Book Reviews

Logic for Applications. By *Anil Nerode* and *Richard A. Shore.* Springer-Verlag, 1993. xvii+365 pages. ISBN 0-387-94129-0. $39.95.

Nerode and Shore's *Logic for Applications* must surely rank as one of the most fruitful textbooks introduced into computer science and related areas. It covers in broad strokes the fundamental material in logic with emphasis on classical propositional and predicate logic, and nonclassical logics.

In the preface of the book, Nerode and Shore give a road-map in the form of a logical dependence tree between sections of the text. Although each chapter is organized in a way to allow independent reading, the book is far from being eclectic.

Chapter I (59 pages) starts with an introduction to the notion of trees, and proceeds with the syntax and semantics of classical propositional logic. Nerode and Shore pursue the semantic tableau proof method. The resolution principle for theorem proving is introduced at the end of the chapter. This chapter also covers resolution refinement, in particular linear resolution which specializes to Horn clauses, thus forming a basis for logic programming in PROLOG.

Predicate logic with its syntax and semantics is presented in Chapter II (102 pages). This chapter treats the quantifiers "for all" and "there exists", omitted in Chapter I, and has a redefinition of formation trees for predicate logic formulas. Explanation of skolemization provides a nice transition to show how predicate logic can be reduced to propositional logic via Herbrand's theorem. Then, unification is described and is combined with resolution. This turns out to be the starting point for the authors to focus on deduction for predicate logic as a form of computation. The semantic tableau proof method is solely used to prove the soundness, completeness, and compactness theorems for propositional logic and predicate logic.

Basic concepts of Horn clauses and foundations of logic programming, explained in Chapter I, are elaborated for predicate logic in Chapter III (62 pages). Although the main emphasis is on PROLOG, this chapter covers common topics in logic programming such as negation (as failure), equality, searching, and backtracking. Relating negation on the right level with a nonstandard form of reasoning, i.e., nonmonotonic inference, Nerode and Shore establish appropriate grounds for treating stable model semantics. Readers interested in grasping the theoretical underpinnings of logic programming can benefit from this chapter.

In the last decade, nonclassical logics have gained considerable importance in building intelligent systems that overcome real-life problems. Contrary to classical logics, truth of a sentence in nonclassical logics may depend not only on the truth of its parts but also on factors such as context, intensions, and beliefs. Computer science researchers have a long-standing interest in these logics. The last two chapters are devoted to two such logics: modal logic (Chapter IV, 41 pages) and intuitionistic logic (Chapter V, 46 pages). The semantics of these logics are described in terms of Kripke frames which are obtained by associating a partial ordering (preference criterion) on the classical models. Nerode and Shore's treatment of these logics is again based on the tableau method. (As an aside, the first reviewer learned for the first time the application of this method in the lectures of Nerode during the *International Summer School on Logic, Algebra, and Computation*, Marktoberdorf, Germany, 1989.) Individual sections are devoted to the soundness and completeness of modal logic and intuitionistic logic in their respective chapters. A section added to the end of Chapter V helps the reader realize the essential differences between these logics.

Nerode and Shore supply an indispensable section on the history of logic (16 pages) in the appendix; this has not been considered to be crucial by most logic textbooks. By drawing an analogy between the history of calculus and the history of mathematical logic, a solid review of the origins and development of logic including the twentieth century logic, deduction, and computation is given. An extensive collection of further readings attached to the end of this section together with a 7-page bibliography on the history of logic make a useful reading list for the interested reader.

A well-prepared section on suggested readings finalizes each chapter. Nerode and Shore arrange the bibliography with respect to subjects: history of mathematics and logic, mathematical logic, intuitionistic, modal, and temporal logics, and logic and computation. Each of these is further divided into bibliographies of related topics. The bibliography gives capsule reviews of the most important books and papers. Suggestions on further readings and a fairly extensive annotated bibliography draw a clear guideline for not only students but also for researchers already possessing a fair amount of knowledge about these domains. An important aspect of *Logic for Applications* is that it provides elegant explanations for decidability issues at the right places throughout the text.

Each section is accompanied with a good deal of assorted examples and exercises clarifying the covered material. Logical expressions, symbols, and notations are introduced in a consistent manner. The visually pleasing format of illustrations and PROLOG code segments save the textbook from the boring characteristics of some classical logic books. Theorems, propositions, and lemmas are mostly proven fully; only a small set of proofs is left as exercises. We could not spot any errors, even typographical ones, in the text and commend the authors for their precision and attentiveness to minute detail.

In an on-line document (*ASL Guidelines for Logic Education*, June 23, 1994) that we have obtained via Internet, the *Association for Symbolic Logic (ASL)* adopts some guidelines for logic instruction at various levels of education. They recommend institutions of higher learning to offer a course (or a sequence of courses) to cover the following:

- Elementary facts about sets.
- Basic facts about inductive definitions and proofs by induction.
- Propositional and Predicate Calculi (the formalization of informal arguments, the axiomatic method in mathematics and science).
- Semantics (truth and validity, definability, the Soundness Theorem, the notion of consistency, the Gödel Completeness Theorem).
- An introduction to model theory (at least the Compactness Theorem for countable languages).

- The Gödel Incompleteness Theorems and their philosophical consequences.

Set theory falls out of the main scope of *Logic for Applications*. Gödel's Completeness and Incompleteness Theorems are discussed only to the extent a historical review permits. Fundamental notions of model theory are presented in a comprehensive manner in Chapters I–III. As also noted by the authors, all the material in these chapters can be covered a in one-semester course for advanced undergraduate students. Nerode and Shore try to minimize the need for supervision of an instructor to explain the basic ideas in the classroom. But a prospective undergraduate student should not expect to learn logic from this book by way of self-study.

In addition to the core topics above, ASL also recommends, among others, the following:

- An introduction to proof theory (e.g., Natural Deduction, Herbrand's Theorem).
- Some additional model theory (e.g., Löwenheim-Skolem theorems for countable languages, the non-expressibility of various mathematical notions in first-order logic).
- An introduction to computability theory (machine models of effective computability, Church's Thesis; unsolvable problems).
- An introduction to other kinds of logic (e.g., intuitionistic logic, higher-order logic, modal logic, temporal logic, etc.).
- An introduction to uses of logic in computer science (e.g., unification and the resolution method, and the λ-calculus and its connections to LISP).

Logic for Applications as a whole covers most of these topics. The main thrust of the book is commendable. We strongly suggest it as a textbook for courses in which students preparing for further study in computer science, philosophy, cognitive science, or linguistics can be exposed to the above topics.

Erkan Tın and Varol Akman
Department of Computer Engineering and Information Science
Bilkent University, Bilkent, Ankara 06533, Turkey
E-mail: {tin,akman}@bilkent.edu.tr