Understanding, formal verification, and the philosophy of mathematics*

Jeremy Avigad

August 23, 2010

Abstract

The philosophy of mathematics has long been concerned with determining the means that are appropriate for justifying claims of mathematical knowledge, and the metaphysical considerations that render them so. But, as of late, many philosophers have called attention to the fact that a much broader range of normative judgments arise in ordinary mathematical practice; for example, questions can be interesting, theorems important, proofs explanatory, concepts powerful, and so on. The associated values are often loosely classified as aspects of "mathematical understanding."

Meanwhile, in a branch of computer science known as "formal verification," the practice of interactive theorem proving has given rise to software tools and systems designed to support the development of complex formal axiomatic proofs. Such efforts require one to develop models of mathematical language and inference that are more robust than the the simple foundational models of the last century. This essay explores some of the insights that emerge from this work, and some of the ways that these insights can inform, and be informed by, philosophical theories of mathematical understanding.

1 Introduction

Since Plato and Aristotle, the philosophy of mathematics has been concerned with clarifying the nature of mathematical objects, and determining the appropriate means of justifying claims of mathematical knowledge. But in our

*This essay is based on a talk presented in Paris in January, 2010, to a fellows seminar in Mic Detlefsen's *Ideals of Proof* project; and in February, 2010, in the University of Pittsburgh's *Center for Philosophy of Science* colloquium series. I am grateful to members of both audiences for comments and discussion after the talk. Comments from Anil Gupta, Tom Ricketts, and Mark Wilson in particular have influenced the presentation in Section 4. Section 4.3 incorporates material I presented in other workshops in Paris in May of 2009 and January of 2010, where I received helpful comments from Andrew Arana, Alan Baker, Karine Chemla, Evelyn Fox Keller, Paolo Mancosu, and Marco Panza, among others. Discussions with Penelope Maddy have also influenced the writing of that section. Finally, I am grateful to an anonymous referee for corrections and helpful suggestions. This work has been partially supported by NSF grant DMS-0700174 and a grant from the John Templeton Foundation.

daily mathematical practice, we often employ normative judgments that go beyond assessments of justification and correctness. For example, mathematical questions can be interesting, or not; questions can be natural; concepts can be fruitful or powerful; some proofs provide better explanations than others; some historical developments are important; and some observations are insightful. Even though our ways of expressing these judgments are often vague and imprecise, the evaluations matter to us a great deal. They bear on the kinds of mathematics we decide to do and they way we go about doing it, they way we teach and communicate mathematics, and the kinds of mathematics we praise and support. In other words, such judgments amount to normative assessments in one of our core scientific practices, and so deserve philosophical attention.

Intuitively, what unifies these kinds of judgments is that they evaluate the extent to which pieces of mathematics—concepts, proofs, questions, conjectures, theories, and so on—contribute to our *understanding*. This last word is often used to invoke an aura of mystery and ineffability, suggesting that once we look beyond well-worn questions of justification our modes of hard-nosed philosophical analysis break down entirely, leaving us with nothing to do but shake our heads in wonder. The point of this essay is to argue to the contrary. Specifically, I will consider some recent scientific advances in the formal modeling of mathematical reasoning and proof, and argue that these advances give us some leverage in making sense of such normative assessments.

The outline of this paper is as follows. In Section 2, I briefly explore some of our (vague) intuitions as to what we are talking about when we talk about mathematical understanding. In Section 3, I shift gears and discuss some technical developments in a branch of computer science known as "formal verification." More specifically, I will discuss efforts in interactive theorem proving, which involves the use of computational proof assistants to construct complex mathematical proofs. I will describe some of the ways the field has been forced to model mathematical language and inference and, in each case, I will consider what these efforts have to tell us about mathematical understanding. Finally, in Section 4, I will try to bring the vague intuitions and the technical work closer together, and explore some of the ways that philosophical work can inform, and be informed by, the technical developments. In particular, I will consider the ways that better a philosophical understanding of mathematical methods and abilities, mathematical concepts, and mathematical ease and difficulty can both help us come to terms with the kinds of evaluations expressed above, and support scientific work in fields that rely implicitly on notions of mathematical understanding.

I am by no means the only one now trying to make sense of normative assessments in ordinary mathematical practice; see, for example, the collection of essays *The Philosophy of Mathematical Practice* [21] and the references there for an overview of some of the recent work in the area. This essay draws on and expands prior writings of my own, including [2, 3].

2 Understanding

Section 1 drew a distinction between traditional philosophical concerns regarding correctness and justification on the one hand, and a broader class of normative assessments on the other. I will now pose two philosophical "problems" that help make some of the issues salient. The first, which I will call "the problem of multiple proofs," goes like this. On the standard account, the value of a mathematical proof is that it warrants the truth of the resulting theorem. Why, then, is it often the case that a new proof of a theorem is often highly valued? For example, Furstenberg's ergodic-theoretic proof [8] of Szemerédi's theorem [29] is recognized as a seminal element of ergodic Ramsey theory, and Tao [30] notes that a host of proofs of the theorem have been published since then, each one providing new insights. Clearly the proof of a theorem does *something* more than establish that the final result is true; can we say, in precise terms, what that something is?[1]

The second philosophical problem, which I will call "the problem of conceptual possibility," is as follows. It is often said that some mathematical advance was "made possible" by a prior conceptual development. For example, Riemann's introduction of the complex zeta function and the use of complex analysis made it possible for Hadamard and de la Vallée Poussin to prove the prime number theorem in 1896. What is the sense of "possibility" here? It is certainly not physical possibility, that is, the claim that someone like Chebyshev was physically incapable of writing down a proof in 1850. And it is not hard to make the case that nor is it a matter of logical possibility, that is, the fact that Chebyshev's axioms and principles of inference were not strong enough to entail the desired conclusion (see, for example, [1]). An intuitive answer is that Chebyshev did not have the right definitions in place, but that just pushes the problem back to explaining why he could not have written down those definitions. In other words, answering the question requires us to adopt a viewpoint in which writing down a good definition can be a hard thing to do.

In both cases, the answer seems to have something to do with understanding: new proofs provide us with a better understanding of the theorem they prove, as well as the surrounding mathematics; and historical developments provide us with understanding that supports further advances. Indeed, informal talk about mathematical understanding arises in a number of scientific and academic pursuits. Educational research aims to determine the ways of communicating understanding to students efficiently and effectively; psychology and cognitive science aim to explain how subjects acquire mathematical understanding; the history of mathematics focuses on events that have have furthered our understanding; in formal verification and automated reasoning, one tries to get

[1] The question may bring to mind Georg Kreisel's "unwinding program," which involves the use of formal methods to extract additional information from mathematical proofs. This has become a fruitful and active branch of proof theory, which now generally goes by the name of "proof mining" (see [6, 19]). With regard to the evaluation of informal proofs, information that is "implicit" in such proofs is certainly an important part of the story; see, for example, the discussion in [2].

computers to understand the mathematics we give them.

The purpose of this essay is to explore the prospect of developing a philosophical theory that can help us come to terms with these various notions of understanding. In order to avoid misunderstandings, I would like to make three points clear.

First, I am not claiming originality or priority in raising these issues. For example, one can find the problem of multiple proofs neatly expressed in the writings of Wittgenstein [37, III–60]:

> It might be said: "—that every proof, even of a proposition which has already been proved, is a contribution to mathematics". But why is it a contribution if its only point was to prove the proposition? Well, one can say: "the new proof shews (or *makes*) a new connexion"

A number of people working in the philosophy of mathematics today have come at these issues in various ways, and it is not possible for me to provide an adequate survey of such work here. For further reading, see the suggestion at the end of Section 1.

Second, I would like to emphasize that the "problems" I have raised are *not* great mysteries, set out only for us to marvel at how deeply inscrutable mathematical understanding is. On the contrary, we have a lot of good intuitions to start with, and it is easy to begin enumerating reasons why we might prefer one proof to another, or why we might value a historical development. The point is simply that, until recently, these issues have not received serious philosophical attention, and so the language we use to discuss them is still vague and imprecise. The challenge is to sharpen our intuitions so that they can better support rational discussion and scientific inquiry.

