doi:10.1017/S1471068410000037

Book review

Handbook of Practical Logic and Automated Reasoning, John Harrison, Cambridge University Press, 2009. Hardcover, ISBN-13: 978-0-521-89957-4, 681 pp. + xix, $135.00.

1 Overview

Contemporary research in computer science has produced an abundance of formal methods designed to enable hardware and software systems to reason correctly, and to enable us to reason better about these systems. Indeed, the explosion of research and specialized techniques can make it hard for students and newcomers to enter the field. John Harrison's *Handbook of Practical Logic and Automated Reasoning* is a significant addition to the expository literature on the subject, and will serve as a valuable resource for beginners and experts alike.

Almost 700 pages long, the book is remarkable in many ways, including the breadth of its scope, the depth of its insights, and the clarity and readability of its prose. It is somewhat hard to classify, and the title does not help. To start with, it is not a handbook; while it is more than just an introductory text, it is not designed to be a comprehensive reference. And while Harrison does focus on implementation and practical issues, he does not skimp on the underlying theory; indeed, the text passes between the two poles with surprising fluidity. The book does not assume background in logic, and appendices review the requisite background in mathematics and programming. As a result, the text is accessible to a broad audience, including sufficiently advanced undergraduates.

The title also fails to provide a clear delimitation of the book's scope. These days, modal and temporal logics are used in writing hardware and software specifications, say, to describe the desired behavior of a circuit or network or security protocol; fuzzy, defeasible, description, and default logics are used in knowledge representation and artificial intelligence; various logical languages and frameworks form the basis for managing database operations and queries; higher order logic, sometimes in the form of simple or dependent type theory, is implemented in a number of interactive theorem provers; other logical frameworks are used to specify and manage constraints which are then processed and solved automatically. Even methods of inductive and statistical inference may reasonably be taken to be subsumed by the phrase "practical logic and automated reasoning." But none of these are discussed in Harrison's book.

Rather, the book's focus on *mathematical* logic, that is, the core logical principles that underly mathematical reasoning. Classical first-order logic is the overarching framework, but Harrison also devotes considerable effort to describing automated methods of reasoning about mathematical structures, both concrete (such as the natural numbers, the real numbers, and the complex numbers) and abstract (groups, rings, and algebraically closed fields). This is a reasonable choice: historically and conceptually, mathematics plays a central role among the many patterns of reasoning we employ, and the other types of reasoning alluded to above can often be embedded in a mathematical context via syntactic or semantic modeling. The book thereby covers an important core of the subject, and provides a solid foundation for automated reasoning more broadly construed.

The choice of subject matter is reflective of Harrison's own interests, which lie in a branch of formal verification known as "interactive theorem proving." He is well known as the author of a proof assistant, HOL Light, which holds its own against systems developed by much larger research teams. Some of the most complex mathematical theorems verified to date have been verified in that system; Harrison himself has, for example, formalized complex–analytic proofs of the prime number theorem and Dirichlet's theorem on primes in an arithmetic progression, and Thomas Hales has verified the Jordan curve theorem. Moreover, Hales is currently directing an ambitious project to verify his 1998 proof of the Kepler conjecture using, primarily, HOL Light. (For an overview of interactive theorem proving and these topics, see the December 2008 issue of the *Notices of the American Mathematical Society*.) But there is also an applied side to this research: Harrison works at Intel, where he uses HOL Light to verify circuit design. As a result, his work is a place where pure mathematics and industrial application come together.

Not surprisingly, then, the automated methods treated in the book are those often used by contemporary proof assistants (with the caveat that proof assistants like HOL Light bring the additional burden of constructing formal proof objects to back up the conclusions reached). The book presents decision procedures for axiomatic theories and structures when it is possible to do so, as well as search procedures in situations where the problem of determining the validity is undecidable. Only classical logic is considered, and devotees of intuitionistic logic may feel somewhat slighted. But in showing us how classical methods can be modeled effectively, the text is highly relevant to constructive and computational mathematics as well. A discussion of higher order reasoning and automated methods of proof search in that setting would have been natural in this context; given that HOL Light is based on higher order logic, it seems likely that the omission is simply due to time and space constraints.

