.1. The object theory. By *Lean’s type theory* we mean the kernel-level calculus that is described in [1], consisting of dependent function types (Π -types), a universe hierarchy $\text{Sort } 0 = \text{Prop}$, $\text{Sort } (n+1) = \text{Type } n$, universe polymorphism, inductive type families with their recursors, definitional proof irrelevance in Prop , together with the three axioms of propositional extensionality, quotient soundness, and choice. A *derivation* is a finite object checkable by a terminating algorithm; consequently the set of theorems of Lean is recursively enumerable, and derivations can therefore be coded by natural numbers in any theory interpreting elementary arithmetic; this includes Lean itself, which constructs $\mathbb{N}$ as an inductive type and proves (far more than) Peano arithmetic.

We write $\text{Con}(\text{Lean})$ for the arithmetized statement

"No natural number codes a kernel-accepted derivation of **False**."

2.2. The metatheory. Our metatheory is

$$\text{ZFC}^+ = \text{ZFC} + \text{“there exist infinitely many inaccessible cardinals.”}$$

Recall that a cardinal κ is said to be *inaccessible* if it is uncountable, regular, and a strong limit ($\lambda < \kappa \Rightarrow 2^\lambda < \kappa$). The role of inaccessibility is *closure*. For each inaccessible κ , the cumulative-hierarchy stage V_κ is closed under all the set-forming operations that Lean’s typing rules demand, namely, dependent products indexed by members of V_κ , least fixed points of the monotone operators arising from inductive specifications, quotients, and choice functions. Indeed, we have $V_\kappa \models \text{ZFC}$ (since it is a Grothendieck universe), and this is precisely why nothing weaker than a large-cardinal hypothesis will do. Lean proves $\text{Con}(\text{ZFC})$ (see [1]), and therefore by Gödel’s second incompleteness theorem, ZFC alone cannot prove $\text{Con}(\text{Lean})$.

REMARK 2.1 (Calibration). *The large-cardinal hypothesis of Theorem 1.1 is essentially optimal. Carneiro [1] calibrates the two theories level by level, with the fragment of Lean having $n+1$ universes proved consistent in ZFC plus $n+1$ inaccessibles, and conversely the fragment of Lean having $n+2$ universes proving the consistency of ZFC plus n inaccessibles; in particular Lean proves the consistency of every finite fragment of ZFC^+ . Thus Lean and ZFC^+ , the latter read as a schema, are equiconsistent. Both sit strictly below*

$\text{ZFC} + \text{“there is a Mahlo cardinal”}$ in consistency strength, a standard fact, since the inaccessible cardinals below a Mahlo cardinal are unbounded and the cumulative-hierarchy stage at such a cardinal therefore models ZFC^+ .

3. UNPROVABILITY

We first record the fact that frames the significance of Theorem 1.1.

THEOREM 3.1 (Unprovability; Gödel II). *If Lean is consistent, then Lean cannot prove the statement $\text{Con}(\text{Lean})$.*

Proof sketch. Lean is recursively axiomatized with decidable proof checking, and it interprets elementary arithmetic with enough strength to verify the derivability conditions of Hilbert, Bernays, and Löb for its own provability predicate. Gödel’s second incompleteness theorem [6] applies verbatim to any consistent theory of this kind and shows that no such theory proves its own canonical consistency statement. $\square$

4. PROOF OF THEOREM 1.1

The classical strategy is to construct a set-theoretic model of the calculus, prove soundness of the typing rules with respect to the model, and observe that **False** denotes the empty set.

4.1. Step 1: The interpretation of universes. Let $\kappa_0 < \kappa_1 < \kappa_2 < \dots$ be a fixed increasing sequence of inaccessible cardinals. Define

$$\llbracket \text{Sort } 0 \rrbracket = \llbracket \text{Prop} \rrbracket = \{\emptyset, \{\emptyset\}\}, \quad \llbracket \text{Sort } (n+1) \rrbracket = \llbracket \text{Type } n \rrbracket = V_{\kappa_n}.$$

Propositions are interpreted *proof-irrelevantly*, which means that each proposition is interpreted as one of exactly two sets, $\emptyset$ when the proposition is false in the model and the singleton $\{\emptyset\}$ when it is true, the sole element $\emptyset$ serving as its unique formal witness. This matches Lean’s definitional proof irrelevance, under which any two proofs of the same proposition are definitionally equal. Universe polymorphism is handled by interpreting a polymorphic constant as a family of interpretations indexed by assignments of natural numbers to its universe variables.

