Book review

Computation, Proof, Machine: Mathematics Enters a New Age by Gilles Dowek, Cambridge University Press, 2015. Hardback, ISBN 978-0-521-11801-9, 152 pages.

The book under review is a translation of the book that appeared first in French, in 2007. The fact that it was translated only 8 years later does not take anything away from its relevance. Still, as mentioned at the end, some small additions could have been made to make it more up to date.

The subject of the book is a historical account of the developments in logic, and (using Dijkstra's terminology) computing science, interpreted as the science of computing, culminating with a presentation and justification of the computer-assisted proofs. In the process some small incursions in the history of mathematics are made and philosophical issues related to computing are discussed.

This is an interesting story, spanning more than 2500 years, and the author tells it well. The book is meant for a 'general reader', which means that no mathematical notation and concepts beyond, say, primary school are assumed. Adopting such restrictions makes the task of writing such book quite a challenge, but the reward is that it can be read by a wide audience.

Chapter 1 of the book starts with a short reference to the Babylonian times with an example of the problem of dividing 1,152,000 measures of grain by 7, that was found on an ancient tablet dating from about 2500 BC. Such specific problems were generalised by the Greeks to questions requiring *reasoning* instead of just *computing*. As an example the problem of establishing the irrationality of $\sqrt{2}$ is given.

Then the Aristotle logic of syllogisms and the improvement upon it by the Stoics are discussed and Euclid's *Elements* are mentioned. The author notes that Euclid's approach was based on an axiomatic method, yet in spite of it Greeks never applied logic to mathematics (in their case geometry and number theory). The explanation is of course, that their logic was just too simplistic. The other reason, mentioned only in the next chapter, is that it was only thanks to Descartes (and Fermat) that geometric problems could be *expressed* in a logical form, as formulas of the theory of real numbers. The side effect of this separation between logic and mathematics was that computing fell in-between and was not further investigated, in particular not by means of reasoning.

Chapter 2 provides a minimalistic crash course in the history of computing by discussing in turn a couple of seminal points: the contributions of Thales, Euclid's algorithm, positional notation, and calculus. One specific contribution that could have been mentioned here was Heron's method of computing a good approximation of the square root, as it can be viewed as the starting point of numerical analysis.

The conclusion is that computing led to the rise of the concept of an algorithm and that the algorithmic methods deeply affected geometry.

Chapter 3 focuses on the history of predicate (i.e., first-order) logic, by resuming the discussion where it was left in Chapter 1. The departure point is Kant's distinction between on the one hand *analytic* and *synthetic* judgments and on the other hand *a priori* and *a posteriori* judgments. Recall that a judgment is analytic if it is true by definition and synthetic if it is true but not only by definition. In turn, it is *a priori* if it takes place in one's mind without any interaction with the outside world, while it is *a posteriori* if such an interaction is necessary.

Kant claimed that all mathematical judgments are synthetic and a priori. Frege contested this view, at least as far as arithmetic is concerned. To this end he embarked on an ambitious programme that aimed at showing that arithmetic and parts of real-variable analysis can formulated in logic. In this endeavour he vastly generalised propositional logic by adding function and relation symbols and quantifiers.

Discussion of Frege's work somehow lacks references to some stages in its development. What we just mentioned is the subject of his 1879 book *Begriffschrift* (*Concept Script*). It was followed in 1884 by *Die Grundlagen der Arithmetik* in which he tried to prove consistency of mathematics. The paradoxes found by Burali-Forti (1897) and then Russell (1902) concern the later work, not mentioned in the book. The next relevant contribution was Russell's type theory, followed by Hilbert's simplification of it to the predicate logic.