Finally, let us not get too hung up on the word "understanding." In ordinary usage, the word has social and even moral connotations; for example, we praise children for understanding and hold criminals liable only insofar as they have understood the consequences of their actions. Here I am only concerned with much more focused issues having to do with the methodology of mathematics. I have invoked the word "understanding" because it is often used in our informal talk about these issues, but what I am arguing for is the importance and promise of a certain type of philosophical analysis, rather than an exhaustive and univocal analysis of the notion of understanding as it applies in every domain. I do not mind if you prefer to characterize the project I am describing here as developing a theory of "mathematical values," "mathematical competence," "mathematical ability," or something of that sort. In short, I wish to focus on the phenomena, not the word.

Let us begin with some straightforward observations. Mathematics is hard; mathematical solutions, proofs, and calculations involve long sequences of steps, that have to be chosen and composed in precise ways. The problem is not that there are too few options, but too many. For example, at each stage in a deduction or calculation there are arbitrarily many facts we can interpolate from our background knowledge ($2 + 2 = 4$, $4 + 4 = 8$, ...), most of which will

be no help at all. From among all the options available to us, we have to settle one initial step that may plausibly take us closer to our goal, and then another, and then another. To compound matters, even the best among us have limited cognitive capacities; we can only keep so many pieces of information in mind at one time, and anticipate only a small number of consequences of the definitions and facts before us. It should strike you as something of a miracle that we are able to do mathematics at all. And yet, somehow, we are; being mathematically competent means being able to proceed reasonably, if imperfectly, under these circumstances. What I would like to understand are the complex mechanisms that make it possible for us to do so.

One way of posing the challenge is to note that whereas logic and traditional foundational research aims to determine what is *allowed* in a mathematical argument or calculation, this falls short of determining which steps are *appropriate*, or likely to be *fruitful*, in a given situation. This distinction was neatly expressed by Poincaré in 1908, in *Science et méthod* [24, Book II, Chapter II][2]:

> Logic teaches us that on such and such a road we are sure of not meeting an obstacle; it does not tell us which is the road that leads to the desired end.

In other words, logic tells us how to verify that a proof of a given theorem or the solution to a given problem is correct, but it does not tell us how to find such a solution or proof in the first place. Something more is needed to explain how we find manage to select a fruitful path from among a bewildering array of useless options:

> Discovery consists precisely in not constructing useless combinations, but in constructing those that are useful, which are an infinitely small minority. Discovery is discernment, selection.

While the image of finding a selecting a path towards our goals provides a helpful metaphor, I find literary metaphors helpful as well. For example, Herman Melville's *Moby Dick* is largely a story of humankind's attempts to come to terms with a chaotic and indifferent universe; this image accords well with mathematics since, after all, mathematics doesn't really care whether we understand it or not. One of the most difficult aspects of doing mathematics is sitting down to a blank sheet of paper, and trying to figure out where to begin. Blankness, as a metaphor, comes up often in *Moby Dick*; for example, in the final pages, the great white whale presents a "blank forehead" to the ship. The following passage is taken from an entire chapter, Chapter 42, devoted to a discussion of the color white.

> But not yet have we solved the incantation of this whiteness, and learned why it appeals with such power to the soul; and more strange and far more portentous... and yet should be as it is, the intensifying agent in things the most appalling to mankind.

[2]The next two passages are also quoted in [3].

> Is it that by its indefiniteness it shadows forth the heartless voids and immensities of the universe, and thus stabs us from behind with the thought of annihilation, when beholding the white depths of the milky way? Or is it, that as in essence whiteness is not so much a colour as the visible absence of colour; and at the same time the concrete of all colours; is it for these reasons that there is such a dumb blankness, full of meaning, in a wide landscape of snows–a colourless, all-colour of atheism from which we shrink?

Melville also offers us a grand and eloquent account of what happens when we get an unfiltered glimpse of the infinity of possibilities before us, with the story of Pip, one of the ship's deck hands, who falls overboard while his fellow shipmates sail off in chase of a whale.

> The sea had jeeringly kept his finite body up, but drowned the infinite of his soul. Not drowned entirely, though. Rather carried down alive to wondrous depths, where strange shapes of the unwarped primal world glided to and fro before his passive eyes; and the miser-merman, Wisdom, revealed his hoarded heaps; and among the joyous, heartless, ever-juvenile eternities, Pip saw the multitudinous, God-omnipresent, coral insects, that out of the firmament of waters heaved the colossal orbs. He saw God's foot upon the treadle of the loom, and spoke it; and therefore his shipmates called him mad. So man's insanity is heaven's sense; and wandering from all mortal reason, man comes at last to that celestial thought, which, to reason, is absurd and frantic; and weal or woe, feels then uncompromised, indifferent as his God.

These passages give us a good sense of what we are up against. Vast and complex, mathematics offers us great riches, but, at the same time, threatens to overwhelm us. A theory of mathematical understanding should explain how we cope with the complexity and maintain our sanity while exploring the wonders before us.

3 Formal verification

The phrase "formal verification" refers to a branch of computer science which uses formal methods to verify correctness. This can mean verifying the correctness of hardware and software design, for example, to ensure that a circuit description, an algorithm, or a network or security protocol meets its specification. But it can also mean verifying that a proof of a mathematical theorem is correct. There is a lot of overlap between these two pursuits, but also a number of differences in emphasis. Here I will focus on the latter type of verification.

"Interactive theorem proving" provides on important approach. Working with an interactive proof assistant, users enter enough information for the system to confirm that there is a formal axiomatic proof of the theorem that the

user has asserted. In fact, many systems enable one to extract a formal proof object—a complex piece of data representing a fully detailed axiomatic proof—which can be manipulated and verified independently of the system that constructed it.

There are a number of such systems currently in use; those in which substantial portions of mathematics have been formalized include Mizar, HOL, HOL light, Isabelle, Coq, and ACL2 (see [35] for an overview). The technology is still young, and it will be a while before such systems are commonly used in mathematical circles. But initial achievements make it clear that the potential is there. Notable theorems of mathematics that have been formalized to date include the four-color theorem [10], the prime number theorem [5, 16], Dirchlet's theorem on primes in an arithmetic progression [15], and the Jordan curve theorem [13]. At the time of writing of this article, two very ambitious projects are well underway: Thomas Hales is heading a project [14] to verify a proof of the Kepler conjecture, which asserts that there is no way of filling space with spheres that can beat the density of the familiar lattice packing; and Georges Gonthier is heading a similar project [12] to verify the Feit-Thompson theorem, which asserts that every finite group of odd order is solvable. Once again, I do not have sufficient space to provide an adequate overview of the field and its history; a good starting point for that is the December 2008 issue of the *Notices of the American Mathematical Society*, a special issue on formal proof, with articles by Hales, Freek Wiedijk, John Harrison, and Gonthier. My goal here is to consider some of the issues that arise with respect to formal verification, and what they have to tell us about mathematical understanding.

3.1 Understanding mathematical language

To start with, an interactive proof system relies on an underlying formal framework, which specifies a language in which assertions are to be expressed and the admissible rules of inference. There are a number of frameworks currently in use. Zermelo-Fraenkel set theory has long been recognized as a powerful foundational framework for mathematics, and, for example, the Mizar system uses a variant thereof. In the language of set theory, everything is a set; but one introduces definitions that allow one to recognize some sets as being natural numbers, some as being real numbers, some as being functions from the natural numbers to the reals, and so on. Other proof systems, in contrast, use frameworks that take such "typing" information to be built into the basic language. For example, HOL, HOL light, and Isabelle use a formulation of higher-order logic in Church's simple type theory, in which every term is assigned such a type. What makes simple type theory "simple" is that the type of a variable cannot depend on a parameter. On the other hand, in many mathematical contexts, it is natural to let a variable x stand for an element of the vector space $\mathbb{R}^n$, where n is another variable ranging over the natural numbers. Some systems, like Coq, use a more elaborate type theory, where $\mathbb{R}^n$ can be represented as a type. Adding rules to manipulate these more elaborate complicates the underlying logical framework. But, as we will see below, this kind of information

is fundamental to mathematical reasoning, and making it part of the underlying logical framework means that one can build general-purpose mechanisms to handle such information into the core of the system itself. (Coq is moreover based on a constructive logic, and computational aspects of the mathematics in question play a central role in the formalization process.)

