The changing practices of proof in mathematics

Andrew Arana
University of Paris 1 Panthéon-Sorbonne, Department of Philosophy, and Institute for History and Philosophy of Science and Technology (IHPST)

Metascience 26:1 (2017), pp. 131--135
The final publication is available at https://link.springer.com/article/10.1007%2Fs11016-016-0150-1

Review of Dowek, Gilles, *Computation, Proof, Machine*, Cambridge University Press, Cambridge, 2015. Translation of *Les Métamorphoses du calcul*, Le Pommier, Paris, 2007. Translation from the French by Pierre Guillot and Marion Roman.

When deciding how many tiles we need in order to cover our kitchen floor, we calculate: compute the total area of the kitchen, the area of the tiles, and divide as needed. One could use this particular, practical problem to raise a more abstract problem, concerning the partition of arbitrary finite polygon configurations in the plane. The interest of this abstract problem might be to generalize the given particular problem so as to optimize our problem-solving, and resolve a family of such particular problems with one solution. Or it may simply seem to be interesting in its own right. This second, abstract problem will not be readily resolved by calculations of the type familiar from home renovation, though. Instead, the problem calls for *reasoning*: for instance, an analysis, to break it into smaller, more manageable

problems.

Gilles Dowek's fascinating book begins with this cleavage between computation and reasoning, illustrated with the transition from the algorithmic solutions to accounting problems of ancient Mesopotamia's, to the analyses of abstract geometrical problems of ancient Greece. These abstract problems, such as finding an isosceles right triangle with three sides each a integral multiple of a given unit measure, led the Greeks to develop the axiomatic mathematics we now associate with Euclid, for no algorithmic approach seemed capable of handling the infinitely many cases such problems presented. Instead, one could reason generally from generally-accepted first principles inferentially strong enough to deduce answers to such problems. With the canonization of Euclid's *Elements* as the exemplar of mathematical rigor, computation was demoted in mathematical discourse, if not practice. Perhaps a geometer used computations to discover her results, but they should not be included in the "official" presentation. A celebrated example in the modern era is the calculus of Newton and Leibniz, evidently employed by Newton in his workings on mechanics but left out of the *Principia* in favor of Euclidian reasoning. Thus mathematical proof came to be synonymous with axiomatic reasoning, relegating algorithmic computation to the background.

Axiomatic reasoning underwent a reinvention in the nineteenth century, as is well-known. The development of predicate logic by Frege, Boole and Schroeder

enabled axiomatizations of arithmetic with light shed on their logical frameworks. These axiomatizations, by Frege, Peano, and Russell and Whitehead mixed logic and set theory, thus leading to further axiomatizations just of set theory itself. Set-theoretic axioms could then be seen as a “foundation” of mathematics, providing a base of suppositions from which one could logically derive all truths of arithmetic, and given the arithmetization of geometry in the Cartesian style, of geometry as well. With these developments the understanding of axiomatics changed as well: following reflections of Poincaré and Hilbert, one could think of axioms as implicit definitions of the non-logical concepts occurring in the axioms.

Dowek dwells on these familiar matters in order to make clear the central tenets of the conception of mathematical proof as axiomatic reasoning, a conception he has been showing as dominant going into the twentieth century. His aim going forward in the book is to present a new conception of mathematical proof, the *algorithmic* conception, that today vies for attention as well. As an actor in this latter development, Dowek understandably spends most of the rest of the book on the development of this new conception.

A first step in this new development is the new focus on algorithms emerging in the early twentieth century. Hilbert led the way here, asking for an algorithm to determine the provability of any proposition in predicate logic. This “decision problem” led to the clarification of the notion of computability as the step by step, regulated

transformation of expressions. Viewing inferences as computations, the decision problem asked if any algorithm, any series of computations, could determine whether a given expression was provable in predicate logic. Here it was supposed that an algorithm must always halt after a finite number of steps. As Church and Turing showed, if there were such an algorithm, then one could construct another algorithm that halts if and only if it does not. This result, known as “Church’s theorem”, computes with expressions formalizing the rules of predicate logic itself, a method pioneered by Gödel. Its negative answer to Hilbert’s decision problem clarifies the difference between algorithmic calculation and axiomatic reasoning: the latter is able to carry out metamathematics on the former, showing that not every mathematical problem can be solved by computation.

A key step in the work of Church and Turing was a precisification of the notion of algorithm, in order to make definitive their claim that *no* algorithm could resolve the provability of propositions of predicate logic. This is of course what we now call Church’s thesis. Dowek distinguishes two forms of Church’s thesis, psychological and physical, wherein all the algorithms executable by a human being / machine, respectively, in order to resolve a particular problem are expressible by a set of rules of computation. The physical form is strictly stronger, Dowek observes, since the computational capacities of nature may exceed those of all human minds. After discussing a proof of the physical Church thesis was offered by Robin Gandy, Dowek shows how the physical Church thesis can be used to

argue that nature is mathematizable, by considering natural systems as satisfying functional relationships. Consider for example a ball dropped from a tower whose height is measured by a height gauge where the time of descent of the ball to the ground is measured by a clock. For a duration of *n* seconds, this system yields a distance fallen by the ball. Dowek suggests viewing this functional relationship between time and distance as an algorithm of nature, which by the physical Church thesis is expressible by a set of computation rules, and thus mathematizable in an algorithmic form. This algorithmization of natural science has already begun in linguistics, and is rapidly advancing in physics and biology under the guises of quantum computing and bioinformatics, in which one studies the computing processes carried out in particular quantum systems and cells, respectively.