4.2. Step 2: The interpretation of terms and types. By recursion on (well-typed) syntax one assigns a set-theoretic denotation $\llbracket \Gamma \rrbracket$ to every context Γ and, for a given environment $\rho \in \llbracket \Gamma \rrbracket$, denotations $\llbracket T \rrbracket_\rho$ and $\llbracket e \rrbracket_\rho$ to every type T and term e in that context, as follows.

- *Dependent products.* The interpretation $\llbracket \Pi x:A. B \rrbracket_\rho$ is the set-theoretic dependent function space $\prod_{a \in \llbracket A \rrbracket_\rho} \llbracket B \rrbracket_{\rho[x \mapsto a]}$, with the convention that a product landing in **Prop** is collapsed to its proof-irrelevant truth value. Inaccessibility of κ_n ensures that this set again lies in V_{κ_n} whenever its data do.
- *Inductive types.* An inductive specification determines a monotone operator on V_{κ_n} ; its least fixed point exists because κ_n is regular and a strong limit, and interprets the inductive type. Constructors denote the evident injections and recursors denote definition by well-founded recursion on the fixed point. In particular,

$$\llbracket \text{False} \rrbracket = \emptyset \quad \text{and} \quad \llbracket \mathbb{N} \rrbracket = \omega.$$

- *Quotients.* For a binary relation r on a type A , the set $\llbracket \text{Quot } r \rrbracket$ is the genuine set quotient of $\llbracket A \rrbracket$ by the equivalence relation generated by $\llbracket r \rrbracket$; Quot.mk denotes the projection and Quot.lift the induced map, so the quotient axioms hold.
- *Choice.* The constant Classical.choice , whose type is

$$\prod \alpha : \text{Type } u. \text{Nonempty } \alpha \rightarrow \alpha,$$

is interpreted by a choice function on V_{κ_u} , which exists by the axiom of choice in the metatheory. Propositional extensionality holds because $\llbracket \text{Prop} \rrbracket$ has exactly two elements.

4.3. Step 3: Soundness.

LEMMA 4.1 (Soundness). *For every judgment derivable in Lean’s kernel, we have*

- (1) *if $\Gamma \vdash T : \text{Sort } u$ then $\llbracket T \rrbracket_\rho \in \llbracket \text{Sort } u \rrbracket$ for all $\rho \in \llbracket \Gamma \rrbracket$;*
- (2) *if $\Gamma \vdash e : T$ then $\llbracket e \rrbracket_\rho \in \llbracket T \rrbracket_\rho$ for all $\rho \in \llbracket \Gamma \rrbracket$;*
- (3) *if $\Gamma \vdash e \equiv e' : T$ (definitional equality) then $\llbracket e \rrbracket_\rho = \llbracket e' \rrbracket_\rho$ for all $\rho \in \llbracket \Gamma \rrbracket$.*

Proof sketch. By induction on derivations, checking each typing and conversion rule against the clauses of Step 2. The computation rules (β , η , ι for recursors, quotient computation) hold as literal equalities of sets under the interpretation. However, two points require genuine care, and these constitute the technical heart of [1].

Proof irrelevance. Definitional proof irrelevance makes Lean’s declarative definitional equality ill-behaved as an algorithm (it is undecidable, and the kernel’s algorithmic approximation is not transitive on all terms). The soundness proof therefore proceeds for an idealized, declarative presentation of the equality judgment, of which the kernel’s algorithm is shown to accept only instances; semantically, proof irrelevance is validated because $\llbracket \text{Prop} \rrbracket$ -inhabitants are subsingletons by construction.

