shade below that of Lewis and Papadimitriou's ***Elements of the theory of computation*** (XLIX 989), and far below that of Hopcroft and Ullman's ***Introduction to automata theory, languages, and computation*** (Addison-Wesley, 1979). This is not a criticism since Gurari's text is aimed at undergraduates.

The author's stated motivation is "the desire to provide an approach that would be more appealing to readers with a background in programming." His main technique for this is to illustrate all models by a type of program written in a formally defined structured language. For example, it is *proved* that a function is computable by a finite-state transducer iff it is computable by a finite-memory program. He also motivates many concepts with examples from programming. The nonstandard topics should appeal to people interested in programming. Overall, he succeeds in his goal.

There are diagrams for all machines discussed, not just finite automata. In addition there are excellent pictures of machines, tracing their executions. The author includes a large (but finite) number of examples; he is not afraid to include rather complex examples. This should help many students.

Several topics are absent which *might* be missed. Parsing, Chomsky normal form, Greibach normal form, Rice's theorem, and primitive recursion are not included. It is likely that an instructor may desire some topics that are not there. In addition, no hints about harder material (e.g. two-way automata) are in the text or exercises, except for the last line of every chapter which refers to a more advanced book (often Hopcroft–Ullman).

A more serious problem is the lack of a *sense* of rigor. The theorems and proofs are fine, but the style of the book underestimates the importance of rigor. Most of the definitions are not called definitions, they are just defined in prose (the first use of the traditional boldface "**Definition**" is on p. 171). For example, the formal definition of a finite-state transducer is presented by first exhibiting a particular finite-memory program P, and then saying "The computational behavior of P can be abstracted by a formal system $\langle Q, \Sigma, \Delta, \delta, q_0, F\rangle$, which is defined through the algorithm below." Although the definition is fine, the reader does not get a sense that something has been formally defined. In addition, there is no Chapter 0 on proof techniques. This is surprising since this course is usually the first exposure to serious theory. Although the lack of a sense of rigor is a serious problem, it may be compensated for by a good teacher.

The book contains some oddities. The first chapter is a rather extensive overview of concepts that will be used throughout the text (e.g. grammars, reductions). This chapter's exercises mention the undecidability of Hilbert's tenth problem, but this does not resurface in the chapter on Turing machines where it would be better understood. Transducers are defined before acceptors because this fits in with the author's stated motivation (this is not a fault). The equivalence of finite-automata-recognizable languages, and regular-expression-generable languages is relegated to exercises. The term "finite-domain program" is used to mean a program with finite memory.

If an instructor is happy with a current textbook, then there is no compelling reason to switch to this one; however, if an instructor is looking for a new book, and will stress to students the importance of rigor, then this book may be a good choice. WILLIAM I. GASARCH

RAYMOND TURNER. ***Logics for artificial intelligence.*** Ellis Horwood series in artificial intelligence. Ellis Horwood, Chichester 1984, also distributed by Halsted Press, New York, 121 pp.

This very short book is apparently intended as a supplementary text in a graduate AI course. The author describes it as a "text and reference work on the applications of non-standard logics to artificial intelligence (AI)." It gives short and concise (too short and too concise, in the reviewers' opinion) introductions to dynamic logic, modal logic, many-valued logic, non-monotonic logic, temporal logic, type theory, and fuzzy logic. Surprisingly, it does not contain a discussion of applications of *standard* logic to AI—automated theorem-proving tools such as resolution, and uses such as answer-extraction, robotic planning, and so on. Equally surprising is omission of any discussion of identity, probabilistic logics, inductive logics, intensional logics, relevant logics, paraconsistent logics, and omission of the whole topic of representation of natural language by logic.

Although it is salutary for a computer scientist to know something about the theory of types (and perhaps even about intuitionistic logic), it *does* take some stretching of these topics to make them fall under "logics for AI"; and Turner does not do the stretching but rather introduces these logics as methods of giving semantics for programming languages. Even in areas that require no stretching at all to be central to "logics for AI," Turner sometimes seems to take a perverse pleasure in making the discussion fall outside of that topic. Modal logic is very briefly introduced (three pages for syntax and semantics), and his attention turns to the semantics of programming languages and of program execution as an interpretation of modal logic.