A somewhat striking omission in this account is George Boole. In fact, his name does not even appear in the book (though Boolean algebra is mentioned once without explanation). It is true that Boole's contributions consisted in linking logic with algebra and were not used by Frege (in contrast to Peano who is briefly mentioned in the chapter). However, in retrospect his work can be seen as a systematisation of Aristotle's and Stoics' approach to logic. Further, his and Augustus De Morgan's (also not mentioned in the book) contributions are relevant to the subject of the book because of the success of the SAT solvers in verifying software, hardware components and even establishing new mathematical results, like most recently, in M.J.H. Heule, O. Kullmann and V.W. Marek, *Solving and Verifying the Boolean Pythagorean Triples Problem via Cube-and-Conquer*, in Proc. Theory and Applications of Satisfiability Testing (SAT), pp. 228–245, 2016.

In Chapter 4 Hilbert's decision problem is discussed, that is, the question whether the first-order logic is decidable. Subsequently the elimination of quantifiers is explained, followed by a reference to Church's theorem that provides a negative answer to Hilbert's problem. Then the need for a definition of computable functions is put forward, with brief references to the works of Herbrand, Gödel, Church, Turing and Kleene. The halting problem is also discussed and the diagonalisation method to prove its undecidability is explained (in plain English, which made it for me difficult to follow). I was initially confused by this statement on page 46: "The addition algorithm is used to decide the provability of the propositions of the form "$x + y = z$". " and only after an exchange with the author I understood that he meant all ground instances of such formulas. Finally, the discussion 'Analytic does

not mean obvious' is rather confusing for the simple reason that it is not clear how to define the 'obvious'.

Chapter 5 provides a useful account of Church's thesis by distinguishing psychological and physical versions of it and comparing them with the computations defined by means of computation rules. Subsequently a critical account of Gandy's proof of the physical version of the thesis is given. The rest of the chapter consists of an interesting discussion of the often debated (in the words of Eugene Wigner) "unreasonable effectiveness of mathematics in the natural sciences" and the relevance to it of the physical version of Church's thesis. Also, some alternative form of computing (quantum computing and bioinformatics) are very briefly mentioned. One could have have been slightly more informative here and refer to DNA computing and neurocomputing.

The short Chapter 6 is devoted to an informal exposition of the lambda calculus and Church's type theory. Also, the notion of a self-application is discussed. It would have been useful to mention here typed lambda calculus, especially since Russell's type theory is discussed in Chapter 3. Further, I miss a corresponding short chapter about the Turing machines.

Chapter 7 provides a good exposition of constructivism by clarifying first the notion of excluded middle and explaining how non-constructive conclusions can arise in logic (due to reasoning by cases). Subsequently Brouwer's intuitionism is discussed under the name of constructivism (that is indirectly identified with the intuitionism). This terminology is unfortunate, since it is generally assumed that intuitionism is *an example* of constructivism. The author mentions a dispute on the subject of intuitionistic proofs between Brouwer and Hilbert. To readers interested in details I recommend the excellent H. Hellman, *Great Feuds in Mathematics*, Wiley, 2006, that has a whole chapter devoted to it.

The chapter ends by mentioning Gödel's (double) negative translation of the formulas of the predicate logic (the crucial step of which is the transformation of the atomic formula ϕ to $\neg\neg\phi$) that transforms formulas provable classically into formulas provable in the intuitionistic logic.

Chapter 8 is the first place in the book in which computation and proofs meet. It starts by mentioning cut elimination, unfortunately, without any explanation (apart of mentioning that it is an operation on proofs that makes them constructive). The exposition continues by explaining why constructive definitions of functions turn them into algorithms and why, more generally, constructive proofs can be seen as algorithms.

Chapter 9 returns to the connections between proofs and computations. It discusses Martin-Löf's intuitionistic type theory, that is a constructive counterpart of Church's type theory. The theory is higher-order (and hence more expressive) and the proofs in it continue to be constructive. As a result they can be rendered as algorithms. Quite some space is devoted to the issue of equality, by clarifying that one should distinguish between equality by definition and equality by computation and that they coincide in the case of the intuitionistic type theory. Finally, it is noted that in general there is a trade-off between the length of a proof and its checking.

