Carnegie Mellon University
Research Showcase @ CMU

Department of Philosophy | Dietrich College of Humanities and Social Sciences

2004

Notes on a Formalization of the Prime Number Theorem

Jeremy Avigad
Carnegie Mellon University, avigad@cmu.edu

Follow this and additional works at: http://repository.cmu.edu/philosophy

Part of the Philosophy Commons

This Technical Report is brought to you for free and open access by the Dietrich College of Humanities and Social Sciences at Research Showcase @ CMU. It has been accepted for inclusion in Department of Philosophy by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

Notes on a formalization of the prime number theorem*

Jeremy Avigad

September 10, 2004

1 Introduction

On September 6, 2004, using the Isabelle proof assistant, I verified the following statement:

```
(%x. pi x * ln (real x) / (real x)) ----> 1
```

The system thereby confirmed that the prime number theorem is a consequence of the axioms of higher-order logic together with an axiom asserting the existence of an infinite set. All told, our number theory session, including the proof of the prime number theorem and supporting libraries, constitutes 673 pages of proof scripts, or roughly 30,000 lines. This count includes about 65 pages of elementary number theory that we had at the outset, developed by Larry Paulson and others; also about 50 pages devoted to a proof of the law of quadratic reciprocity and properties of Euler's φ function, neither of which are used in the proof of the prime number theorem. The page count does not include the basic HOL library, or properties of the real numbers that we obtained from the HOL-Complex library.

The formalization was a collaborative effort. David Gray developed a substantial part of our number theory library, including basic facts about primes and multiplicity, the μ function, and a general inversion lemma. Kevin Donnelly and I worked together to develop a library to support the requisite big O calculations [4], and he derived a number of basic analytic identities. Paul Raff proved Chebyshev's theorem $\psi(x) = O(x)$, and did most of the work needed to prove the equivalence of statements of the PNT in terms of the functions π, θ, and ψ. For reasons indicated below, we chose to formalize a version of Atle Selberg's "elementary" proof.

Our efforts were designed to get us to the prime number theorem as quickly as possible rather than as cleanly as possible, and there is still a good deal of work to be done. In these preliminary notes, I would simply like to share some relevant information and initial thoughts.

*Carnegie Mellon Technical Report CMU-PHIL-163

2 The prime number theorem

For each natural number x, let $\pi(x)$ denote the number of primes less than or equal to x, and let log denote logarithm with base e. The prime number theorem states that $\pi(x)$ is asymptotic to $x/\log x$, i.e. that $\pi(x)\log x/x$ approaches 1 as x approaches infinity.

Gauss and Legendre both conjectured that this is the case, on the basis of computation, around the turn of the nineteenth century. In the 1850's, Chebyshev obtained the first significant advances towards proving it. In particular, he showed that each term is "big O" of the other, i.e. that $\pi(x)\log x/x$ is bounded above and below by positive constants (his estimates, in fact, yield specific bounds .89 and 1.1); and that *if* the expression has a limit, that limit must be 1. He was also introduced the functions ψ and θ and gave equivalent versions of the prime number theorem in terms of these. In a landmark work of 1859, Riemann introduced the complex-valued function ζ into the study of number theory, though it was not until 1894 that von Mangoldt provided an expression for ψ that reduced the prime number theorem, essentially, to showing that ζ has no roots with real part equal to 1. This last step was achieved by Hadamard and de la Vallée Poussin, independently, in 1896.

The resulting proofs make strong use of the theory of complex functions. In 1921, Hardy expressed strong doubts as to whether a proof of the theorem was possible which did not depend, fundamentally, on these ideas. In 1948, however, Selberg and Erdös found elementary proofs based on a "symmetry formula" due to Selberg. (The nature of the interactions between Selberg and Erdös at the time and the influence of ideas is a subtle one, and was the source of tensions between the two for years to come.) Since the libraries we had to work with had only a minimal theory of the complex numbers and a limited real analysis library, we chose to formalize the Selberg proof.

