Journal of Logic, Language, and Information **7:** 228–229, 1998.

Book Review

Logic for Applications, Anil Nerode and Richard A. Shore, Graduate Texts in Computer Science, New York: Springer-Verlag, 1997 (2nd edition). Price: DM 78.00, xiii + 438 pages, Index of symbols, Index of terms, ISBN: 0-387-94893-7.

In this impressive monograph, one finds a thorough approach to logic in five chapters, organised along the following topics: *Propositional* and *Predicate Logic*, PROLOG, *Modal* and *Intuitionistic Logic*. Then, there is a sixth chapter (*Elements of Set Theory*), which, according to the diagram of dependencies between chapters, can be read and studied more or less independently from the other chapters.

Appendix A provides a historical overview of logic and the foundations of mathematics (to appreciate this fully, the reader might also want to read the first six sections of the chapter on set theory), whereas Appendix B provides a genealogical database of PROLOG facts of the form "fatherof(a,b)," based on the Chronicles from the Hebrew Bible. These facts are used for various programming problems and exercises.

Finally, the book contains an extensive bibliography, ordered by subject, and an index of symbols and one of terms. Moreover, exercises are provided at the end of each section, and each chapter ends with "Suggestions for Further Reading."

Whereas a computer scientist's approach to logic is mirrorred in tableaux proofs, resolution and unification in the first two chapters, in Chapter III on PROLOG the real computational application starts with the specialization of resolution to Horn clauses. Negation as failure is then used as a bridge to a brief introduction to nonmonotonic logic. In spite of this single section on the topic (one nonmontonic formal system is introduced, similar to default logic), the authors are able to apply their setup to stable models. Chapters IV and V are devoted to nonclassical logics that are important for reasoning about computation: Intuitionistic and Modal Logic.

The second edition of 1997 differs from the 1993 first edition in having the above mentioned Chapter VI on set theory. This chapter can be ". . . either used as a reference to standard set-theoretic notations and concepts . . . " but is ". . . also a self-contained introduction to axiomatic set theory." The first six sections of this chapter treat standard notions from elementary set theory like functions, relations, orderings (the concept of order and, more specifically, tree, is also introduced in Section I.1) and sequences, whereas the remaining five sections are devoted to transfinite induction, ordinals, cardinals and variations on the axiom of choice. Of course, in adding a specific foundational chapter, dangers of bootstrapping problems always arise. For instance, in order to formalize set theory, some notation from predicate logic is already presumed.

Another, more dangerous example is given by the treatment of the principle of induction. In Section I.2, it is demonstrated how to perform a proof by induction on the structure of formulas. It is left as an exercise in Chapter I to show that this method can be related to the procedure of induction on numbers. However, the latter procedure is only defined in Section VI.4, in a rather abstract setting, that of inductive sets.

According to its cover the book is ". . . a first introduction to mathematical logic [. . .] no previous exposure to logic is assumed . . . ." The introduction says that the book aims at upper level undergraduate and beginning graduate students of mathematics or computer science. I like to underscore that such students should at least be familiar with some mathematical notation and also some standard

arguments in reasoning. In my (Dutch) context of teaching logic, I see a tendency to introduce logic in a semi-formal way: students should play with reasoning patterns, get experienced with manipulations on quantifier sequences and should be able to easily come up with counterexamples for incorrect arguments. The focus of the book under review is not so much on these techniques, and personally, I would prefer to use it in an advanced logic class.

The book is definitely written in the spirit of what the authors see as the *applications* of logic to computer science: instead of reasoning patterns, logic is about resolution theorem proving and deduction as computation. Models typically have ground terms in their domain, and attention for implementation issues (searching and backtracking, control of implementation, termination conditions) is justified in such a setup.

I am particularly excited about the chapter on PROLOG; I am not aware of any comparably thorough and still accessible approach in this field. The first five sections of this chapter are very rigid and clear, extensive proofs are provided (completeness of several forms of resolution, including lifting and independence lemmas). These rather hard theoretical results are accompanied with several examples and SLD-trees. And if one decides to give a course on logic programming, Chapters I and II seem to be suitable for putting the student in the "right logical mood." I doubt whether I would use these chapters as a first introduction to logic, though.

It is generally accepted that modal and intuitionistic logic are important for computer scientists, but I feel that in Chapters IV and V this importance is only mentioned, not really demonstrated. What kind of reasoning is offered by dynamic logic? How is epistemic logic exactly used? What are the mechanisms of constructive proof checkers and reasoning systems? How natural are proofs in a system like NUPRL?

All in all, the book seems a good buy. If not for students in computer science, it is for teachers in this and related areas (expert systems, AI), because of its composition, the thoroughness of proofs and the many valuable references for further reading.

Wiebe van der Hoek
Department of Computer Science
Utrecht University
P.O. Box 80089
3508 TB Utrecht
The Netherlands
E-mail: wiebe@cs.ruu.nl