As an example of how mathematics is expressed in such systems, here is Hales' statement of the Jordan curve theorem in HOL light:

```
!C. simple_closed_curve top2 C ==>
  (?A B. top2 A /\ top2 B /\
    connected top2 A /\ connected top2 B /\
  ~(A = EMPTY) /\ ~(B = EMPTY) /\
   (A INTER B = EMPTY) /\ (A INTER C = EMPTY) /\
       (B INTER C = EMPTY) /\
      (A UNION B UNION C = euclid 2)
```

Here, the exclamation point denotes a universal quantifier, and the question mark denotes an existential quantifier. The predicate "top2" denotes the standard topology on the Euclidean plane. In ordinary mathematical language, the expression above asserts that if C is a simple closed curve in the Euclidean plane, then the entire plane can be written as a disjoint union $A \cup B \cup C$, where A and B are connected open sets.

While one can get used to such typography and notation, it is less pleasant than reading ordinary mathematical text. As part of his MS thesis work at Carnegie Mellon, Steve Kieffer implemented a parser for an extension of set theory designed by Harvey Friedman, and entered hundreds of definitions from Suppes' *Set theory* and Munkres' *Topology* (see [18]). For example, here is his rendering of Munkres' definition of the topology X generated by a basis $\mathscr{B}$:

```
Definition MunkTop.13.2: 2-ary function Basisgentop.
If TOPBASIS[\mathscr{B},X] then
Basisgentop(\mathscr{B},X) \simeq (!\mathscr{T} \subseteq \wp(X))(
  (\forall U \subseteq X)(U \in \mathscr{T} \iff
    (\forall x \in U)(\exists B \in \mathscr{B})(
      x \in B \wedge B \subseteq U))).
```

And here is his definition of a certain topology, the K-topology, on the real numbers:

```
DEFINITION MunkTop.13.3.c: 0-ary function Krealtop. Krealtop \simeq
Basisgentop(
  Stdrealtopbasis \cup
  {V \subseteq \mathbb{R} :
    (\exists W \in Stdrealtopbasis)(
      V = W \less {Incl_{FrR}(1_{N}/n) : n \in \mathbb{N}})},
  \mathbb{R}).
```

These may not look like much, but they do come close to the structure of ordinary mathematical language. To make this point, Kieffer added a feature which allows the user to specify natural-language equivalents for the symbols in the language, and implemented a simple heuristic to determine when to use symbols or the expanded language. With these in place, the definitions above were rendered as follows:

Definition: If $\mathscr{B}$ is a basis for a topology on X then *the topology on X generated by $\mathscr{B}$* is the unique $\mathscr{T} \subseteq \wp(X)$ such that for every $U \subseteq X$, $U \in \mathscr{T}$ if and only if for every $x \in U$, there exists $B \in \mathscr{B}$ such that $x \in B$ and $B \subseteq U$.

Definition: *The K-topology on* $\mathbb{R}$ is the topology on $\mathbb{R}$ generated by the standard basis for a topology on $\mathbb{R}$ union the set of $V \subseteq \mathbb{R}$ such that there exists W in the standard basis for a topology on $\mathbb{R}$ such that $V = W \setminus \{1/n : n \in \mathbb{N}\}$.

The prose is not literary and tends to have a run-on feel, but it is not terribly far from ordinary mathematical text. What this seems to suggest is that our conventional modeling of mathematical language is on the right track. And insofar as this modeling captures the structure of mathematical language, it follows that when we read and write mathematical assertions, we are implicitly able to recognize and make use of this structure. Thus:

> Understanding mathematical language, involves, in part, being able to identify the fundamental logical and mathematical structure of an assertion, that is, recognize logical connectives and quantifiers, function application, predication, and so on.

3.2 Understanding mathematical proof

Those who work in interactive theorem proving are attuned to the fact that representing mathematical arguments requires not only an "assertion language," but a "proof language" as well. This fact is often glossed over in conventional logic texts, where a formal mathematical proof typically amounts to little more than a sequence of assertions. But ordinary textbook proofs have a lot more structure than that. It is sometimes helpful to think of ordinary mathematical proofs as being higher-level descriptions of low-level formal axiomatic proofs, or recipes for constructing such proofs. In fact, in the field of interactive theorem proving, it is common to refer to the user's input as "code."

For example, here is a formal proof, in the Isabelle proof assistant, of the statement that if n is any natural number not equal to 1, then n is divisible by a prime.

```
lemma prime_factor_nat: "n ~= (1::nat) ==> EX p. prime p & p dvd n"
  apply (induct n rule: nat_less_induct)
  apply (case_tac "n = 0")
  using two_is_prime_nat apply blast
  apply (case_tac "prime n")
```

```
  apply blast
  apply (subgoal_tac "n > 1")
  apply (frule (1) not_prime_eq_prod_nat)
  apply (auto intro: dvd_mult dvd_mult2)
done
```

The first line contains a statement of the lemma to be proved. This statement becomes the current goal; subsequent lines then apply formal rules and procedures that reduce the goal to simpler ones. For example, the first line applies a form of induction, which then requires the user to prove the statement for a given natural number, n, assuming that it holds of smaller ones. The second line splits on cases, depending on whether n is 0 or not; the first case is easy dispensed with using the previously established fact that 2 is prime. (The procedure "blast" is a generic automated routine that fills in the details.) If n is not 0, the fact that it is not 1 implies that it is greater than 1, in which case on applies the previously established fact that any number greater than 1 that is not prime can be written as a product of two strictly smaller numbers, at which point the inductive hypothesis applies.

What makes this "proof script" hard to read is that the text only gives the *instructions* that are used to act on the current goals; one has to "replay" the proof with the assistant to see the goals evolve. Fortunately, Isabelle also allows one to use a proof language called Isar [34] (modeled after Mizar's proof language [27]), which makes intermediate goals explicit. Here is a proof of the same lemma in Isar:

```
lemma prime_factor_nat:
  fixes n :: nat
  assumes "n ~= 1"
  shows "EX p. prime p & p dvd n"
proof (induct n rule: less_induct_nat)
  fix n :: nat
  assume "n ~= 1" and
    ih: "ALL m < n. m ~= 1 --> (EX p. prime p & p dvd m)"
  then show "EX p. prime p & p dvd n"
  proof -
  { assume "n = 0"
    moreover note two_is_prime_nat
    ultimately have ?thesis by auto }
  moreover
  { assume "prime n" then have ?thesis by auto }
  moreover
  { assume "n ~= 0" and "~prime n"
    with `n ~= 1` have "n > 1" by auto
    with `~prime n` and not_prime_eq_prod_nat obtain m k where
      "n = m * k" and "1 < m" and "m < n" by blast
    with ih obtain p where "prime p" and "p dvd m" by blast
    with `n = m * k` have ?thesis by auto }
```

```
  ultimately show ?thesis by blast
qed
```

Other proof languages are designed with different desiderata in mind. For example, here is a proof written in a language called Ssreflect [11], which is designed to be used with the Coq proof assistant. In the following theorem, known as the Burnside normal complement theorem, p denotes a prime number and S is assumed to be a Sylow p-subgroup of G. I have only shown the first few lines of the proof.

```
Theorem Burnside_normal_complement :
  'N_G(S) \subset 'C(S) -> 'O_p^'(G) ><| S = G.
Proof.
move=> cSN; set K := 'O_p^'(G); have [sSG pS _] := and3P sylS.
have [p'K]:  p^'.-group K /\ K <| G by rewrite pcore_pgroup pcore_normal.
case/andP=> sKG nKG; have{nKG} nKS := subset_trans sSG nKG.
have{pS p'K} tiKS: K :&: S = 1 by rewrite setIC coprime_TIg ?(pnat_coprime pS).
suffices{tiKS nKS} hallK: p^'.-Hall(G) K.
  rewrite sdprodE //= -/K; apply/eqP; rewrite eqEcard ?mul_subG //=.
  by rewrite TI_cardMg //= (card_Hall sylS) (card_Hall hallK) mulnC partnC.
```

The language is not for the faint-hearted. It is, however, remarkably efficient for writing proofs, allowing one to combine a number of small steps naturally into one line of code.

