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 treaty 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 P. 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 nondeterministic 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 nondeterministic automata and stochastic automata. All results are treated here. In general, 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 difference 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 reduc 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 \( \phi \) is a mapping from the set of input words \( W(X) \) into the set of output words \( W(Y) \). The function \( \phi \) is extendable \([1]\) if \( \phi(pq) = \phi(p) \phi(r) \) for some \( r \in W(Y) \). The necessary and sufficient condition for a long preservation word function \( \phi \) to be generated by a state of some automaton is that \( \phi \) 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 in Section 6. 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 finite states is discussed in Section 6. The author presents the results by Myhill and Nerode characterizing regular events by right congruence relations on \( (W(X) \) of finite rank (in Section 7). Then Kleene's work on regular expression is presented.

To decide whether two regular expressions represent the same set, Salom'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 it only outputs 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.
Various identification experiments on automata are discussed in Section 9. Author’s work in this area as well as works by Ginsburg, Hibbard, and Moore are presented.

Incompletely specified automata, called partial automata, are treated in Section 10. The notion of compatibility between states is introduced, along with the useful concept of a generalized decomposition (covering with substitution property), the author investigates the minimization problem for partial automata. An algorithm is given to find all automata coverings containing compatibility classes for any given finite partial automaton.

Part I concludes with an important section (Section 11) dealing with the generalized decomposition for finite automata. The author introduces three types of connections between automata: series, parallel, and feedback. If a connection of component automata realizes a given automaton \( B \), then it is called a decomposition of \( B \). The notion of feedback decomposition is similar to that of generalized decomposition by Pu [2]. Most of the results presented here are the works of Hartmanis-Stearns, Smolensky, and Yoeli on loopfree (series-parallel) decompositions for finite automata. The works by Krohn and Rhodes are not included. The feedback decomposition is not treated in depth. Interested readers can read the work on generalized decomposition by Pu [3].

Several areas in the theory of deterministic automata such as algebraic theory of automata, automata with various structural inputs, automata and formal languages are not covered in this book. The author purposely left these out, since no parallel results are yet known for the nondeterministic and stochastic cases.

Part II is devoted to nondeterministic (ND) automata. The fundamental notion of ND-automaton is discussed in Section 2. In the rest of Section 2, works by Evan, Carlyle, and Bacon on a reduction, minimality, minimum, etc., are presented. Homomorphism and strong homomorphism are discussed in Section 3. Also investigated are the properties preserved under both types of homomorphisms. The results are similar to Section 4 for the nondeterministic case.

Section 4 discusses the input-output relation generated by a state of a \( S \)-automaton. Similar to the word functions in the deterministic case and the ND-operatrs in the nondeterministic case, stochastic operators (S-operators) are introduced here. A S-operator is a mapping taking any input word \( pW(X) \) into a discrete PM over \( W(Y) \). The necessary and sufficient condition for a S-operator \( \Phi \) to be generable is to be sequential. The results of Künzler, characterizing regular pseudo-stochastic operators, is then used to characterize S-operators which can be generated in a finite state \( S \)-automaton. The presentation in Section 4 is similar to that for the ND case.

The definition of homogeneity is extended to \( S \)-automata in Section 5 under conditional probability. It is also shown that this class contains all Z-deterministic \( S \)-automata. S-operators which are generable in homogeneous \( S \)-automata are studied. Most of the results are similar to Section 5 for the ND case.

In Section 6, two operators are defined on any given S-automaton \( B \); the closure \( \bar{B} \) of \( B \), and the realizable closure \( \bar{B} \) of \( B \). Properties which are invariant under these operations are studied. Also discussed are the connections between \( B, \bar{B}, B, \bar{B}, \bar{B}, \bar{B} \). The results in this section are similar to Section 6 of Part I. Strong and terminal equivalence of \( S \)-automata are discussed in Section 7. Several results similar to those for ND automata are established.

In the last two sections, 8 and 9 of Part III, events which are representable by \( S \)-automata are studied. Two approaches are taken; author’s and that of Rabin’s. The final section concludes with some remarks about definite stochastic events and the corresponding sets of words.

Throughout the book, the author purposefully provided rather complete and detailed proofs. In some places, he also gives intuitive explanation and interpretation to definitions and results. The reviewer feels that perhaps there should be more of these intuitive motivations—especially in the more advanced sections on ND-automata and \( S \)-automata.

As the book is the outgrowth of a seminar with mature audiences, the author employs a great amount of notation from predicate calculus. This abbreviates proofs and avoids ambiguity. However, predicate calculus notation does not provide feeling and too much of that may make the reader lose sight of the substance. There is a minor comment about notation: i.e., while the transition and output functions of an automaton are both denoted by small Greek letters, yet homomorphisms between automata are also denoted by small
Greek letters. The use of the same type of letters for functions of different level make simple concepts hard to grasp.