Universe polymorphism and large elimination. Soundness is proved uniformly in universe-level assignments, and the eliminators of Prop -valued inductive types are checked to respect the collapsed, proof-irrelevant interpretation (Lean’s restriction of large elimination to syntactic subsingletons is exactly what makes this clause go through). $\square$

4.4. Step 4: Conclusion and consequences.

Proof of Theorem 1.1. Suppose on the contrary that Lean derives $\vdash e : \text{False}$ for some term e in the empty context. By Lemma 4.1(2), $\llbracket e \rrbracket \in \llbracket \text{False} \rrbracket$. By Step 2, $\llbracket \text{False} \rrbracket = \emptyset$. Thus $\emptyset$ has an element, contradicting the definition of $\emptyset$ in the metatheory. Hence no such derivation exists; that is, $\text{Con}(\text{Lean})$ holds. $\square$

THEOREM 4.2 (Non-refutability). *Work in ZFC^+ . Then Lean does not prove $\neg \text{Con}(\text{Lean})$.*

Proof. The negation $\neg \text{Con}(\text{Lean})$ is a Σ_1 arithmetic sentence, for it asserts the existence of a specific finite object, namely a code of a kernel-accepted derivation of False . By Step 2 the model interprets $\mathbb{N}$ as the standard ω , so the arithmetic sentences of Lean are evaluated over the standard natural numbers, and Lemma 4.1(2) shows that every arithmetic theorem of Lean is true there; in particular Lean proves no false Σ_1 sentence. By Theorem 1.1 the sentence $\neg \text{Con}(\text{Lean})$ is false, hence unprovable in Lean. $\square$

COROLLARY 4.3. *Work in ZFC^+ . Then $\text{Con}(\text{Lean})$ holds, whereas*

$$\text{Lean} \not\vdash \text{Con}(\text{Lean}) \quad \text{and} \quad \text{Lean} \not\vdash \neg \text{Con}(\text{Lean}).$$

$\text{Con}(\text{Lean})$ is thus a sentence in *Lean*'s own language that is provable in the metatheory but independent of *Lean* itself.

5. DISCUSSION: WHAT HAS AND HAS NOT BEEN PROVED

5.1. Why the theorem cannot be internalized. Theorem 3.1 is not an artifact of *Lean*'s design but a law governing every sufficiently strong formal system. Any proof checker capable of formalizing ordinary mathematics is thereby capable of formalizing its own syntax; Gödel's diagonal construction then manufactures the sentence whose unprovability is equivalent to consistency. The system can *state* its own reliability and can even verify the conditional $\text{Con}(\text{ZFC}^+) \rightarrow \text{Con}(\text{Lean})$ (implications of this shape are not blocked) but it can never discharge the antecedent. Adjoining new axioms (say, an inaccessible cardinal, stated type-theoretically) yields a stronger system that proves $\text{Con}(\text{Lean})$, but that system is then barred from proving *its* own consistency. The frontier moves, but it never closes.

5.2. The regress, stated honestly. The unconditional content of this paper is the implication

$$\text{Con}(\text{ZFC} + \text{infinitely many inaccessibles}) \implies \text{Con}(\text{Lean}),$$

together with the standard reasons for confidence in its hypothesis. Those reasons are the coherence of the intuitive picture of the cumulative hierarchy of sets, a century of set-theoretic practice in which no contradiction has ever appeared, and the calibration of Remark 2.1, which shows that the assumption is no stronger than the task demands. No consistency proof, by anyone, for any theory of this strength, does better. The consistency of the metatheory itself can be established only from a still stronger standpoint, and that standpoint from a stronger one yet, *ad infinitum*. Gentzen's proof of $\text{Con}(\text{PA})$ from transfinite induction up to ε_0 (see [5]) set the pattern, with consistency purchased in currency that is minted one level up, and Theorem 1.1 is its modern, large-cardinal descendant. The theorem therefore delivers not absolute certainty but the strongest substitute mathematics has found, a precise reduction of the reliability of a proof checker to the soundness of the classical foundations of set theory.