One of the most striking features of the book (and one of the factors contributing to its length) is that all the algorithms are implemented in Ocaml. The code is actually worked into the exposition, and is available on the author's web page for further experimentation. These snippets do not replace mathematical content, and fundamental properties of the logical systems considered, like the completeness and compactness of first-order logic, are stated explicitly and proved. Moreover, the mathematical claims needed to justify the correctness of the algorithms that are implemented are often stated as lemmas and theorems and proved as well. Conversely, the programs themselves can be taken to establish computational claims; for example, a decision procedure for propositional logic is itself a proof that

https://doi.org/10.1017/S1471068410000037 Published online by Cambridge University Press

propositional logic is decidable. There is thereby a synergetic relationship between the mathematical theory and the explicit implementations. Given the tight link between theory and implementation, one might have expected the mathematical content to have a very syntactic flavor, laden with the kinds of deductive calculi and rules of inference that are so common to the proof theory literature. In fact, Harrison tends to favor semantic approaches, giving model-theoretic arguments whenever possible. So despite the implementations, one finds explanations and justifications that stand largely independent of a particular choice of formalism.

Although the book is not designed to be a comprehensive reference, Harrison does a very good job of providing broader context and pointers to the literature. Each section ends with copious notes and references. As a result, the book can serve as a helpful introduction to the contemporary literature, like having a friendly uncle working in the field. Confused about the difference between the Davis–Putnam algorithm and Davis–Putnam–Logemann–Loveland (DPLL)? Uncle John will set you straight. Vaguely aware that there are important differences between "local" and "global" search methods, but not sure what is at stake? Uncle John makes it clear. Long been wanting to learn something about Gröbner bases and real closed fields, but find the literature on computational algebra daunting? Here, too, Uncle John will help you out. Heard that "combination methods" are currently all the rage, but at a loss as to what they are? Once again, Uncle John will tell you what you need to know to get started. The technical details are couched in a wealth of historical information, and the bibliography cites original papers by Russell, Post, Skolem, Łukasiewicz, Turing, Presburger, Gödel, Quine, Wittgenstein, and Robinson, alongside more recent work. Each chapter ends with a list of interesting and carefully chosen exercises. The prose is friendly yet crisp and precise, and a pleasure to read.

2 Contents

The book contains a preface and seven chapters, followed by three appendices, bibliography, and index.

The short first chapter provides a genial introduction to symbolic logic and formal systems, and, pragmatically, addresses syntactic tasks like parsing, and pretty printing.

Chapter 2 is devoted to propositional logic. It covers its syntax and semantics (and, of course, implements them), and introduces canonical normal forms like disjunctive normal form and negation normal form. The chapter then discusses some of the most important decision procedures for propositional logic: the Davis–Putnam and DPLL procedures, Stålmarck's method, and binary decision diagrams. The chapter employs an expository strategy that is employed throughout: with respect to each algorithm, Harrison presents (and implements) a prototypical and illustrative version, but then goes on to discuss variations and optimizations, providing the reader with a good sense of the range of possibilities.

Chapter 3 moves to first-order logic, including the syntax and semantics, prenex and Skolem normal forms, and Herbrand's theorem. A discussion of unification then paves the way to a discussion some of the most important classes of search

https://doi.org/10.1017/S1471068410000037 Published online by Cambridge University Press

procedures, including tableau search, resolution, and model elimination, as well as Prolog-style backchaining for Horn clauses. This is the second-to-longest chapter in the book, and one where Harrison's erudition really shines, as he helpfully compares and contrasts the different approaches.

Chapter 4 deals with equality, in the full first-order setting as well as restricted to the quantifier-free fragment. On the theoretical side, we get the completeness of standard first-order calculi for first-order logic with equality, as well as Birkhoff's completeness theorem for purely equational logic. The book also covers the congruence closure decision procedure for quantifier-free equational logic; rewrite systems and various important term orderings; Knuth–Bendix completion, which often makes it possible to transform an equational theory into a confluent rewrite system, and paramodulation, which can be used to extend resolution proof search to theories with equality.