While there are striking differences between these various proof languages, there are also many features in common. Ordinary mathematical proofs call upon us to perform many different types of reasoning. At any point in a proof, we may be unwrapping hypotheses or establishing small facts that set the context for the subsequent proof; we may be applying previous lemmas or theorems and checking that the side conditions are satisfied; we may be unfolding a definition, or naming an object asserted to exist; we may be carrying out a calculation; and so on. Any proof language that aims to capture ordinary mathematical argumentation has to have mechanisms that allow one to carry out these steps, and, conversely, the formal mechanisms that are designed to allow one to do this efficiently helps shed light on what is necessary to read and write ordinary mathematical proofs.

> Understanding mathematical proof involves, in part, being able to recognize contextual cues, explicit or implicit reliance on local assumptions, background knowledge, recently established facts, and so on; and to determine whether inferences are a matter of calculation, unwrapping definitions, applying a lemma, etc.

3.3 Understanding mathematical domains and structures

Let us engage in with a little exercise. Suppose we know that z be a complex number satisfying $|z| \leq 1$, and we want to bound the absolute value of e^z. We

might start expanding e^z as a Taylor series as follows:

$$|e^z| = \left|\sum_{i=0}^{\infty} \frac{z^i}{i!}\right| \leq 1 + |z| + \left|\sum_{i=2}^{\infty} \frac{z^i}{i!}\right| \leq \ldots$$

In this expression, what type of object is i? z^i? 1? What does the division symbol denote? The symbol $\leq$? The summation symbol?

On inspection, we see that the variable i indexes a sum, so it ranges over the nonnegative integers. Since z is a complex number, so is z^i. We then divide z^i by the integer $i!$; this is possible because $i!$, an integer, can also be viewed as a complex number, and we can divide complex numbers. But taking the absolute value returns a real number; thus the symbol "1" here denotes the corresponding real number. Indeed, The ordering relation doesn't make sense on the complex numbers; so $\leq$ *has* to be viewed as a comparison between real numbers. As far as the summation symbol is concerned, keep in mind that in the expression $\sum_{i=0}^{\infty} \frac{z^i}{i!}$, i is a dummy variable, which is to say, writing $\sum_{j=0}^{\infty} \frac{z^j}{j!}$ does not change the value of the expression. One way to analyze the notation is to view the inner expression as denoting the *function* which maps any integer, i, to $\frac{z^i}{i!}$. Summation then becomes a higher-order operator, which takes a function as an argument.

What is interesting is that we are typically not mindful of these subtle issues when reading and working with expressions like the one above. Mathematical competence involves being able to recognize these facts implicitly and use that information in appropriate ways. When it comes to formalizing such proofs, it turns out to be remarkably difficult to spell out such details precisely. For example, one can take integers to be a different sort of object than complex numbers, in which case one has make use of the standard embedding of the integers in the complex numbers to make sense of the expression; or one can take integers to be complex numbers satisfying the additional property of being integral, in which case, one has to rely on closure properties of operations like addition and multiplication to keep track of which objects in an expression have this property.

A good deal of technology has been borrowed from the theory of programming languages and the theory of automated reasoning to cope with this. For example, *type inference* involves determining, in a given context, what type of object a given expression denotes. *Overloading* is the act of using the same symbol for more than one purpose, such as using $\cdot$ as the multiplication symbol in more than one group, or using $+$ for the natural numbers and the reals. *Polymorphism* and *type classes* provide means of making use of the fact that operations like addition have common properties (like $x + y = y + x$) in different instantiations. A *coercion* is a means of casting of a value of one type to another, for example, viewing an integer i as a real number in contexts where the latter is expected. *Implicit arguments* provide ways of systematically leaving out information when it can be inferred from the context, for example, writing $g \cdot h$ for multiplication in a group when the appropriate instance of group multiplication can be inferred. Coercions and implicit arguments are often insert

automatically using *unification* and *matching* algorithms, which find ways of instantiating variables to get two terms to agree.

The kinds of algebraic reasoning that require such inferences are ubiquitous in mathematics. For example, when manipulating an expression $\sum_{i<n} a_i$, it may not matter whether the summation symbol is taken to mean addition in the integers, the complex numbers, or an abelian group. All the following laws hold in any commutative monoid:

$$\sum_{i<n+1} a_i = \left(\sum_{i<n} a_i\right) + a_n$$
$$\sum_{i \in S \cup T} a_i = \sum_{i \in S} a_i + \sum_{i \in T} a_i \quad \text{if } S \cap T = \emptyset$$
$$\sum_{i \in S} (a_i + b_i) = \sum_{i \in S} a_i + \sum_{i \in S} b_i$$

Also,

$$c \cdot \sum_{i \in S} a_i = \sum_{i \in S} c \cdot a_i$$

holds if $\cdot$ distributes over $+$. In fact, these laws still hold when the summation operator is instantiated not only by summation in the integers or complex numbers, but also as various types of products ($\prod_{i \in S} a_i$), boolean operations or meets and joins in a lattice ($\bigvee_{i \in S} a_i, \bigwedge_{i \in S} a_i$), the minimum and maximum operations on the natural numbers ($\min_{i \in S} a_i, \max_{i \in S} a_i$), unions and intersections of sets ($\bigcup_{i \in S} a_i, \bigcap_{i \in S} a_i$), or the least common multiple or greatest common divisor functions on the integers ($\mathrm{lcm}_{i \in S} a_i, \gcd_{i \in S} a_i$).

Moreover, algebraic reasoning often requires us to view the same object in multiple ways. For example, if F is a field with a subfield E, then F can simultaneously be viewed as a field, a vector space over E, and an algebra over E. If H and K are subsets of a group G that are closed under the group operations, then H and K are also groups in their own right. An expression like $H \cap K$ can therefore be viewed as describing the intersection of the two sets, so that an element g is in $H \cap K$ if and only if g is in both H and K. But $H \cap K$ is also a group, containing the identity of G and having group operations that arise by restricting those of G. Proof assistants need to be able to handle these multiple views, just as we do when we do mathematics. Indeed, that ability is a fundamental part of mathematical competence:

> Understanding mathematical conventions regarding domains and types involves being able to resolve ambiguities and infer type information from the context; being able to recognize concrete domains as implicitly embedded in other domains; being able to recognize concrete and abstract structures as instances of more general classes of structures; and so on.

3.4 Understanding mathematical inference

So far, we have considered only some of the most basic aspects of mathematical competence, namely, the ability to parse and understand general mathematical language, and keep track of the kinds of objects at play in a mathematical proof. We have not even begun to consider even the mildest forms of mathematical reasoning proper.