The book seems most useful as a guide to the literature in certain fields of logic-applied-to-AI (and applied also to other things). It would be more suitable in this role if it had broader coverage. (One might even say it is unsuitable because of this lack of breadth.) It would also be better if the discussions were carried out with AI in mind, rather than with respect to other areas of computing. It is unsuitable as an introduction to the fields themselves because of its brevity and lack of depth. One simply cannot introduce whole areas of logic in a page or three without assuming some general sophistication on the part of the reader. (Turner is quite misleading when he says on page 16 that his two-page introduction to classical logic is all the reader need know.) The presentations of the areas themselves rarely go beyond definitions of basic terms, which are themselves sometimes difficult to follow. As a textbook it is flawed by having no exercises. And although a guide to the literature is better when it is augmented by the author's reactions to the literature, sometimes Turner is just too informal for a logic book; for example, "I prefer the account of Allen's to that of McDermott if only because events are taken as primitive (which seems intuitively sound) and 'chunks of time' are utilised instead of instants" (p. 88).

We would be remiss as reviewers if we did not point out that the book is quite liberally sprinkled with typos and misprints—sometimes reducing the text and formulas to incomprehensibility. Finally, we think that the serious student of logic in AI would be better served by a selection of articles from ***Handbook of philosophical logic***, Volumes I–IV, edited by D. Gabbay and F. Guenthner, D. Reidel Publishing Company, 1983, 1984, 1986, 1989. Although this would not include all the topics of interest under the rubric of "logics for AI," it would properly include those discussed in the book under review. Furthermore, the introductions and motivation in these articles are better done, and more advanced results are given. FRANCIS JEFFRY PELLETIER and LENHART K. SCHUBERT

MICHAEL BARR and CHARLES WELLS. ***Toposes, triples and theories.*** Grundlehren der mathematischen Wissenschaften, no. 278. Springer-Verlag, New York etc. 1985, xiii + 345 pp.

Topos theory is arguably the most profound contribution that category theory has made to the development of mathematical logic and foundations. In general, category theory attempts to describe mathematical structures and their properties in terms of their transformations, or "morphisms," rather than in terms of the detailed construction of the structures from sets, relations, functions, and so on. When the mathematical structures in question are merely unstructured sets, this approach requires us to try to formulate set-theoretic concepts in terms of *functions between sets*, rather than in terms of *elements of sets*. Take for example the power-set $\mathscr{P}(X)$ of a given set $X$, which can be characterized uniquely up to bijection by the following purely function-theoretic definition: there is a one-to-one function $\varepsilon_X: E_X \to X \times \mathscr{P}(X)$ with the property that for any other one-to-one function $r: R \to X \times Y$, there exists a unique function $\chi(r): Y \to \mathscr{P}(X)$ such that the inverse image of $\varepsilon_X$ along $\mathrm{id}_X \times \chi(r): X \times Y \to X \times \mathscr{P}(X)$ is $r$. The auxiliary notions of Cartesian product ($\times$), one-to-one function, and inverse image used in the above definition can all be given category-theoretic characterizations (as categorical product, monomorphism, and pullback of monomorphism, respectively). So one can abstract these definitions from sets and functions to the objects and morphisms of an arbitrary category—to obtain the notion of the *power-object* of an object in a category possessing finite products. Quite simply, a topos is a category with finite products in which every object has a power-object.

This definition of a topos is simple—but deceptively so. The existence of finite products and power-objects ensures that the category has many special properties. Indeed, some of these properties (exponentials and finite colimits) were included in the original definition of topos given by Lawvere and Tierney in 1969 and only later were found to be derivable as the rich theory of toposes unfolded. To quote from the book under review: "Probably the best analogy elsewhere in mathematics in which a couple of mild-sounding hypotheses pick out a very narrow and interesting class of examples is the way in which the Cauchy–Riemann equations select the analytic functions from all smooth functions of a complex variable" (page 65). Just how "narrow and interesting" is the class of toposes? A good answer is provided by the precise connection that exists between a certain sort of constructive logic and topos theory. The logic in question is a brand of higher-order intuitionistic predicate calculus. (It can also be viewed as a limited form of intuitionistic set theory—the principal limitation being that quantified variables are restricted to range over the elements of sets.) Use of the impredicative power-set provided by this logic permits much (constructive) mathematics to be carried out in it, and hence to be interpreted in any topos.

So there is a method of developing many properties of toposes using logic as a tool rather than relying on a purely category-theoretic development. It is the latter approach, however, which is adopted in this book. To see how things appear using a "logical" approach, the reader is referred to J. Lambek and P. J.