Book review

Rosella Gennari

Mathematical Logic
Wei Li
Birkhäuser
Hardback, ISBN-978-3-7643-9976-4

1 Introduction

The book is entitled "Mathematical Logic", a title that will remind any logician of notorious predecessors, like (Enderton, 2001) or (Mendelson, 1997). However, the subtitle and back cover help us collocate the target of the book: to show how mathematical logic can "provide a foundation for the development of information science and technology" to undergraduates and graduates in information science, as well as to interested researchers. As such, the book should be compared to existing textbooks on mathematical logic for computer scientists, like (Bradley & Manna, 2007) or (Huth & Ryan, 2010).

The book is divided into two main parts: the first five chapters present core concepts of mathematical logic; starting from there, the remaining chapters present the author own work in mathematical logic. The table of contents is briefly summed up as follows:

- Part I:
  1. syntax of first order logic;
  2. model theory;
  3. formal inference systems;
  4. computability and representability;
  5. Gödel theorems;

- Part II:
  6. sequences of formal theories;
  7. revision calculus;
  8. version sequences;
  9. inductive inference;
  10. workflows for scientific discovery.

This review first analyses the contents of the two parts, and then tries to asses who could profit from reading the book.

2 Part I: Prerequisites of Mathematical Logic

This part is a quite concise presentation of core concepts of mathematical logic. The author should be praised for his survey effort. Warnings are due, however, if the readers are undergraduates or, more in general, novices in mathematical logic. The choice and presentation of material in mathematical logic is, in fact, quite ad hoc for the second part of the book, with some exceptions. Some specific considerations follow.

Section 2 presents the reader the classical definition of first order formulae; no mention to propositional formulae is made here. No mention to parsing trees is eiether made whereas, in modern popular books of logic for computer scientists like (Huth & Ryan, 2010), structural induction is usually explained through parsing trees. An excellent example in this respect is (Chiswell and Hodges, 2007).

Quite surprisingly for a book meant for information science students and researchers, the only proof calculus presented in the first part of the book is a Gentzen-style calculus. This said, resolution is strangely introduced in the second part of the book, and restricted to "predicates with no variables" and the "empty sentence"... After going through all the details of the Gödel coding, the semi-decidability of the Gentzen calculus is only informally reported in a rather compressed sentence in p. 57, and only briskly revisited in p. 116 after presenting the halting problem.

Similarly, considerations like the "repeated" usage of a (function) symbol is "a characteristic of structural induction definitions" or "the advent of formal proofs makes it possible to convert proofs of mathematical theorems into a symbolic calculus which can be accomplished by computers within a man-machine interactive software system" may mislead instead of helping the intuitions of novices in logic.

3 Part II: Formalisation of Axiomatisation

The work presented in Part II is motivated in the preface of the book by the following claim: "deductive logic was well analyzed, the process of axiomatization had not been studied in depth". Reading it, the (logically inclined) reader can be rather unsettled at first. She needs to be patient until the opening of the second part of the book in order to grab the meaning of the claim.

The second part opens with the author's view on the process (or method) of scientific research, which reminds the reader of Popper method (empirical falsifiability). However, no mention to this or more recent debates in philosophy of science is made in the opening to the second part. For the benefit of the wide audience to which the book aims, the author could frame and ground his views and work within this area, and properly cite sources.

The first goal of the work in the second part is that of formalising the inductive process of generating new axioms and that of revising theories in the process of scientific research. The second is to "formalize the description of research methodology and and analyze the properties of sequences of theory versions". I am not an

expert of this new work and, to me, crucial notions, albeit interesting, seem in want of stronger motivations and comparison with existing work.

The proposed framework of (version) sequences of formal theories should account for "describing the evolution of domain knowledge". This sounds very interesting. What the reader misses is a throughout comparison with existing work, showing the novelty of the proposed framework. For instance, after Lemma 8.2, which states that "default sequences" (mentioned in Section 6.5) are special cases of version sequences, the reader is left alone to wonder what is crucially novel and relevant with respect to Reiter default logic, or other epistemic logics.

The same notion of sequence of formal theories sounds quite generic—any countable set of theories. A similar remark applies to other crucial notions like that of "new conjecture" for a formal theory, which is defined as any formula A so that both A and $\neg A$ are consistent with the theory, and that of "new axiom", which is any formula A independent of the theory.

The examples the author draws from various scientific or technological fields aim at grounding the intuitions of his new work, and as such they are potentially useful. Sometimes, however, they may mislead the reader. In the example of software development in Chapter 6, the author parallels "the new functionality" for a software version to a "new axiom". Why so (and not, say, to a new formula consistent with the theory)? More in general, it is difficult to understand why the author refers to software prototyping and testing as key areas for providing new concepts to mathematical logic, without ever mentioning software verification, in which computer science and (computational) logic have already much benefitted one from the other, e.g., think of dynamic logic, (Harel *et al.*, 2000).

The references to various branches of science reveal the wide spectrum of interests of the author, for which he should be complimented. Some quotations may still require the revision of experts of those branches, e.g., "a photon is a rigid body" sounds counterintuitive (a rigid body is an idealisation of a solid body, and has properties that a photon does not seem to satisfy). Minor typo: the title of the book by Galileo Galilei has "copernico" instead of "copernicano".

4 The Ideal Reader

Reading through its examples or observations, the book seems to aim alternatively at a computer-science reader, novice in mathematical logic, and at times at a mathematical logician, curious about applications of mathematical logic to computer science. Some examples seem in fact written for the computer scientists willing to explore (mathematical) logic, e.g., "the difference between programs and their compiled executables is precisely the same as that made between first-order languages and their models". Elsewhere, when discussing how concepts born in mathematical logic "like structural induction" found their applications in computer science, the book seems to talk to the mathematical logician.

Writing a mathematical logic book for such a varied audience is definitely a difficult job, and some readers may easily stumble on the more technical notions outside of their expertise field. A novice in mathematical logic may feel at loss

when facing the digest of mathematical logic given in the first part of the book. On the other hand, the mathematical logician may find it hard to grab the authors' motivations for the new work, presented in the second part of the book, whenever grounded in notions or methods of software engineering.

Overall, the book seems more suited to the logician that is interested in the author work, presented in the second part of the book. For the others, an introductory book to mathematical logic like (Enderton, 2001) may help complement the first part of the book.

References

Bradley, A.R., & Manna, Z. (2007). *The Calculus of Computation.* Springer.

Enderton, Herbert B. (2001). *A Mathematical Introduction to Logic.* Harcourt Academic Press.

Harel, D., Kozen, D., & Tiuryn, J. (2000). *Dynamic Logic.* MIT Press.

Huth, Michael, & Ryan, Mark. (2010). *Logic in Computer Science.* Cambridge University Press.

Mendelson, Elliott. (1997). *Introduction to Mathematical Logic.* Chapman and Hall.