Spelling out every textbook inference in terms of elementary logical steps is tedious and difficult, and most interactive proof assistants employ various methods to fill in small gaps automatically. One can get a sense of such methods from the two-volume *Handbook of Automated Reasoning* [26], or John Harrison's excellent introductory textbook *Practical Logic and Automated Reasoning* [17]. Once again, here I only have space to offer a cursory glance at the field. Some broad categorizations can be used to characterize different approaches to the problem. To start with, one can distinguish between decision procedures and search procedures. The former are algorithms that are guaranteed (at least, in principle) to terminate when called on a class of inferences, and determine whether or not the inference is valid (ideally, with some kind of formal certificate or proof of validity when the answer is positive). Alas, thanks to Gödel, we know that many classes of inferences are undecidable; to handle such inferences, we can design procedures which search for a proof that an inference is valid, but may not halt if not. One can also distinguish between methods that are domain-general—that is, generic strategies that are designed to work in a wide-range of contexts—and methods that are domain-specific, that is, targeted toward very particular kinds of inferences. Finally, one can distinguish between "principled" search methods, which, for example, guarantee completeness and rely on fundamental theoretical considerations, and "heuristic" methods, that is, algorithms which one has tinkered with and modified to ensure that they work will in practice, often at the expense of having a clean theoretical characterization of their behavior.

When it comes to domain-general methods, one finds systems designed for propositional theorem proving; first-order theorem proving; higher-order theorem proving; and equality reasoning, among others. Each of these is a vast industry, and the references above provide a good entry to the literature. As of late, there has also been interesting research on general ways of combining different procedures in effective ways, such as including some domain specific procedures in general frameworks for proof search. "Nelson-Oppen" methods, which provide ways of combining decision procedures for domains with restricted overlap, represent one important approach.

Research in domain-specific methods is equally active. For example, linear arithmetic packages can determine the solvability of linear equalities and inequalities in the reals, integers, or combinations of these domains. Problems involving nonlinear inequalities are much more difficult, but there has been a lot of work on handling manageable fragments of the theory of real closed fields, or reasoning in the presence of transcendental functions. Interactive proof assistants have also begun to incorporate techniques from computer algebra systems;

for example, methods based on Buchberger's method of *Groebner bases* can be used to solve a number of algebraic problems.

This barely scratches the surface. Automated reasoning is a vibrant field, and despite the great progress that has been made in recent years, there is still a lot we do not understand. When it comes to ordinary mathematical reasoning, one can summarize the state of affairs by saying that automated methods do especially well on large, homogeneous problems, where the search space can be kept under control and the inferences reduced to large but relatively straightforward calculations; but we are still unable to capture straightforward mathematical inferences that chain heterogeneous bits of background knowledge together in various ways. The ability to do so is an important part of our mathematical competence:

> Understanding mathematics involves being able to carry out straightforward mathematical inferences in specific mathematical domains, even when those inferences are difficult to spell out in formal axiomatic terms.

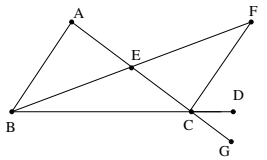
3.5 Understanding mathematical diagrams

Finally, let us briefly consider diagrammatic reasoning, which plays an important role in mathematics. Since the end of the nineteenth century, diagrams have been used only sparingly in professional mathematical texts, and conventional attitudes hold that all rigorous mathematical content should be borne by the text. Nonetheless, diagrams are often used to accompany and illustrate a mathematical argument, and some arguments can be nearly unintelligible until one has drawn a good diagram. Moreover, a good diagram can help guide the writing of a proof, and sometimes a diagram can be entirely convincing in and of itself.

Until recently, it has been common for philosophers of mathematics to dismiss diagrammatic reasoning as being merely a heuristic, psychological artifact of mathematical practice, outside the philosopher's purview. But as of late, a number of philosophers have begun to take visualization and diagrammatic reasoning more seriously [9, 22, 31]. And, it turns out, diagram use if often governed by implicit norms and conventions that can be studied and analyzed.

Consider, for example, Proposition 16 in Euclid's *Elements*.

Proposition I.16. *In any triangle, if one of the sides be produced, then the exterior angle is greater than either of the interior and opposite angles.*



Proof. Let ABC be a triangle, and let one side of it BC be produced to D. I say that the exterior angle ACD is greater than either of the interior and opposite angles CBA, BAC.

Let AC be bisected at E, and let BE be joined and produced in a straight line to F. Let EF be made equal to BE, let FC be joined, and let AC be drawn through to G.

Then, since AE is equal to EC, and BE to EF, the two sides AE, EB are equal the two sides CE, EF respectively; and the angle AEB is equal to the angle FEC, for they are vertical angles. Therefore the base AB is equal to the base FC, the triangle ABE is equal to the triangle CFE, and the remaining angles equal the remaining angles respectively, namely those which the equal sides subtend; therefore the angle BAE is equal to the angle ECF.

But the angle ECD is greater than the angle ECF; therefore the angle ACD is greater than the angle BAE. Similarly also, if BC be bisected, the angle BCG, that is, the angle ACD, can be proved greater than the angle ABC as well. Therefore etc. Q.E.D. □

Later in the *Elements* (Proposition 32 of Book I), Euclid shows that in fact the external angle is equal to the sum of the internal angles, but that depends on facts about parallel lines that are established with the help of Proposition 16. Notice that the last paragraph of the proof simply asserts that angle ECD is greater than angle ECF, presumably because the latter is properly contained in the former. But what justifies this last claim? The diagram "makes it clear," but it is just such "intuitive" uses of the diagram that were called into question during the nineteenth century, with the rise of the axiomatic method.

With some effort, we can show that the desired conclusion is, indeed, warranted by diagrammatic information that is set forth in the proof. For example, points E and F are on the same side of line CD since B is on both lines and, by the construction, E is between B and F. Similarly, we can show that D and F must be on the same side of line CA, since they are both opposite from point B. But these two facts essentially say that F is "inside" the angle formed by CD and CA, which is implies that angle ECF is properly contained in angle ECD.

What is interesting about the *Elements* is that such arguments are never carried out, whereas other inferences are spelled out in great detail. Ken Manders has observed [22] that in a Euclidean proof, topological facts (the inclusion of one angle in another, the intersection of lines, the fact that one point lies between two others along a line, and so on) are often "read off from the diagram," whereas metric facts, such as the congruence of angles are segments, are always justified explicitly in the text. Inspired by his analysis, Ed Dean, John Mumma, and I [4] designed a formal system that exhibits these features, and hence is capable of representing Euclid's arguments more faithfully. Our project involved, in particular, undertaking a careful study of the diagrammatic inferences that occur in the first four books of the *Elements*, and characterizing the norms and conventions that determine the kinds of information that one is able to read off from the diagram in a Euclidean proof. Understanding Euclidean geometry

means, in part, being able to distinguish the valid diagrammatic inferences from invalid ones. More generally:

> Understanding mathematical diagram use involves being able to represent information in a diagram appropriately, and draw valid inferences from the information so represented.

4 The philosophy of mathematics

At this stage, it would be reasonable for you to ask, "what does all this have to do with the philosophy of mathematics?"

To answer this question in a constructive way, it will be helpful to set some ground rules. The question is not meant to spark a turf war, with mathematicians, computer scientists, and philosophers squabbling over who is allowed to make pronouncements over mathematical understanding. Nor is asking whether issues in formal verification have any role in philosophy a matter of passing value judgment on the former; mathematical logic and computer science are important fields of inquiry in their own right, independent of their interaction with philosophy. Rather, let us take the question above to ask what role distinctly philosophical methods can play in relation to the methods of mathematical logic and computer science, and the extent to which philosophical inquiry can inform, and be informed by, work in mathematical logic and software engineering. Towards the end of *The Problems of Philosophy* [28], Bertrand Russell highlighted the role of philosophy in sharpening concepts and clarifying vague intuitions in order to make further scientific inquiry possible. Here I will argue that the philosophy of mathematics can play just such a role here, in helping us come to terms with what exactly we are talking about when we try to talk about mathematical understanding in various scientific contexts. In other words, I am claiming that a better philosophical framework for reasoning about mathematical understanding can support such scientific work, as well as address the kinds of philosophical "problems" that I described in Section 2. The next three sections suggest three ways that such a philosophical framework would be useful.

4.1 Mathematical methods and abilities

Take another look at the pronouncements on understanding that I used to summarize the conclusions of each subsection of Section 3. What do they have in common?