There are a number of good introductions to analytic number theory (see, for example, [1, 11]). Edwards' *Riemann's zeta function* [6] is an excellent source of both historical and mathematical information. (I have also found Havil's *Gamma: exploring Euler's constant* [8], written for a more general audience, to be enjoyable and informative.) A number of textbooks present the Selberg's proof in particular, including those by Nathanson [12], Shapiro [14], and Hardy and Wright [7]. We followed Shapiro's excellent presentation quite closely, though we made good use of Nathanson's book as well.

We also had help from another source. In [5], Cornaros and Dimitricopoulis showed that the prime number theorem is provable in a weak fragment of arithmetic, by showing how to formalize Selberg's proof (based on Shapiro's presentation).[1] Their concerns were different from ours: by working in HOL, we were allowing ourselves as logically stronger theory; on the other hand, Cornaros and Dimitricopoulis were concerned solely with axiomatic provability and not ease of formalization. Their paper was, however, quite helpful in stripping the proof down to its bare essentials. Also, since, our libraries did not have a good theory

[1] For issues relating to the formalization of mathematics, and number theory in particular, in weak theories of arithmetic, see my survey [3].

of integration, we had to take some care to avoid the mild uses of analysis in the textbook presentations.[2] Cornaros and Dimtricopoulis' paper was again often helpful in that respect.

3 Isabelle

Isabelle [18] is a generic proof assistant developed under the direction of Larry Paulson at Cambridge University and Tobias Nipkow at TU Munich. The HOL instantiation [13] provides a formal framework that is a conservative extension of Church's simple type theory with an infinite type (from which the natural numbers are constructed), extensionality, and the axiom of choice. Isabelle offers good automated support, including a term simplifier, an automated reasoner (which combines tableau search with rewriting), and decision procedures for linear and Presburger arithmetic. It is an LCF-style theorem prover, which is to say, one repeatly applies "tactics" to reduce a current subgoal to simpler ones; correctness is guaranteed by the fact that all tactics are ultimately built up from basic applications of the specified rules for the formal system. But Isabelle also allows one to take advantage of a higher-level proof language, similar to Mizar's, called Isar; this was implemented by Marcus Wenzel [16].

I just said that the Isabelle axiomatization is a "conservative extension" of simple type theory. Specifically, HOL extends ordinary type theory with set types, and a schema for polymorphic axiomatic type classes designed by Tobias Nipkow and implemented by Marcus Wenzel [15]. It also includes a definite description operator ("THE"), and an indefinite description operator ("SOME").[3]

As noted above, our formalization makes use of the basic HOL library. It also makes use of those parts of the HOL-Complex library, developed primarily by Jacques Fleuriot, that deal with the real numbers.

4 Time frame

A priori, one interesting feature of our formalization of the prime number theorem is simply its existence, which shows that current technology makes it

[2] Since the project began, Sebastian Skalberg managed to "port" the more extensive analysis library from the HOL theorem prover to Isabelle. By the time that happened though, we had already worked around most of the applications of analysis needed for the proof.

[3] The extension by set types is mild, since they are easily interpretable in terms of predicate types $\sigma \to bool$. Similarly, the definite description operator can be eliminated, at least in principle, using Russell's well-known interpretation. It is the indefinite description operator, essentially a version of Hilbert's epsilon operator, that gives rise to the axiom of choice. Though we occasionally used the indefinite description operator for convenience, these uses could easily be replaced by the definition description operator; and my guess is that uses in the libraries we relied on are similarly mild. In short, the reference to the axiom of choice above can be dispensed with. In any event, it is a folklore result that Gödel's methods transfer to higher-order logic to show that the axiom of choice is a conservative extension for a fragment the includes the prime number theorem.

possible to treat a proof of this complexity. The question naturally arises as to how long the formalization took.