5.3. What has been machine checked. There is a coda that returns the story to *Lean* itself. Gödel's second incompleteness theorem, the engine of Theorem 3.1, has been formalized and verified in *Lean*. The Foundation development [4] contains kernel-checked proofs of both incompleteness theorems for arithmetic theories extending $\text{I}\Sigma_1$, so the very fact that bounds what *Lean* can certify about itself carries a certificate that *Lean*'s kernel accepts. The certificate stops exactly where Theorem 3.1 says it must. The formalized statement concerns first-order arithmetic, not *Lean*'s own type theory; instantiating it at *Lean* would require formalizing kernel derivability inside *Lean* and verifying the derivability conditions of Hilbert, Bernays, and Löb for it, a program begun in the Lean4Lean project [2] but not yet complete. We propose this as a natural continuation. A future development could certify Theorem 3.1 for *Lean* itself, while Theorem 1.1 would remain, provably, beyond its reach.

ACKNOWLEDGMENTS

The author thanks Ade Irma Suriajaya and Jesse Thorner for their comments on the original manuscript. He also thanks the Max Planck Institute for Mathematics in

Bonn for its support and excellent working conditions during the preparation of this paper. Part of this work was carried out during the workshop “Arithmetic, L-functions, and Pseudorandomness” (26w5546) at the Banff International Research Station, and the author thanks BIRS for its hospitality. Travel support from a Faculty International Travel award of the College of Arts and Science of the University of Missouri is gratefully acknowledged.

In accordance with the AMS policy statement on generative artificial intelligence, the author discloses that the large language model Claude (Anthropic) was used extensively in the preparation of this paper. The model produced initial drafts of much of the text, including the exposition of the proofs, and assisted with literature searches, proof checking, and brainstorming. The mathematical results are due to the authors cited in the text. The author directed and revised the entire manuscript, verified every mathematical claim and every reference, and takes full and sole responsibility for the final content.

DECLARATIONS

Funding. The author’s work was supported in part by a PIMS–BIRS grant, by the Max Planck Institute for Mathematics in Bonn, and by a Faculty International Travel award of the College of Arts and Science of the University of Missouri.

Competing interests. The author has no competing interests to declare that are relevant to the content of this article.

REFERENCES

- [1] M. Carneiro, The type theory of Lean. Master’s thesis, Carnegie Mellon University, 2019. (pp. [1](#), [2](#), and [4](#))
- [2] M. Carneiro, Lean4Lean: verifying a typechecker for Lean, in Lean. In: *31st International Conference on Types for Proofs and Programs (TYPES 2025)*, Leibniz Int. Proc. Inform., vol. 384, Schloss Dagstuhl, Dagstuhl, 2026, 2:1–2:23. (p. [5](#))
- [3] L. de Moura and S. Ullrich, The Lean 4 theorem prover and programming language. In: *Automated Deduction – CADE 28*, Lecture Notes in Comput. Sci., vol. 12699, Springer, Cham, 2021, 625–635. (p. [1](#))
- [4] The Formalized Formal Logic project, Foundation: formalization of mathematical logic in Lean 4. <https://github.com/FormalizedFormalLogic/Foundation>. (p. [5](#))
- [5] G. Gentzen, Die Widerspruchsfreiheit der reinen Zahlentheorie. *Math. Ann.* 112 (1936), 493–565. English translation: The consistency of elementary number theory, in M. E. Szabo (ed.), *The Collected Papers of Gerhard Gentzen*. Stud. Logic Found. Math., vol. 55, North-Holland, Amsterdam, 1969, 132–213. (p. [5](#))
- [6] K. Gödel, Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatsh. Math. Phys.* 38 (1931), 173–198. (p. [3](#))
- [7] B. Werner, Sets in types, types in sets. In: *Theoretical Aspects of Computer Software (TACS ’97)*, Lecture Notes in Comput. Sci., vol. 1281, Springer, Berlin, 1997. (p. [1](#))

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF MISSOURI, COLUMBIA, MO 65211, USA

Email address: bankswd@missouri.edu