You will notice that the phrase "being able to" occurs in each; in other words, in each case I have characterized an aspect of mathematical understanding in terms of "being able to" perform certain tasks. Informally, we often explain our ascriptions of understanding by describing the associated abilities. For example, if I tell you that my calculus students don't understand integration by parts and you ask me what I mean, I am likely to respond by giving examples of what they can and cannot do.

This provides a helpful way of thinking about mathematics. On traditional foundational accounts, mathematical knowledge is viewed as a collection of propositions. In the context of a formal background theory, we formulate definitions and prove theorems; our knowledge then amounts to knowing that our terms have been defined in thus-and-such a way, and knowing that thus-and-such a theorem is a consequence. But once we have fixed our definitions and axiomatic framework, all the consequences are determined, and the growth of mathematical knowledge is then simply a matter of cranking out these consequences. If we think of mathematical understanding, more broadly, in terms of a body of methods and abilities, new modes of analysis are opened up to us. Rather than a collection of facts, mathematics becomes something much richer and more interesting, namely, a way of thinking and confronting the mathematical challenges we face. It is not just a matter of knowing *that* certain statements are true, but, rather, a matter of knowing *how* to proceed appropriately in mathematical contexts.

Providing a theory of mathematical understanding then amounts to giving an account of the relevant methods and abilities. Such an account can be used to address the philosophical problems raised in Section 2, providing us with better means to explain what we obtain from different proofs of a theorem and why certain historical developments are so important.

There is a straightforward model that can be invoked. Doing mathematics means undertaking various tasks, such as solving problems, proving theorems, verifying inferences, developing theories, forming conjectures, and so on. "Reasoning" involves a passage though various epistemic states en route to our goals. "Understanding" then consists of the methods, techniques, procedures, protocols, tactics, and strategies that make this passage possible. As Section 3 suggests, this involves all of the following:

- being able to recognize the nature of the objects and questions before us
- being able to marshal the relevant background knowledge and information
- being able to traverse the space of possibilities before us in a fruitful way
- being able to identify features of the context that help us cut down complexity

Emphasizing the word "method" means focusing on the procedures that carry us from one state to another; emphasizing the word "ability" means focusing on the net result of the transformation.

But we face a number of problems when we try to fill out the details and develop ways of talking about "methods" and "abilities" in more scientific terms. The notion of a "method" has the connotations of an algorithm, which is to say, a specific way of going about something. But often we only care about what it is that the method accomplishes, and not the particular details of how it is accomplished. That is, different methods can give rise to the same ability; you and I may multiply three-digit numbers in different ways, and, in some contexts, it might only matter that we can both carry out the multiplication. On the other

hand, there is a compositional aspect to our mathematical abilities, in that some abilities can be explained in terms of others. For example, my ability to solve a problem may depend on my ability to apply a certain lemma, which may in turn depend on my ability to expand a definition appropriately. Or my ability to carry out a calculation may depend on the ability to recognize that certain background conditions obtain. These features then push us to think of methods in terms of algorithms and subroutines, which, again, may push us to overly specific descriptions of what they are doing.

There are other features of mathematical abilities and methods that pose challenges. For example, the identity criteria are murky; when should we take two descriptions of a method or an ability to denote the same object? Moreover, methods are inherently fallible. For example: one can show that a subgroup H of G is normal in G by showing that it is a characteristic subgroup of another normal subgroup of G; but this is not the only way to show that H is normal in G, and not every normal subgroup can be identified in this way. Thus we need a way of describing methods that are appropriate, though at the same time imperfect, in a given context.

In sum, the challenge is to develop a language for talking about mathematical methods and abilities that is well-suited to studying the issues raised in Sections 2 and 3. The computational models employed in the field of formal verification provide a good starting point, but, ultimately, we need to focus on the features of mathematics that render it intelligible, rather than proof assistants and their implementation. The goal, then, is to find a level abstraction that is appropriate for talking about mathematical understanding.

4.2 Mathematical concepts

Developing a better language for talking about mathematical methods and abilities may have some side benefits as well. Consider, for example, the notion of a mathematical *concept*. Conventional psychological approaches to the notion of concept, involving prototypes and exemplars, don't do a good job of characterizing mathematical concepts and the role the play in mature mathematical reasoning. For example, some objects may fall more distinctly under the concept of "table" than others, and among various tables, some are more prototypical than others. In contrast, mathematical concepts can have sharp boundaries. There is a precise modern definition of what it means to be a "group," and any particular mathematical object either is or is not an instance of the group concept. To be sure, there are more natural or common instances of groups; but that naturality does not make them any more group-ish than contrived examples.

Yet mathematical concepts have a number of properties that make it hard to pin them down. For example, mathematical concepts, like the group concept, can evolve over time. Moreover, understanding a concept admits degrees: an undergraduate understanding of the group concept is different from that of a graduate student working in group theory, which, in turn, differs from that of the leading experts in the field. Various things "improve our understanding"

of a concept, and not just seeing more of them. For example, representation theory, the method of representing elements of groups as linear transformations of vector spaces, gives us a better understanding of groups. When we consider the historical record, we often recognize "implicit uses of a concept" in the forerunners of our modern theories. For example, Euler's work on power residues (and, particularly, residues modulo a prime) provides a good example of an implicit use of the group concept years before the concept had been defined or axiomatized.

Can we come up with precise ways of thinking and talking about mathematical concepts that accord with these informal observations? One solution may be to think of mathematical concepts as collections of abilities bundled around a central token (see footnote 18 of [3]). Surely one important ability that can be associated with any mathematical concept is the ability to state the corresponding definition and apply it correctly. As a result, we can still follow the traditional Fregean route by saying that an object "falls under a concept" if it satisfies the associated definition. But now we can analyze the notion of "understanding a concept" more generally as possessing the associated abilities. For example, understanding the group concept involves knowing the definition of a group; knowing common examples of groups, and being able to recognize implicit group structures when it is fruitful to do so; knowing how to construct groups from other groups or other structures, in fruitful ways; recognizing that there are different kinds of groups (abelian, nilponent, solvable, finite vs. infinite, continuous vs. discrete) and being able and prone to make these distinctions; knowing various theorems about groups, and when and how to apply them; and so on.

You can check that this way of thinking about mathematical concepts jibes well with the observations in the previous paragraph. For example, concepts evolve as the ways and contexts we use them expand, and using a concept "implicitly" can mean employing an instance of a method associated to the concept without identifying it as such. To be sure, this makes mathematical concepts somewhat vague and open-ended. But the point is, our talk of mathematical concepts *does* treat them as vague and open-ended; and this analysis makes them vague and open-ended in just the right away.

It may be helpful to compare this to the more traditional view of mathematical concepts, which treats them as static and unchanging entities. On the Fregean view, the analysis of a concept amounts to fixing the proper definition, which, in a sense, determines everything there is to say about the concept. The stark difference between the two views is tempered by the fact that Frege was more specifically focused on the problem of justification of mathematical knowledge. When it comes to accounting for the normative rules for mathematical justification, Frege's account fares rather well. There are at least three senses, however, in which a Fregean analysis comes up short.

The first is foundational. In the early twentieth century, mathematical logic was able to reduce much of mathematical reasoning to a few basic concepts and axioms; Zermelo-Fraenkel set theory, for example, provides a foundation for mathematics based on a small list of assumptions about the universe of sets. But

one is still left with the question as to what justifies *those*. Most philosophers of mathematics take logicism to have failed, in the sense that doing mathematics requires commitments to entities and principles that cannot be accounted for by logic alone. But attempts to find compelling extralogical principles that can provide us with perfect knowledge of abstract mathematical objects have largely stalled. In other words, we have swept all our dust into one neat little pile and invested a good deal of effort in moving the pile of dust around, but it never really goes away. It seems likely that if one is aiming to justify one's axioms and basic concepts on broader grounds, one will ultimately have to attend to the roles they play in organizing our mathematical knowledge, and the role that knowledge plays in organizing and structuring our scientific experiences (Maddy [20] urges something like this approach). A more robust notion of concept can help in that regard, by giving us a richer vocabulary to explain why certain ways of organizing and structuring our knowledge are better than others.