This is a hard question to answer. I first decided to undertake the project in March of 2003, having learned how to use Isabelle and proved Gauss' law of quadratic reciprocity with David Gray and Adam Kramer the preceding summer and fall. But this was a side project for everyone involved, and time associated it includes time spent learning to use Isabelle, time spent learning the requisite number theory, and so on. David Gray made most of his subsequent contributions working a few hours per week in the summer of 2003, before his thesis work in ethics took over. Most of Kevin Donelly's contributions also came from half-time work during the summer of 2003, that is, the summer after his junior year at Carnegie Mellon. Paul Raff started working on the project in the 2003-2004 academic year, but much of that time was spent getting comfortable with Isabelle; most of his contributions came working roughly half-time in the summer of 2004, just after he obtained his undergraduate degree. Though my own involvement was more constant, I rarely put in more than a few hours per week before the summer of 2004, and set the project aside for long stretches of time. The bulk of my proof scripts (including everything from "MuSum.thy" onwards, "RealLnSum.thy," most of the proofs in "Inversion.thy," and a number of lemmas that have been moved to earlier libraries) were written during the summer of 2004, when I worked roughly half-time on the project from the middle of June to the end of August.

Some specific benchmarks may be more informative. Proving most of the inversion lemmas we needed, starting from David's general inversion formula "general-inversion-aux" in "Inversion.thy," took about a day. (For a "day" read eight hours of dedicated formalization. Though I could put in work-days like that for small stretches, keep in mind that, in some of the estimates below, the work was spread out over longer periods of time.) Proving the first form of the Selberg symmetry formula, i.e. the material through "selberg2" in "Selberg.thy," took another day. Along the way, I was often sidetracked by the need to prove elementary facts about things like primes and divisibility, or the floor function on the real numbers. This process stabilized, however, and towards the end I found that I could formalize about a page of Shapiro's text per day. Thus, the derivation of the main formula in "Error.thy," taken from pages 428–431 in Shapiro's book, took about three-and-a-half days to formalize; and the remainder of the proof, corresponding to 432–437 in Shapiro's book and the file "PrimeNumberTheorem.thy" took about five days. The increase in length is notable: the three-and-a-half pages of text associated with "Error.thy" translate to more than 1,600 lines of proof script, and the five pages of txt associated with "PrimeNumberTheorem.thy" translate to more than 4,000 lines of proof script.

I suspect that over the coming years the time requirements will drop significantly. Reflecting on the project, much of the time was spent on the following:

- Proving obvious, basic facts about the concepts involved.
- Proving trivial lemmas and spelling out "straightforward" inferences.

- Finding the right lemmas and theorems to apply.
- Entering long formulas and expressions correctly, and adapting ordinary mathematical notation to machine notation.

The first requirement will be ameliorated over time, since better libraries will gradually come into existence as verification efforts proceed. A theorem only has to be proved once, by anyone, anywhere; it won't be long before most elementary mathematics is in place. Of course, there are nontrivial concerns as to how to maintain the library, and, perhaps more important, how to share libraries between different formal and implementational frameworks. But these are issues that are being discussed. See, for example, the Logosphere project [19], the Mathweb project [21], the Mathscheme project [20], or just do a Google search on "Mathematical Knowledge Management."

The second bullet item above holds the most theoretical interest. An ordinary mathematics text proceeds with straightforward inferences that are obvious to a reader with sufficient mathematical experience, but currently need to be spelled out in much greater detail for formal verification. There is no reason that automated inference engines should not eventually be able to capture such inferences. Getting them to do so will require is a sustained reflection on the procedures by which we "see" that a statement follows from previous ones in domain-specific situations, and ongoing implementation and improvement of these mechanisms.

The third item is a database problem. When an inference can be supported by a well-known fact, the challenge is to find this fact efficiently. Isabelle produces browser pages with lists of theorems, and through the Proof General interface one can automatically search for theorems involving a certain list of symbols, or theorems that unify with the current goal. But there are likely to be better (though more involved) ways of finding relevant facts that are not so sensitive to representation (for example, capable of matching theorems up to easy logical equivalence, equations up to the associativity and commutatitivy of addition and multiplication and the symmetry of equality, etc.).