Church's views on computation were supported by his work on the lambda calculus, a model of computation as strong as Turing machines. The lambda calculus, today the foundation of functional language programming, was envisioned originally as a notation for functions. In time Church came to realize that the lambda calculus could engender an foundational alternative to set theory, in taking functions rather than sets as the basic object of mathematics. Dowek notes that this would have given computation a fundamental role in the foundations of mathematics, since the functions in lambda calculus are all expressibly by algorithms. It was not to be, as Kleene and Rosser proved this foundational version of lambda calculus to be inconsistent. As with Russell's paradox,

the problem was self-reference, here of a function to itself. In response Church gave a new formulation of this foundational theory barring functional self-reference, nowadays called Church's type theory in keeping with the Russellian insight that stratifying objects into different types and barring some cross-type applications can forestall paradox.

While computation failed to take its part at the heart of foundations of mathematics in Church's work, simultaneously work on constructive mathematics was taking hold. In brief, the value, and sometimes validity, of proofs that did not provide explicit constructions of their objects, were called into doubt. One thinks of the classical intermediate value theorem, which asserts that a continuous function over the reals will take on a particular value at some point, without necessarily constructing that point. Following Brouwer and Markov, "intuitionist" schools developed in which such proofs were avoided in favor of proofs that give the relevant explicit constructions, avoiding in particular use of the law of excluded middle. These constructive proofs can be seen as defining algorithms; for instance, a constructive proof of a proposition P of conditional form A->B can be interpreted as an algorithm that transforms a proof of the antecedent A into a proof of the consequent B. In this way logicians developed an algorithmic interpretation of proofs that promised to give a central place to computation in the foundations of mathematics.

This algorithmic interpretation of proofs is the focus of

the rest of Dowek's book. Martin-Löf's development of intuitionistic type theory was a key stage in this development. Starting from Church's type theory, Martin-Löf added the capacity to assert two propositions to be "equal by definition", though as Dowek notes, this might better be called "equality by computation". For example, in intuitionistic type theory the propositions "2+2=4" and "4=4" are equal by definition, because using the definitions of integers and integer addition these two propositions can easily be transformed into one another. Proofs in intuitionistic type theory differ from proofs in other frameworks in that they add computational rules to the usual components of proofs, axioms and inference rules. We can thus think of intuitionistic type theory as permitting proofs that leave calculations "for the reader". This leads then to shorter proofs, since the computational steps may be left out, but it does not make proofs any shorter to check as valid, since those steps must be carried out at that step. Chapters 10 and 11 discuss how this addition of computational rules to proof invigorated the search for automated theorem provers and, more modestly, automated theorem checkers.

Proofs employing computations have become a controversial part of ordinary practice since the 1970s, when Appel and Haken's proof of the four-color theorem employed computer methods to check thousands of potential counterexamples. Hales' proof of the Kepler conjecture in 1995 was another such instance, with the *Annals of Mathematics*, a top mathematics journal, footnoting Hales' article with the admission that its

editors could not fully vouch for the correctness of its algorithm having handled all cases. Dowek notes that while such proofs may not be explanatory, in the sense of answering *why* their theorems are true, they can still be checked by automated methods, and thus can be incorporated into rigorous mathematical practice. Indeed these proof checking methods thus become another instrument of mathematical practice, like the ruler and compass for the classical geometer.

Dowek's book is a superb overview of the transformation of mathematics toward becoming a computational science. It is historically rich, philosophically inquisitive, and mathematically rigorous. One point of contention might be the stress Dowek places on the role of computation in all branches of mathematics. The cleavage between computation and reasoning that frames the book's dialectic was rooted, in ancient times, in a classical conception of mathematical knowledge, held for instance by Aristotle. With the notable exception of Archimedes, ancient Greek mathematicians attempted to avoid calculation in geometry because they thought the knowledge gained by purely geometrical proofs of geometrical theorems was *better* than the knowledge gained by proofs of geometrical theorems involving arithmetic. More moderately, purely geometrical proofs of geometrical theorems provide *different* information than proofs mixing geometry and arithmetic, valuable in its own right, and provide knowledge that is more "stable" than knowledge provided by mixed proofs. And indeed this is so not just for geometry, but for any single branch of mathematics.

That's not to say that mixing, say, geometry and calculation is bad, obviously; it's just to say that in emphasizing the role of calculation in proof generally, it is worth remember that something valuable is lost in geometry, for instance, when calculations play a significant role. (For details, see M. Detlefsen and A. Arana, "Purity of Methods", *Philosophers' Imprint*, 11:2 (2011), and A. Arana, "On the alleged simplicity of impure proof", in R. Kossak and P. Ording, editors, *Simplicity: Ideals of Practice in Mathematics and the Arts*, Springer, 2016.)

One can see Dowek's work as clarifying this logical analysis of proof, to distinguish between the purely logical and computational parts of proofs. Is this distinction artificial or intrinsic to different modes of mathematical thought? Its persistence through the development of mathematics suggests the latter, but the new tradition being developed by Dowek and others demands consideration.