Chapter 10 deals with the automated theorem proving. On page 97 the early, widely exaggerated, claims of the researchers working in this field are mentioned, in particular that within ten years computers, equipped with the ability to construct proofs, will beat humans in chess. It would have been useful to mention in this context that forty years later, in 1997, the program Deep Blue did win in chess with Kasparov (though admittedly by means of a 'brute force approach'). The chapter offers a short discussion of unification, the resolution method and paramodulation. Subsequently a rationale for the Knuth-Bendix completion procedure is given. The chapter ends by a brief exposition of Plotkin's extension of Robinson's unification algorithm and Huet's higher order unification algorithm.

In a short Chapter 11 a convincing case for the proof-checking programs is made: one of such programs discovered an, admittedly small, mistake in one of Newton's proofs concerning the motion of planets. Subsequently the author briefly discusses the first proof-checking program, De Bruijn's Automath (that was applied to verify mathematical proofs). Since the length of a correctness proof of a program is at least proportional to the length of the program, a natural need for automated proof-checking arises. The author then mentions the LCF and ACL2 (so not ACL) programs as the first tools that were developed to automatically verify correctness of programs. Unfortunately, the subsequent comment about the works of Boyer and Moore (who designed the ACL2 program) that "[they] realized as much of Hilbert's program of replacing reasoning with computing as Church's theorem allowed them to." is too vague to appreciate it.

Chapter 12 provides an interesting account of some of the achievements of automated theorem proving: a verification of the proof of the Four Colours theorem, of Morley's theorem (stating that the trisectors of the angles of a triangle form an equilateral triangle) and of the proof of Kepler's conjecture (about the optimal packing of spheres in a box). Also, a reference is made to the Computer Algebra Systems (CAS) with an interesting remark that one such system could in a couple of minutes carry out calculations concerning lunar motion, while in the nineteenth century C.-E. Delaunay devoted to it twenty years. In particular Morley's theorem was proved using a CAS.

The chapter ends by a discussion whether there can exist theorems that have only long proofs. However, I found this discussion misleading. First, the posed question about the existence of such theorems was already answered on the previous page. I believe the author's question referred to 'natural', or 'published' theorems. I miss some in-depth discussion on this topic. For instance, the answer depends on the underlying proof system and the definition of 'long'. The first theorem of this kind was established in A. Haken, *The intractability of resolution*, Theoretical Computer Science, 39, pp. 297-308, 1985, where it was shown that there exists a family F of propositional formulas such that for every $\phi \in F$ the length of the shortest resolution proof of ϕ cannot be bounded by any polynomial of the length of ϕ. Since then more results of this kind were established, in particular in S. Buss and G. Turán, *Resolution Proofs of Generalized Pigeonhole Principles*, Theoretical Computer Science, 62(3), pp. 311-317, 1988, an exponential lower bound on the length of these proofs was given.

Alternatively, these results could have been discussed on the last two pages of the book. The author states there "Besides, the philosophical debate about the link between proof and explanation would be greatly clarified if one could state with certainty that a specific theorem has no short axiomatic proof". Well, the result of Buss and G. Turán seems to provide an answer, unless the author again had in mind some 'natural' specific theorem.

Returning to the discussion of Chapter 12, the author correctly points out that the true but unprovable (in Peano's arithmetic) statement constructed by Gödel is artificial but fails to mention that in the 1970s Paris and Harrington found a natural true but unprovable statement. Instead, the result of Cohen on the unprovability of the continuum hypothesis (in the ZFC set theory) is mentioned. This is confusing since Gödel's incompleteness theorem is discussed within the context of set theory.