Finally, dealing with mathematical language and notation is nontrivial. Even with the care that Isabelle's designers have given to issues of syntax, reading and entering complicated formulas correctly can be a chore, especially when it involves spelling out details that are left to convention in an ordinary mathematical text (like coercions between types, or naming the relevant variable in a big O expression).

None of these, however, present any conceptual hurdles. In the long run, it should be possible to improve the ratio to three or four pages of mathematics per day. This is about what it currently takes to write up a result carefully, for publication. All it will take to get to this point is ongoing theorizing, engineering, experimentation, and hard work.

5 Formalization and rigor

Implicit in the last assessment is the assumption that the efforts described are worthwhile, i.e. that mathematics *should* be verified. Discussions of the role of formalization and rigor in mathematics tend to raise hackles (and I am sure I have raised some already). Often the issues are cast in terms of the difference between recognizing the importance deep ideas, creativity, and inspiration in mathematics, on the one hand, and recognizing the importance of precision, clarity, and correctness on the other. Such issues were famously raised Jaffe and Quinn's article " 'Theoretical mathematics': toward a cultural synthesis of mathematics and theoretical physics" [9], and the ensuing debate [2, 10]. They are also common topics of discussion on the FOM (Foundations of Mathematics) forum [17]; see, for example, Timothy Chow's posting of August 3, 2004.

Of course, it is possible to recognize the importance of *both* creativity and rigor in mathematics. Emphasizing the importance of one should not be interpreted as a denial of the importance of the other. The claim that formal mathematical verification is worthwhile rests only on the following two assumptions:

- Rigor is important to mathematics, both for communication of results and as a standard of correctness.
- Formalization is an appropriate standard of rigor. That is, formal systems provide an appropriate "specification language" for communicating results, and the existence of a formal proof provides a strong standard of correctness.

For the moment, we can set aside the more tendentious question as to *how* important rigor is, or how it is to be weighed against intuition and ideas.

"Rigor" in the first bullet item includes things like clarity and consistency of terminology, the precise statement of theorems, the correct application of prior results, and the conviction that there are no unfillable gaps in argumentation. I take it as relatively uncontroversial that these are important; otherwise, why do we take such pains to see to it that publications are checked carefully by referees? One may, on the other hand, question the extent to which informal standards of rigor are enforced by formalization. But clearly the two are at least related. Formal axiomatic systems have been carefully designed to model and clarify standards of mathematical correctness; and, pragmatically speaking, formalization often uncovers ambiguities in ordinary mathematical statements, apparently obvious details that turn out to be not-so-obvious on closer inspection, and special cases (base cases, exceptions) that require extra attention. In both theory and practice, then, it seems clear that formalization is at least affiliated with the informal notion of rigor.

Formal verification can also help guarantee correctness when, as is becoming increasingly common these days, proofs rely on computations that are too long to check by hand. Computer algebra systems are notoriously fraught with inconsistencies, though they are indispensable to a good deal of mathematical

research. And, without formal verification, it is hard to ensure that complex code is doing what the author intended. The proposed solution is to have such code not only perform the calculation, but also output a certificate witnessing correctness, one that can be transformed to a formal proof object in the target formal system.

These ideas form the basis for Tom Hales' flyspeck project, which aims for a fully verified form of his proof of the Kepler conjecture. This is by far the most ambitious project of its kind to date, since the informal proof involves hundreds of pages of ordinary mathematical text and has a substantial computational component. For details, see the web page:

http://www.math.pitt.edu/∼thales/flyspeck/

One may be concerned about the effects that formalization will have on exposition. Whereas a formally verified proof script may bolster the conviction *that* a theorem is true, it may do little to explain *why* the theorem is true. Good mathematical exposition conveys a more robust understanding of the main ideas behind a proof, the concepts and methods involved, and the ways that these concepts and methods function in the context of a broader theory. The worry is that if formal verification becomes a primary goal, important aspects of mathematical practice will fall by the wayside.

