Irish Math. Soc. Bulletin
Number 92, Winter 2023, 65–67
ISSN 0791-5578

Y.A. Gonczarowski and N. Nisan: Mathematical Logic through Python, Cambridge University Press, 2022. ISBN:978-110-8949-47-7, GBP 22.99, 284 pp.

REVIEWED BY TIM SWIFT

Mathematical logic is one of the most significant sub-areas of mathematics for computer science. For many working mathematicians, the subject is perhaps less important, even though, of course, an important everyday task in mathematics is to show the truth of a proposition by correctly deriving it from appropriate axioms using the rules of logic. Thus, a course on mathematical logic is a core component of most undergraduate computer science programmes, but only sometimes appears in undergraduate mathematics programmes, and then usually as an option module that might also include other foundational aspects.

The book under review, which has been written mainly for computer science students, but also contains much of interest for mathematics students, is an impressive contribution to a literature that already includes many excellent texts on mathematical logic. Although written from a modern viewpoint, it might be said that the book also tips a hat at some of the twentieth-century roots of the subject, in particular the work of Alonzo Church and Alan Turing, so important for the development of computer science. Via an imaginative pedagogical approach based on the Python programming language, the book covers a one-semester course on propositional and first-order predicate logic, up to and including Gödel's Completeness Theorem. The underlying idea of the book is that the student develops their knowledge and understanding by writing a carefully structured sequence of working programs in Python that implement both logical concepts and mathematical proofs. The authors' intention is that such an approach should appeal to computer science students, equipped with their existing programming expertise, more than would a traditional mathematical development of the subject via the proving of theorems. The authors carefully explain their pedagogical approach and why they have developed this method of teaching logic, which is an outcome of their experience of delivering the subject to undergraduate computer scientists over several years.

The book covers the standard topics in propositional logic, namely syntax, semantics, proof, the Tautology Theorem, and completeness, and then follows a similar path for predicate logic. The final chapter provides a preview of what a second course along the same lines would cover, and, in particular, an introduction to Gödel's Incompleteness Theorem is provided. The authors estimate that the 150+ programming exercises cover about 95% of the mathematical content of a standard first course in mathematical logic. The remaining content, mainly results that involve infinite sets, is treated in the traditional mathematical fashion. Given that, according to several indices, Python is currently the world's most popular and fastest-growing programming language, the use in the book of this open-source language is entirely appropriate and should help to ensure widespread appeal. Although the text is aimed primarily at computer science students, it could also be used as a text for mathematics undergraduates, perhaps used

Received on 18-10-2023; revised 01-11-2023.
DOI:10.33232/BIMS.0092.65.67.

in parallel with a more traditional book on mathematical logic. Another use of the book might be as a resource for a final-year undergraduate mathematics project that combines developing an account of mathematical logic with associated investigations based around Python programming.

Beyond the innovative development of the material, there are several other features that should make the book useful from the learning and teaching perspectives. These include: an introductory overview chapter, which contains an explanation of how to use the book, in particular the Python exercises; a useful roadmap; footnotes that make connections with other areas of mathematics; optional reading sections at end of each chapter; the summary list of axioms and inference rules at the end of the book; a comprehensive index; extra online resources including skeleton code and unit tests for all of the programming exercises. In addition, it is good to see the mathematical examples that are used to illustrate proofs in predicate logic.

This reviewer, as a mathematician who is far from being an expert in Python programming, worked through some - but not all! - of the exercises, which was very instructive; indeed, because more Python skills had to be acquired, my engaging with the book certainly had an additional flavour of 'Python through mathematical logic'. I feel that the pre-knowledge of Python required could be better described: the authors state only that '... basic proficiency in Python is assumed', and perhaps a brief list of particular skills could have been provided. On the other hand, the Python exercises are carefully selected, enabling the traditional mathematical development to be followed, and I found the authors' suggestion to consult the examples within the test code for a given task before attempting a particular exercise to be very good advice.

On the whole, the notation and terminology are standard, and, where differences occur, these are pointed out. A mathematics student using the text should take note of a few differences between mathematics and computer science terminology, e.g., operator, rather than operation. Also, it was slightly disconcerting to see the plus symbol used for the binary operation in a possibly non-abelian group. The method of numbering the environments, e.g., there are Definition 12.1, Lemma 12.1 and Theorem 12.1, is not to this reviewer's taste, and makes navigation a little more difficult.

In summary, I feel that this publication is a very useful addition to the existing library of mathematical logic textbooks. It should certainly be helpful, as intended, to computer science undergraduates, but also to mathematics students, and, indeed, to mathematics lecturers who wish to integrate Python coding in a meaningful and important way in their teaching.

Finally, it is worthwhile noting the appropriate choice of cover illustration, namely Wassily Kandinsky's wonderful painting 'Serious- Fun'. The authors conclude the Preface by remarking that they hope that the name of this picture will describe the reader's experience as they work through the book. It is interesting to note that Kandinsky made the painting in 1930, the start of a crucial decade for mathematical logic, during which workers such as Gödel, Tarski, Carnap, Church, and Turing made radical contributions to the development of the subject. For this reviewer, at least, 'Serious-Fun' might be a brief, but accurate, description of engaging with the mathematics game itself.

Tim Swift Tim Swift is a senior lecturer in mathematics at the University of the West of England, Bristol. Born in the West Riding of Yorkshire, he was educated at the Universities of Cambridge and Southampton. He has worked in general relativity theory and differential geometry, and currently has interests in graph theory, stochastic processes and mathematical

culture. As undergraduate programme leader, he is interested in curriculum development and in increasing diversity in mathematics.

Mathematics and Statistics, University of the West of England, Bristol
E-mail address: tim.swift@uwe.ac.uk