New Foundations is consistent: an exposition and formal verification

Sky Wilshaw

April 21, 2024

Abstract

We give a self-contained account of a version of Holmes' proof [3] that Quine's set theory *New Foundations* [6] is consistent relative to the metatheory ZFC. We have formalised this proof in the Lean interactive theorem prover [7], and this paper is a 'deformalisation' of that work. We discuss the challenges of formalising new and untested mathematics in an interactive theorem prover, and how the process of completing the formalisation has influenced our presentation of the proof.

Contents

1 Overview

In §2, we will briefly discuss Lean [5], the interactive theorem prover in which our result is formalised. We will also explain why our formalisation in [7] can be trusted as evidence that Holmes' proof in [3] is correct, without needing to understand the underlying details of the proof. We have made frequent use of the community-made repository `mathlib` [2], which encodes standard mathematical definitions and theorems in Lean; without this, we would have needed to write our own libraries for (for example) abstract algebra and cardinal and ordinal arithmetic.

Lean is based on a version of the *calculus of constructions*, which is a dependent type theory. In order to authentically present the formalised proof, the mathematics of this paper will take place in this type theory, or some suitable variant of it. We will rarely make note of this choice, and readers are not expected to be familiar with such type theories. However, this will be relevant for some discussion sections, as some parts of the proof were made significantly harder by the fact that we are working in a type theory.

In §3, we will establish the mathematical context for the proof we will present. In particular, our proof will not directly show the consistency of NF; instead, we will construct a model of a related theory known as *tangled type theory*, or TTT. This is the result which has been formally verified: there is a

structure that satisfies a particular axiomatisation of TTT which we will discuss in §3. The expected conclusion that NF is consistent then follows from the fact that NF and TTT are equiconsistent [4].

[Finish the introduction…]

2 The Lean interactive theorem prover

2.1 Lean and its type theory

Lean [5] is a functional programming language and interactive theorem prover. As indicated in §1, its underlying logic is a dependent type theory based on the calculus of constructions. Carneiro proved in [1] that Lean's type theory is consistent relative to

$$\mathsf{ZFC} + \{\text{there are } n \text{ inaccessible cardinals} \mid n < \omega\}$$

These inaccessible cardinals are needed to support Lean's hierarchy of type universes. Higher universes are commonly used whenever they are convenient, for example in definitions of cardinals and ordinals. However, these uses are not strictly necessary for our purposes, and the entire proof can be carried out in plain ZFC, as shown by [3] and this paper.

Proofs in Lean may be written in its *tactic mode*, which tracks hypotheses and goals, and enables the use of *tactics* to update these hypotheses and goals according to logical rules. There are a large variety of tactics to perform different tasks, such as simplification (`simp`), rewriting of subexpressions (`rw`), structural induction (`induction`), and so on. These tactics output a *proof term*, which is a term in Lean's underlying type theory. The type of this term corresponds to the proposition that we intend to prove under the Curry–Howard correspondence. The proof term is then passed to Lean's *kernel*, which contains a type-checking algorithm. If the proof term generated by a tactic has the correct type, the kernel accepts the proof.

2.2 Trusting Lean

Lean is a large project, but one need only trust its kernel to ensure that accepted proofs are correct. If a tactic were to output an incorrect proof term, then the kernel would have the opportunity to find this mistake before the proof were to be accepted.

It is important to note that the kernel has no way of knowing whether a formal definition written in Lean matches the familiar mathematical definition. Any definitions used in a theorem statement must be manually checked by a human reader; all that Lean guarantees is that the conclusion is correct as written in its own type theory. For example, if verifying a formalised proof of Fermat's last theorem, one should manually check the definitions of natural numbers, addition, exponentiation, and so on, but need not check (for example) definitions and results about elliptic curves.

All of the proofs in this paper (except in §3, upon which no other results depend) are verified by Lean. To help with the verification step, our main result can be found in the `Result.lean` file (source, documentation). Each result is tagged with a hyperlink (such as ∀) to the documentation generated from the corresponding Lean code.

3 The theories at issue

References

[1] Mario Carneiro. *The Type Theory of Lean*. 2019. URL: https://github.com/digama0/lean-type-theory/releases.

[2] The mathlib Community. "The Lean Mathematical Library". In: *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*. CPP 2020. New Orleans, LA, USA: Association for Computing Machinery, 2020, pp. 367–381. ISBN: 9781450370974. DOI: 10.1145/3372885.3373824. URL: https://github.com/leanprover-community/mathlib4.

[3] M. Randall Holmes. *NF is Consistent*. 2023. arXiv: 1503.01406 [math.LO].

[4] M. Randall Holmes. "The Equivalence of NF-Style Set Theories with "Tangled" Theories; The Construction of ω-Models of Predicative NF (and more)". In: *The Journal of Symbolic Logic* 60.1 (1995), pp. 178–190. ISSN: 00224812. URL: http://www.jstor.org/stable/2275515.

[5] Leonardo de Moura and Sebastian Ullrich. "The Lean 4 Theorem Prover and Programming Language". In: *Automated Deduction – CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2021, pp. 625–635. ISBN: 978-3-030-79875-8. DOI: 10.1007/978-3-030-79876-5_37. URL: https://doi.org/10.1007/978-3-030-79876-5_37.

[6] W. V. Quine. "New Foundations for Mathematical Logic". In: *American Mathematical Monthly* 44 (1937), pp. 70–80. URL: https://api.semanticscholar.org/CorpusID:123927264.

[7] Sky Wilshaw, Yaël Dillies, Peter LeFanu Lumsdaine, et al. *New Foundations is consistent*. 2022–2023. URL: https://leanprover-community.github.io/con-nf/.