These concerns seem to me to be misplaced. Formal mathematical verification is by no means supposed to be a *substitute* for ordinary mathematical presentation. There will always be good and bad mathematical exposition, and the mathematical community will always face the challenge of ensuring that institutional incentives are in place that foster effective communication of mathematical understanding. The desire to ensure correctness as well is by no means antithetical to this.

One model for how formal and informal presentations can be combined is to have both prepared from a single source document. For example, Isabelle allows one to embed expository text into proof scripts, which is simply imported into a "session document" prepared from these scripts. One can similarly label parts of a proof uninteresting, thereby suppressing their inclusion in the final document. The advantage to this type of approach is that one can obtain informal expository presentations that come with a guarantee that the definitions and theorems are correct, exactly as stated. And if the reader gets stuck at a point in the argument, there is an associated resource that can at least provide a formal sequence of inferences that warrant the purported conclusion.

Others have questioned whether formal verification really provides absolute certainty. After all, inconsistencies have been found in even the most carefully designed systems, and there is always the possibility that the author has left a back door in the system (such as: if theorem = prime-number-theorem print 'verified!').

Such concerns seem to me to be relatively minor as well, especially since fairly simple mechanisms can be used that reduce the possibility of "cheating" or taking advantage of loopholes to virtually nil. For example, if you doubt the validity of my formalization, I can ask you to implement a proof checker for

higher-order logic and, without sharing the code, specify an appropriate input format. If I then make the effort to translate my proof object to one that meets your specification and passes your verification, you should be strongly convinced that what I have is, in fact, a valid proof. Of course, there is the possibility that I find that your specification does not adequately represent higher-order logic, but then the burden is on me to point out the problem. There is also the possibility that your checker finds fault with a proof that I claim meets your specification, but then the burden is on you (and your verifier) to show me where the proof has gone wrong. Such an exchange would represent a strong form of mathematical objectivity, and disputes like this should be easily resolvable. Passing a test like this with two or more independent verifiers would deliver about as much certainty as can be hoped for.[4]

To be sure, translating complex proof objects between different representations of the same formal system is by no means trivial in practice, and current technology has only begun to address this task. But it is, at least, straightforward in principle; and, again, all that stands between theory and practice is hard work and sound engineering.

6 Future plans

As happy as I am to have the formalization of the prime number theorem behind us, there is still a great temptation to go back now and "do it right." Our formalization of the prime number theorem has generated hundreds of fundamental lemmas and theorems, which should be polished, improved, and added to Isabelle's library infrastructure. The remaining proof scripts should then be commented and cleaned up significantly.

What we can learn from the formalization will ultimately be much more interesting than the formalization itself. Our efforts have provided a wealth of data that can be mined and used to improve the types of support that a proof assistant can offer. For example, an inordinate portion of our formal proof is dedicated to carrying out straightforward and tedious calculations involving equalities and inequalities on the real numbers. Even though in most cases the general theory is undecidable (the calculations may involve basic properties of transcendental functions or the floor function), I suspect that relatively simple heuristic procedures will be quite effective in handling such calculations.

There are moreover a number of aspects of the formalization that one can focus on, and then design domain-specific procedures that eliminate the need to spell out "straightforward" or "obvious" inferences. For example, Kevin Donnelly and I have designed an implemented a prototype algorithm to support inferences between big O equations (though we have not incorporated it into the Isabelle framework). I suspect that with even mild additions of this sort, it

[4] Another issue that may arise is the extent to which the formal statement of a theorem, and associated formal definitions, capture the informal ones. This issue can be resolved in a similar way: if you give me *your* formalizations, I should be able to either prove them equivalent to mine or clarify the discrepancies.

will be possible to reduce the current proof of the prime number theorem to a fraction of its size.