Chapter 13 is devoted to a discussion on the role of instruments in sciences. It is pointed out that until 1976 (what happened in this year is not explained; I guess this is a reference to the proof of the Four Colours theorem discussed in the previous chapter) mathematics was practically the only science that did not use instruments. This changed when the computers started to be used. The author mentions that in other sciences the use of instruments led to a qualitative jump in knowledge and posits that the same happened in mathematics. The chapter mentions an interesting paradox: the use of tools (in the case of mathematics – a computer) ought to make the obtained results less reliable, since the tools can be faulty (think of a hardware fault). Yet, the use of automated theorem provers (so indirectly, of computers) uncovered some mistakes in the existing proofs. The point is that mathematical proofs are not formal proofs in the logic sense.

The final remarks are concerned with the reasons why the idea of specialised computers devoted to, for example, symbolic computing did not survive beyond the 70s. However, this discussion is incomplete as no reference is made to the use of supercomputers that still are being built for the purpose of executing programs involving intensive calculations.

In the short Chapter 14 the author discusses his own work with T. Hardin and C. Kirchner that concerns an extension of first-order logic in which the proofs are constructed with axioms, proof rules and computation rules. But the explanation is so sketchy that it could equally well fit a description of various well-studied extensions of logic programming, for example with the equality theories.

This remark actually applies to a number of other systems, logics and calculi discussed in the book. I am, for instance, not sure whether somebody who has never heard of lambda calculus will be able to understand its essence and basic concepts, in particular the concept of self-application. The idea to introduce such concepts to a 'general reader' is certainly praiseworthy, but one has to be realistic how much can be achieved using an informal style. In any case the main message of the book, that mechanised proofs are important and how we reached a stage that they are feasible, is convincingly presented.

The above summary of the book contents may sound a bit like a student's homework, but avoiding details would turn this review into a collection of loose remarks and would do injustice to author's systematic and lucid exposition.

Of course, in the period since the book originally appeared, so 2007 and today a number of new results relevant to this book were established. The only place where I noted a small effort to make the book up to date is the reference in Chapter 12 to the Flyspeck project the aim of which was to validate the proof of Kepler's conjecture and that was concluded in 2014.

I am not an expert in the area of automated theorem proving, but find that it would have been useful also to mention that Gödel's incompleteness results were finally validated by means of an automated theorem prover in L.C. Paulson, *A Mechanised Proof of Gödel's Incompleteness Theorems Using Nominal Isabelle*, Journal of Automated Reasoning, 55 (1): pp. 1-37, 2015. Paulson uses the proofs of S. Świerczkowski, *Finite sets and Gödel's incompleteness theorems*, Dissertationes Mathematicae, 422: pp. 1-58, 2003, that employ hereditarily finite sets instead of the original Gödel's encoding in natural numbers.

Let me conclude with some minor complaints about the book under review. Given that the author discusses in the book lambda calculus, automated theorem proving and unification it is surprising that there is no single reference to logic and functional programming. A natural place for a short chapter on these two approaches to programming based on logic would be just after the discussion of the automated theorem proving in Chapter 10.

The book does have a chronological listing of the relevant dramatis personae, but misses an index and an updated bibliography. Many bibliographic entries are in French. It would have been useful to mention the corresponding translations and/or natural counterpart books in English. I am not a native English speaker, but I get an impression that the translation is excellent (I did spot one strange slip though: "[. . .] he goes continues (sic) to explore [. . .] and one mistake: 'analogical computer' should have been 'analog computer').

One book not mentioned is the excellent M. Davis, *Engines of Logic*, W.W. Norton & Company, 2000, which traces the history of logic starting with Leibniz and ending with the construction of computers. It provides an alternative historical account of the period starting with Leibniz and ending with Turing, providing a more in-depth historical perspective.

Putting these and earlier mentioned shortcomings aside, it is a highly readable and most useful account of history of the interplay between logic, mathematics and computing, with the emphasis on the importance of mechanisation of proofs.

A second, revised and more up to date edition, would be a good idea.

Krzysztof R. Apt
CWI, Amsterdam, The Netherlands