The reviewer at times finds it most frustrating trying to parse some of the sentences. Some of them have such a complex structure that the reviewer just had to give it up. Perhaps this is unavoidable since this book is a translation from a German text. There are several broken sentences and, of course, some typographical errors. But they are mostly harmless.

In conclusion, the reviewer is pleased to introduce this book as one to be used in a graduate seminar in automata theory and as a reference book for research purposes. The bibliography includes a rather exhaustive list of research works in automata theory published up until about 1969. This can be very useful for people interested in doing research in this area.

REFERENCE


ARTHUR T. POE
Temple University

* Also known as A. T. Pu.


This is the first book to cover the topic of interactive computer graphics. I found the book to be both very useful and very disappointing.


The authors suggest that an eight-week undergraduate course could be built around the introduction and first two parts; adding the third section could lead to a more substantial undergraduate program, and a full year’s graduate course could be built on the full book.

It is not terribly clear what the prerequisites should be. The first section on display devices is not written carefully. An unwary student would need help to stay out of the traps that the poor definitions and explanations present. My reading of the material describing how cathode ray tubes and storage tubes worked just served to make me alternately frustrated and angry, for example, the total cathode description follows.

1.3.1 CATHODE

The cathode is a small metal cylinder which is heated by an enclosed filament to a temperature at which it emits electrons.

1.3.2 CONTROL GRID

Do the authors wish the students to infer that the flow of electrons is a step function of temperature? From a metal surface?

The bibliographic listings and references served to heighten both reactions. I now realize full well this is primarily a book on programming with a computer scientist’s view of computer graphics.

It is not necessary that a programmer understand the physical operation of displays or display devices and this book does nothing to help gain that understanding. For that reason, I feel that the first section could well be deleted, greatly abridged, or rewritten to do justice to its material. It seems to me that any serious physical science or engineering student who would take this course should know more about the topic of CRT’s than this section offers. Others who would take the course deserve a better introduction, and a much better bibliography. Certainly the text by Kazan and Knoll on “Electronic Image Storage” should at least be offered to serious students as a source of more reliable information on storage tubes.

Fortunately, the book gets better as it begins to concern itself with programming-like material. Even so it tends to talk in terms that have never really been defined. Early on it offers examples in a low level language. A paper clip at the appropriate appendix will ease the reader’s trouble in trying to translate the language used.

In this early section much is made of the use of a push down stack, a stack pointer, the use of instruction “push,” “jump,” “popjump,” “push-jump.” These are used often enough that one must either figure it all out or put the book aside, but the book itself really doesn’t either define or explain in any one paragraph what each term is or what it means.

The above criticisms make me believe it would be difficult to teach the undergraduate courses without a great deal of supplementary material to bring everything to a focus.

Part Two: Display Files begins to be more informative and more readable.

Part Three: Interactive Graphics is well done and fun to read. I enjoyed the book from here on. For example, Chapter 9, “Graphic Input Devices,” is well written. The descriptions of the various input devices from the “SRI Mouse” to the Sylvania Tablet and various light pens were very readable.

I must admit my joy somewhat diminished at the beginning of Part Five—“Graphics Systems.” But there is, I’m sure, an audience that will find the treatment here more attuned to their style than to mine.

This book is clearly uneven in its treatment. For one who must learn the topic to be able to use interactive computer graphics in his work, the task to learn what is needed from this book does not appear easy.

The difficulty lies about equally in the material as presented in the earlier chapters and the lack of good up-to-date references in several areas. The bibliography, for example, offers a few papers on display quality which range from 1942-1952 and one from 1968 that is not pertinent. I believe the student deserves better.

The book is a first. If you are a skillful programmer, and want to learn interactive graphics, this book is for you.

LUCIEN M. BIBERMAN
Institute for Defense Analyses
Arlington, Va. 22202

(Maidenhead, England: Infotech Information, Ltd., 1972, 563 pp., individual volume—$95.00, subscriptions available for the 8 volumes of series 1 and of series 2 to be issued at $590.00 per series.)

This is another in a series of state-of-the-art reports from Infotech. It consists of edited discussions and presentations plus invited papers. Because a volume of this price will most likely not be sent out on approval, I think it important to review briefly the contents in detail before commenting on their value.

There are three main sections: analysis, presentations, and invited papers. There is also an annotated bibliography and indices, both to this volume and to others in the series.

The analysis section consists of eight chapters ranging in level from “The Theory of Operating Systems” to “Job Control Language.” In fact, these chapters are merely bits and pieces of transcribed discussions interspersed with editorial comments by the volume editor. They are intended to serve as introduction and pointers to the more detailed information contained in the rest of the volume.

The presentations section consists of the following papers:

“Operating Systems: What Should Today’s User Expect?” by