Of course, the question of generality arises: tools designed to simplify a proof of the prime number theorem may be of little use in algebraic geometry or functional analysis. But at least we can expect them to be useful in supporting similar proofs in analytic number theory, and carry over to nearby domains. After a while, we should be able to discern general features that characterize the types of methods that are effective in supporting mathematical reasoning more generally. We have to start with specific examples, and see what we can make of them; only after a number of case studies are in place will a general theory begin to emerge.

References

[1] Tom M. Apostol. *Introduction to analytic number theory.* Springer-Verlag, New York, 1976. Undergraduate Texts in Mathematics.

[2] Michael Atiyah and et al. Responses to: A. Jaffe and F. Quinn, "Theoretical mathematics: toward a cultural synthesis of mathematics and theoretical physics". *Bull. Amer. Math. Soc. (N.S.)*, 30(2):178–207, 1994.

[3] Jeremy Avigad. Number theory and elementary arithmetic. To appear in *Philosophia Mathematica.*

[4] Jeremy Avigad and Kevin Donnelly. Formalizing o notation in isabelle/hol. In David Basin and Michaël Rusinowitch, editors, *Automated Reasoning: second international joint conference, IJCAR 2004*, Lecture Notes in Artificial Intelligence 3097, pages 357–371. Springer Verlag, 2004.

[5] C. Cornaros and C. Dimitracopoulos. The prime number theorem and fragments of PA. *Arch. Math. Logic*, 33(4):265–281, 1994.

[6] H. M. Edwards. *Riemann's zeta function.* Dover Publications Inc., Mineola, NY, 2001. Reprint of the 1974 original [Academic Press, New York].

[7] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers.* Oxford, fifth edition, 1979.

[8] Julian Havil. *Gamma: exploring Euler's constant.* Princeton University Press, Princeton, NJ, 2003. With a foreword by Freeman Dyson.

[9] Arthur Jaffe and Frank Quinn. "Theoretical mathematics": toward a cultural synthesis of mathematics and theoretical physics. *Bull. Amer. Math. Soc. (N.S.)*, 29(1):1–13, 1993.

[10] Arthur Jaffe and Frank Quinn. Response to: "Responses to: A. Jaffe and F. Quinn, 'Theoretical mathematics: toward a cultural synthesis of mathematics and theoretical physics'". *Bull. Amer. Math. Soc. (N.S.)*, 30(2):208–211, 1994.

[11] G. J. O. Jameson. *The prime number theorem*, volume 53 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 2003.

[12] Melvyn B. Nathanson. *Elementary Methods in Number Theory*. Springer, New York, 2000.

[13] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL. A proof assistant for higher-order logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, 2002.

[14] Harold N. Shapiro. *Introduction to the theory of numbers*. Pure and Applied Mathematics. John Wiley & Sons Inc., New York, 1983. A Wiley-Interscience Publication.

[15] Markus Wenzel. Type classes and overloading in higher-order logic. In E. Gunter and A. Felty, editors, *Proceedings of the 10th International Conference on Theorem Proving in Higher Order Logics (TPHOLs'97)*, pages 307–322, Murray Hill, New Jersey, 1997.

[16] Markus Wenzel. *Isabelle/Isar — a versatile environment for human-readable formal proof documents*. PhD thesis, Institut für Informatik, Technische Universität München, 2002.

[17] Foundations of Mathematics online discussion forum. http://www.math.psu.edu/simpson/fom.

[18] The Isabelle theorem proving environment. Developed by Larry Paulson at Cambridge University and Tobias Nipkow at TU Munich. http://www.cl.cam.ac.uk/Research/HVG/Isabelle/index.html.

[19] The Logosphere project. http://www.logosphere.org/.

[20] The Mathscheme project. http://imps.mcmaster.ca/mathscheme/.

[21] The Mathweb project. http://www.mathweb.org/.