A second sense in which the Fregean analysis falls short is that it fails to account for the kind of mathematical reasoning that often takes place in the absence of a clear foundational framework. For example, in the eighteenth century Euler employed a number of novel and striking arguments involving infinite sequences and series which were not made rigorous, according to the modern understanding of that term, until much later; sometimes not even until the twentieth century (see [33]). Mark Wilson's engaging and thorough exploration of concepts [36] offers a number of similar examples in applied mathematics and physics, as does Urquhart [32]. The work on Euclidean geometry described in Section 3.5 shows that the geometry in Euclid's *Elements* is also governed by precise norms, once again in the absence of a Fregean foundation. Once again, a more robust notion of concept may be able to help explain the way informal concepts guide our reasoning, even in the absence of precise definitions.

Finally, Frege's analysis was simply not designed to account for the kinds of normative evaluations discussed in Section 2. As Poincaré observed, telling us how we are *allowed* to use our mathematical concepts is a far cry from telling us how we *ought* to use them. Thus we can view the Fregean analysis as more specifically trying to provide an idealized account of the normative rules of justification in situations where our concepts can be clearly defined. When it comes to justification in the broader sense of using mathematical notions fruitfully and appropriately, once again, we should be prepared to look beyond the Fregean framework.

4.3 Mathematical ease and difficulty

There are other ways that the issues raised in Sections 2 and 3 push us to look beyond traditional foundational analysis. The problem is simply that foundational reduction washes out many important nuances. For example, from a set-theoretic standpoint, there is only one type of mathematical object (set); there is only one fundamental relationship between mathematical objects (the binary element-of relation); and one only needs one "method" to verify inferences, that is, systematic search for a proof from the axioms of set theory.

From this standpoint, it is hard to recognize differences between algebraic and geometric methods; different styles of proof; or the value of a good definition.

What makes foundational reduction an oversimplification in such contexts are issues of complexity. Knowing that, in principle, definitions and theorems can be unpacked until we get to set-theoretic primitives does not help us reason about them pragmatically. Differences in the way we organize our mathematical knowledge and express our mathematical ideas matter precisely because we have limited time, energy, memory, and reasoning capacities. Part of understanding why we value certain mathematical developments involves understanding how the right concepts and methods simplify the mathematical tasks before us.

But how shall we measure complexity? As philosophers, we won't be the first to grapple with the issue; "complexity" is a term of art in a number of applied disciplines. In computer science, one measures the complexity of problems in terms of asymptotic bounds on the time and space needed to compute solutions in a fixed machine model. In logic, one can measure the complexity of definitions, say, by the number (and type) of quantifiers they employ; and one can measure the complexity of proofs by the number of symbols they contain. Psychologists measure the complexity of basic cognitive tasks in terms of the amount of time it takes to carry them out, or the stage of our development at which we are capable of doing so.

For example, the field of proof complexity [25] provides a number of interesting "speedup results," which show how expanding the language and conceptual resources of a mathematical theory, even conservatively, can serve to shorten the lengths of proofs dramatically. With some cleverness, one can find explicit combinatorial theorems that exemplify this behavior (see [7, 23]). This may seem to offer good explanations as to how a careful choice of language and concepts serves to reduce complexity. But such results do not tell the whole story. First, "length of proof" is a count of the number of symbols in proofs in a formal axiomatic system. Such a measure depends very much on how the system is formulated; although such measures tend to be stable, up to a polynomial, across reasonable systems, that says nothing about the length of the proof of a *single* theorem. Second, ordinary textbook proofs are much higher-level objects than formal axiomatic derivations, and, as I argue elsewhere [2], the kinds of normative assessments that we are interested in here do not accord well with the low-level modeling. Third, the combinatorial examples are somewhat contrived, cooked up by logicians to serve as counterexamples, and the speedup vanishes when one replaces the systems in question with ones only slightly stronger. Indeed, ordinary mathematical theorems that skirt unprovability using ordinary mathematical methods are surprisingly hard to come by; most mathematics can be carried out in fairly weak theories, where the obvious reflection principles that can shorten a proof are uncontroversial (see [1]). Moreover, length of proof is a measure of the formal object, not the ease or difficulty we encounter in trying to find, remember, or reconstruct it; it is a measure of "syntactic complexity" rather than "difficulty." Finally, speedup results are overly dramatic. A definition that makes it possible to carry out our reasoning more cleanly and efficiently, and thereby reduce the length of a journal article

by a third, is clearly a good definition. What we really care about are the subtle ways that good definitions streamline our ordinary mathematical experiences, not the dramatic and clever ways we can abuse a formal axiomatic system.

One can raise similar objections to other complexity measures on offer. We might try to classify the difficulty of common mathematical tasks in terms of their computational complexity, but this is an asymptotic model; what we often care about are the complexity of individual tasks, or, for example, a class of tasks where the parameter in question can reasonably be taken to be bounded by a constant (say a trillion). This objection isn't just pedantic; at small sizes, the particular choice of machine model can make a huge difference. Turing machines are not good models for the kinds of things that *we* find it easy to do, nor are they good models for the kinds of tasks that take place against richly structured background knowledge. Finally, computational complexity is best at modeling deterministic algorithms; in mathematics, we often employ heuristic methods that tend to do well in the kinds of situations that arise in ordinary mathematical practice. What is missing is an informative theoretical characterization of what it means to "do well in the kinds of situations that arise in ordinary mathematical practice." Computational complexity was simply not designed for this purpose.

Psychological measures of difficulty go too far to the other extreme. For one thing, we only have clean experimental results for very basic cognitive tasks. Moreover, this shifts our focus to "incidental" features of our cognitive abilities, rather than the "essential" features of the mathematics we are trying to model. What we want is an account of how mathematics helps us take advantage of the distinctly mathematical features of a problem at hand and tame a combinatorial explosion of possibilities, one that is not overly sensitive to the number of digits we are capable of holding in our short-term memory.

But it is important to keep in mind that saying that the measures of complexity we have considered are not quite right does not mean that they are entirely wrong. Certainly the lengths of proofs and calculations and our cognitive limitations have a lot to do with what makes a piece of mathematics hard or easy. Conventional complexity measures therefore provide a good starting point. What we need now are ways of talking about complexity that are suitable for analyzing the features of the *mathematics* that extend the capacity and reach of our thought.

Once again, getting a better grip on mathematical ease and difficulty may have broader philosophical implications. I began this essay by describing recent efforts to come to terms with the various values that one comes across in ordinary mathematical discourse. Such explorations have given rise to methodological concerns. It is all well and good to make lists of theoretical virtues; but what, exactly, endows them with normative force? Mathematics is ultimately a matter of getting at the truth; isn't everything else incidental? Aren't all the other judgments merely subjective and pragmatic, falling outside the proper scope of philosophy? Tappenden [31] raises such concerns as follows:

> [J]udgements of "naturalness" and the like are *reasoned*. It is not just

> some brute aesthetic response or sudden, irrational "aha!" reaction that brings about the judgement that — for example — "the scheme is the more natural setting for many geometric arguments"... Quite the contrary: elaborate reasons can be and are given for and against these choices. One job facing the methodologist of mathematics is to better understand the variety of reasons that can be given, and how such reasons inform mathematical practice.
>
> The factual observation should be beyond controversy: reasoned judgements about the "proper" way to represent and prove a theorem inform mathematical practice. I have found that more contention is generated by the disciplinary classification of the study of these judgements and the principles informing them: is this *philosophy,* or something else, like cognitive psychology?

At issue is not whether we can clarify and explain our mathematical assessments; instead, the question is whether we can find any "objective" sense in which they should be taken to be normative, rather than reflections of personal preference, historical accident, or incidental features of our biological makeup. In other words, what is lacking is a sense in which our mathematical values are mathematically valuable.