Chapter 5 is the longest chapter in the book. It deals with the decision procedures for fragments of first-order logic (specifically, the "AE" fragment and monadic first-order logic), as well as decision procedures for various axiomatic theories, including linear arithmetic, Presburger arithmetic, algebraically closed fields of characteristic zero, and real closed fields. (These last four can be characterized, respectively, as the theory of the reals as an ordered group, the theory of the integers as an ordered group, the theory of the field of complex numbers, and the theory of the reals as an ordered ring.) Harrison explains why theories that enjoy the finite model property are decidable, and how the method of quantifier elimination provides a general strategy for establishing decidability in many instances. Once again, Harrison presents archetypal illustrative algorithms that are not the most efficient available, but illustrate key ideas. For example, decision procedures for real closed fields are notoriously infeasible, and the Cohen–Hormander algorithm that Harrison implements only works on small examples. But it has the advantage of being (relatively) easy to understand, and with Harrison's explanations in hand, hardy readers will be much better equipped to read about Collins' cyclindrical decomposition algorithm elsewhere. The chapter also introduces the theory Gröbner bases, and ably explains how this yields a decision procedure for the universal fragment of the theory of integral domains, or, equivalently, the theory of fields. The discussion of Gröbner bases also lead naturally to decision procedures for fragments of Euclidean geometry and Wu's method. Finally, the chapter describes Nelson–Oppen combination methods, which, under suitable conditions, allow one to amalgamate decision procedures for the universal fragments of various theories to obtain a decision procedure for the universal fragment of their union.

Chapter 6 deals with interactive theorem proving, where the goal is not only to establish the validity of a mathematical assertion, but, essentially, to construct a formal deductive proof at the same time. Here it is impossible to avoid dealing with proof systems and rules of inference to some extent. As the word "interactive" suggests, proving nontrivial theorems often requires substantial guidance from the user. Harrison discusses the "LCF" framework, where one typically constructs proofs by iteratively applying rules that iteratively reduce the task at hand to smaller subgoals. But he also discusses declarative models of proof, whereby users

https://doi.org/10.1017/S1471068410000037 Published online by Cambridge University Press

provide input that is closer in form to ordinary mathematical proofs. Either way, it is advantageous to rely on automated procedures to fill in details along the way, and Harrison shows how implement a tableau theorem prover in an interactive theorem proving framework.

The final chapter, Chapter 7, is titled "Limitations," and discusses the striking negative results of twentieth century logic: Tarski's theorem on the undefinability of truth, Gödel's first and second incompleteness theorems, the undecidability of the halting problem, and the undecidability of first-order logic and various axiomatic theories. True to form, Harrison never overlooks an opportunity to illustrate logical ideas with code. Thus, we get explicit Ocaml functions that construct self-referential sentences, evaluate Turing machines, and prove true Σ_1 formulas. Using the technology introduced in the previous chapter, we also get a formally constructed proof of the second incompleteness theorem for system satisfying the Löb derivability axioms. Traditional results in mathematical logic thereby take on a new life; but even setting the code aside, Harrison's exposition of these topics is wonderfully cogent and to the point.

The three appendices, in turn, provide general mathematical background, a brief introduction to programming in Ocaml, and the unglamorous details behind the parsing and pretty printing routines that are needed for the book's code. The detailed bibliography is more than 35 pages long.

3 Conclusions

The *Handbook of Practical Logic and Automated Reasoning* is not meant to function as a substitute for more specialized and exhaustive handbooks in automated reasoning, artificial intelligence, computational algebra, constraint programming, and so on. Rather, it is meant to serve as an introduction and guide to topics which form a common core to all these pursuits. It provides a lucid and synoptic overview of these topics, conveys a solid theoretical background, provides tools for experimentation, and offers notes and pointers that facilitate a smooth transition to the contemporary literature.

The book currently priced at $135, or about 90 Euros. The price is somewhat steep, but not unreasonable, given the book's size and scope. Anyone working in automated reasoning, or looking to come up to speed on developments in the field, will want to have a copy at hand.

Jeremy Avigad

https://doi.org/10.1017/S1471068410000037 Published online by Cambridge University Press