chapter also contains a section on error detecting codes and A/D and D/A converters.

The concluding chapter, Computer Organization and Control, begins with a discussion of the basic content of an instruction, next considers multiple address machines and then treats the sequencing of operations. It discusses index registers, indirect addressing, floating point operations and has new material on minicomputers and paging. The author does a commendable job of comparing the techniques used by many machines. This text can be very valuable when used for students just entering the field of computer hardware or just seeking a basic understanding of the functioning parts of a digital machine. It is obviously not deeply theoretical and not appropriate for advanced students. Bartee's writing style is free enough that an instructor can delete, can supplement or use the text as is with equal ease. I believe the third edition will enjoy the same popularity as earlier editions.

ROBERT E. SWARTWOUT
West Virginia Univ.
Morgantown, W. Va.


This is a textbook in Computer Science based on the Vienna Definition Language developed by F. Lucas and his colleagues at the IBM laboratory in Vienna. Their original purpose was to produce a formal definition of PL/I, but the language and the methodology that they have developed has much wider application.

There was much interest and discussion at the time the original Vienna group's work was published, but so far at least their approach has not been very generally adopted. The papers are slow and difficult reading and the details are known only to relatively few specialists.

The author apparently feels strongly that the language and its implied methodology can provide a unifying structure for many parts of Computer Science. In his preface he talks of building a bridge between the theory and the practice of computing.

The first two parts of the book provide an informal and then a more formal description of the Vienna Definition Language, slightly extended by the author. These two parts take up about 100 pages. The rest of the book consists of applications of the method in the three areas of the subtitle: algorithms, processors, and languages. These include a formal presentation of some sorting algorithms, a formal description of the PDP-8 mini-computer, and a formal definition of the language Basic.

This is clearly an extremely useful book for those who wish to learn about the Vienna Definition Language. This reviewer does not go along with the author in his suggestion that this language and this approach can and perhaps should form the basis for the study of all of Computer Science. The author performs a service by presenting a number of carefully worked out examples of the use of this particular formalism, but there are many computer scientists who will not find this formalism attractive.

SAUL ROSEN
Purdue Univ.
W. Lafayette, Ind.


With the increasing interest in automata theory, the number of books on this subject has been steadily increasing since the early 1960's. Depending on the audience, one can now choose books in the graduate level as well as in the introductory junior-senior level.

Depending on the emphasis, one can also choose books that approach the subject from different angles. This book, Abstract Automata, by P. H. Starke of Germany is certainly a welcome addition to this collection.

The author did an excellent job in presenting systematically the fundamentals on three types of automata; the deterministic automata, the non-deterministic automata, and the stochastic automata. These may be typical in an textbook on automata theory, however, the substance in this book is by no means typical. It covers not only the classical results of these automata, but also recent results which can only be found in journals. Many of these are the author's own work on non-deterministic automata and stochastic automata. All topics have been treated here in a general way. There are infinite although special attention is given to finite automata.

This book is an outgrowth of a graduate seminar in Automata Theory given by the author at the Humboldt University, Germany. As such, it involves rather detailed studies on some specific problems. Hence, it is not suited as a text for an introductory course in automata theory. This book contains no exercise, although unproved corollaries could be used. It is perhaps suitable as a text or as a reference in a seminar course for relatively mature audiences. It certainly contains plenty of materials to choose from.

The author has organized the book in such a way that all three types of automata are discussed along a parallel path. This allows the reader to compare the similarity and differences of results.

The book begins with an introduction in which the motivation and the mathematical formulation of three types of automata are described. Included in the Introduction is a short overview of the problems covered in this book. Then there is a short chapter on preliminaries where the author introduces some basic definitions and notations. The reviewer advises a careful reading of this chapter since the predicate calculus notation employed in this book is somewhat different from the usual convention.

The book is divided into three parts, one for each type of automata: Part I is on Deterministic Automata; Part II is on Nondeterministic Automata; and Part III is on Stochastic Automata.

In Part I, the basic notion of a deterministic (abstract) automaton is given in Section 1. The Moore and Mealy models are then introduced. Section 2 discusses the behavior equivalence and the related minimization problem. The equivalence between the Mealy and the Moore model is shown. In Section 3, the uniqueness (up to isomorphism) of the reduced of an automaton is established. It is noted that since the reduct of a Moore automaton is not necessarily a Moore automaton again, the property of being a Moore automaton is not an invariant under automaton homomorphisms.

The input/output relation realized by deterministic automata is discussed in Section 4. A word function \( f \) is a mapping from the set of input words \( W(X) \) into the set of output words \( W(Y) \). The function \( f \) is extendable \([1] \) if \( f(x) = f(y) \) for some \( r \) in \( W(Y) \). The necessary and sufficient condition for a length preserving word function \( f \) to be generated by a state of some automaton is that \( f \) be extendable.

An event \( E \) over a set \( X \) is any nonempty subset of \( W(X) \). Events that are representable by automata are discussed in Section 5. The notion of an event being represented by an output signal in an automaton is introduced first. It is shown that every event not containing the null word \( e \) is so representable. To take care of events containing the empty word \( e \), the author defines the representability of an event by a set of states or a set of transitions. This two definitions of representability, one external and one internal, are indeed equivalent up to \( \{e\} \). Representability of events by finite automata by means of a set of final states is discussed in Section 6. The author first presents the results by Myhill and Nerode characterizing regular events by right congruence relations on \( W(X) \) of finite rank (index). Then Kleene's work on regular expression is presented.

To decide whether two regular expressions represent the same set, Saloorn's result on complete axiom system (infinite) for regular expression is presented in Section 7. The axiom system can be made finite, if the rules of inference are modified.

Section 8 investigates the properties of stable and definite automata. An automaton is stable if \( X^2 \), such that the output at any time depends only on the last \( k \) inputs. An automaton is definite if its transition function possesses this property. Every definite automaton is stable and, furthermore, a reduced automaton is stable if it is definite. The author then established the relationship between definite events and events representable by noninitial automata.