Such concerns are not limited to mathematics; they apply just as well to questions concerning the objectivity of ethical and aesthetic judgments. But when it comes to mathematics, a suitable theory of ease and difficulty may provide an informative way of addressing these concerns. Insofar as we can develop appropriate idealizations of our cognitive capacities and limitations, there is a sense in which we can determine some of our value judgments to be objective; that is, we can show how our various machinations and stratagems serve to extend the capacities for the use and discovery of mathematical knowledge in any beings with cognitive constraints roughly like ours. This may not put all our concerns about normativity and objectivity to rest, but it provides a sense in which clear philosophical progress can be made.

5 Conclusions

Encouraged by these musings, you may find yourself tempted to get right to work and start defining basic terms like "understanding," "ability," and "concept." Resist that temptation! Before we begin to construct an overarching theory, we have to start coming to terms with some of the basic data. It therefore makes sense to begin with more focused questions, ones for which satisfying answers are within reach. To that end, it is also helpful to look to domains of application, among those fields that explicitly or implicitly depend on notions related to mathematical understanding: fields such as formal verification and automated reasoning; mathematical pedagogy and cognitive science; history (and historiography) of mathematics; and mathematics itself. In other words, we should begin by trying to clarify specific *aspects* of mathematical understanding, and

the roles that our conceptions of mathematical understanding play in particular scientific practices. Over time, the data we accumulate from such smaller and more focused studies should come together to provide us with a coherent, comprehensive picture.

But what if they don't? It is conceivable that our disparate attempts to get at mathematical understanding will take us down divergent paths. We may decide, in the end, that notions of understanding the arise in automated reasoning have nothing to do with notions of understanding that arise in cognitive science, which, in turn, tell us nothing about the methods and goals of working mathematicians. What then?

Well, in that case, our work will have merely contributed to the conceptual foundations of automated reasoning, cognitive science, pedagogy, history of science, and so on; and taught us some interesting things about mathematics as well. Surely we could do a lot worse.

References

[1] Jeremy Avigad. Number theory and elementary arithmetic. *Philosophia Mathematica*, 11:257–284, 2003.

[2] Jeremy Avigad. Mathematical method and proof. *Synthese*, 153:105–159, 2006.

[3] Jeremy Avigad. Understanding proofs. In [21], pages 317–353.

[4] Jeremy Avigad, Edward Dean, and John Mumma. A formal system for Euclid's *Elements*. *Review of Symbolic Logic*, 2:700–768, 2009.

[5] Jeremy Avigad, Kevin Donnelly, David Gray, and Paul Raff. A formally verified proof of the prime number theorem. *ACM Transactions on Computational Logic*, 9:2, 2007.

[6] Solomon Feferman. Kreisel's "unwinding" program. In Piergiorgio Odifreddi, editor, *Kreiseliana: About and Around Georg Kreisel*, pages 247–273. A.K. Peters Ltd., Wellesley, MA, 1996.

[7] Harvey Friedman. *Boolean Relation Theory and Incompleteness*. ASL Lecture Notes in Logic, to appear.

[8] H. Furstenberg. Ergodic behavior of diagonal measures and a theorem of Szemerédi on arithmetic progressions. *Journal d'Analyse Mathématique*, 31:204–256, 1977.

[9] Marcus Giaquinto. *Visual Thinking in Mathematics: An Epistemological Study*. Oxford University Press, Oxford, 2007.

[10] Georges Gonthier. Formal proof—the four-color theorem. *Notices of the American Mathematical Society*, 55:1382–1393, 2008.

[11] Georges Gonthier and Assia Mahboubi. A small scale reflection extension for the coq system. Technical Report INRIA-00258384, Microsoft Research and INRIA, 2008.

[12] Georges Gonthier, Assia Mahboubi, Laurence Rideau, Enrico Tassi, and Laurent Théry. A modular formalisation of finite group theory. In Klaus Schneider and Jens Brandt, editors, *Theorem Proving in Higher Order Logics 2007*, pages 86–101. Springer, Berlin, 2007.

[13] Thomas C. Hales. The Jordan curve theorem, formally and informally. *American Mathematical Monthly*, 114:882–894, 2007.

[14] Thomas C. Hales. Formal proof. *Notices of the American Mathematical Society*, 55:1370–1380, 2008.

[15] John Harrison. A formalized proof of Dirichlet's theorem on primes in arithmetic progression. *Journal of Formalized Reasoning*, 2:63–83, 2009.

[16] John Harrison. Formalizing an analytic proof of the prime number theorem. *Journal of Automated Reasoning*, 43:243–261, 2009.

[17] John Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, Cambridge, 2009.

[18] Steven Kieffer, Jeremy Avigad, and Harvey Friedman. A language for mathematical language management. *Studies in Logic, Grammar and Rhetoric*, 18:51–66, 2009.

[19] Ulrich Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Springer, Berlin, 2008.

[20] Penelope Maddy. *Second Philosophy: A Naturalistic Method*. Clarendon Press, Oxford, 2007.

[21] Paolo Mancosu, editor. *The Philosophy of Mathematical Practice*. Oxford University Press, Oxford, 2008.

[22] Kenneth Manders. The Euclidean diagram. In [21], pages 80–133.

[23] Jeff Paris and Leo Harrington. A mathematical incompleteness in Peano arithmetic. In Jon Barwise, editor, *Handbook of Mathematical Logic*, pages 1133–1142. North-Holland, Amsterdam, 1977.

[24] Henri Poincaré. *Science et Méthode*. Flammarion, Paris, 1908. Translated by Francis Maitland as *Science and Method*, Dover Publications, New York, 1952.

[25] Pavel Pudlák. The lengths of proofs. In Samuel Buss, editor, *Handbook of Proof Theory*, pages 547–637. North-Holland, Amsterdam, 1998.

[26] John Alan Robinson and Andrei Voronkov, editors. *Handbook of Automated Reasoning (in 2 volumes)*. Elsevier and MIT Press, 2001.

[27] Piotr Rudnicki. An overview of the Mizar project. In *1992 Workshop on Types for Proofs and Programs*. Chalmers University of Technology, Bastad, 1992.

[28] Bertrand Russell. *The Problems of Philosophy*. Home University Library, 1912. Reprinted by Oxford University Press, Oxford, 1959.

[29] Endre Szemerédi. On sets of integers containing no k elements in arithmetic progression. *Acta Arithmetica*, 27:199–245, 1975.

[30] Terence Tao. A quantitative ergodic theory proof of Szemerédi's theorem. *Electronic Journal of Combinatorics*, 13(1):Research Paper 99, 2006.

[31] Jamie Tappenden. Proof style and understanding in mathematics I: visualization, unification, and axiom choice. In Paolo Mancosu, Klaus Frovin Jorgensen, and Stig Andur Pedersen, editors, *Visualization, Explanation and Reasoning Styles in Mathematics*, pages 147–214. Springer, Berlin, 2005.

[32] Alasdair Urquhart. Mathematics and physics: strategies of assimilation. In Mancosu [21], pages 417–440.

[33] V. S. Varadarajan. *Euler Through Time: A New Look at Old Themes*. American Mathematical Society, Providence, RI, 2006.

[34] Makarius Wenzel. Isabelle/Isar — a generic framework for human-readable proof documents. *Studies in Logic, Grammar, and Rhetoric*, 10(23):277-298, 2007.

[35] Freek Wiedijk. *The Seventeen Provers of the World*. Springer, Berlin, 2006.

[36] Mark Wilson. *Wandering Significance: An Essay on Conceptual Behavior*. Oxford University Press, Oxford, 2006.

[37] Ludwig Wittgenstein. *Remarks on the Foundations of Mathematics*. Blackwell, Oxford, 1956. Edited by G. H. von Wright, R. Rhees and G. E. M. Anscombe, Translated from the German by Anscombe. Revised edition, MIT Press, Cambridge, Mass., 1978.