A. SWITCHING AND AUTOMATA THEORY


Quite frequently, a subject of considerable academic interest will emerge with a flow of textbooks that range from very narrow to very broad in scope. Not quite so frequently, a well-organized and comprehensive textbook is written—Booth has written such a book.

Chapter I is a brief introduction to the terminology, notation, and types of problems in sequential machines and automata theory. Although the state-variable notation is useful, it is also rather cumbersome since both subscripts and superscripts (with parentheses) are required in each writing of a single state-variable.

The fundamental concepts of modern algebra are summarized in Chapter II. This chapter includes such topics as sets, operations on elements, mathematical systems, semigroups, groups, rings, finite fields, isomorphisms, and homomorphisms.

An appropriate introduction to sequential machines is given in Chapter III. In the opinion of this reviewer, this chapter should contain a good treatment of the state assignment problem to accomplish both a thorough presentation of sequential machines and a timely preparation for the decomposition of sequential machines presented in Chapter IV. An average reader should find that the types of decomposition are presented quite clearly, even though the mechanics necessary to accomplish the decomposition of a given machine are rather obscure. A discussion of "self-dependent" and "cross-dependent" sets of functions would help to prepare the reader for "loop-free" and "feedback" representations of composite machines.

Chapter V is a thorough coverage of the control and identification of sequential machines. Particular attention is given to terminal-state and initial-state identification problems.

In Chapter VI, the author considers the problem of machine specification through the use of regular expressions. He states that his goal is to develop a method to generate a state table for a sequential machine from a description of the machine's required external characteristics. His presentation bears him out; he does not delve very deeply into any theory or any uses of regular expressions unless the topic leads toward the formation of a state table. In particular, he avoids the use of nondeterministic machine models which often appear in the literature. The state-table representation of a sequential machine is desired because this is the type of machine specification from which machine behavior is analyzed in previous chapters.

The presentation is begun with a discussion of 1) the relationships between the states of a machine and input sequences to the machine, and 2) input-output relationships of the machine, in order to give motivation and a basis for using regular expressions for machine description. After this is accomplished, the definition of a regular expression is given, and then the problem of arriving at a regular expression for a machine from a word definition of its required external characteristics is illustrated. With the regular expression for a machine available, the concept of the derivative of a regular expression if developed, and from this the state table for a sequential machine is derived.

Under the heading "similar regular expressions," the author gives the criterion for terminating the calculation of distinct derivatives. He states

We continue this process (taking sequences of length 1, length 2, and so on and calculating derivatives with respect to these sequences) until we reach a point at which all sequences of length r give us derivatives similar to derivatives formed for values of r with length less than r. We terminate our process at this point because, for any sequence t such that the length of \( r^* \), \( D^r_i \) would be similar to some \( D^r_0 \) where the length of \( s \) would be less than r.

While the above criterion is not incorrect, it is not a minimum requirement and could result in the calculation of the same derivatives repeatedly. One method of stating a more restrictive requirement that would cause less redundant calculation to be performed is

if for a sequence s of length r a derivative similar to a derivative formed from a sequence of length i is obtained, we do not need to calculate the derivative for any sequence s of length greater than r, starting with r, since it will not result in a dissimilar derivative.

The author then presents an algorithm for constructing a state table for a Mealy machine and a Moore machine from regular expressions. A detailed example is presented where the states, next-state mappings, and output mappings are given for both a Mealy model and a Moore model.

Next, the author presents an example of a multiple-output machine which is an extension of the previous method. It is clear in the examples that the criterion used to terminate derivatives is not the one presented in the book but is similar to the new criterion presented here.

Often pointed out in the chapter is the important result that it is possible to construct a finite-state machine to perform a given set of operations if and only if the set of operations can be represented by a set of regular expressions; if the operations cannot be represented by regular expressions, then a finite-state machine cannot be constructed to perform the operations.

In Chapter VII, the concepts of vector spaces, linear transforms, and matrices are reviewed for the purpose of developing the analytical techniques needed to deal with linear sequential machines. A vector space and the basis of a vector space are defined using the notational conventions of abstract algebra. The concepts of subspaces and vector coordinates are presented along with illustrative examples. The section dealing with combinations might have been better presented, in the interest of continuity, after the section on the coordinates of a vector rather than immediately after the definition of a vector space.

The relationship between two sets of basis vectors \( v_i \) and \( u_i \), \( i = 1, 2, \ldots, n \), for a vector space is given as
REVIEWS OF BOOKS AND PAPERS IN THE COMPUTER FIELD

v_1 = m_11u_1 + m_12u_2 + \cdots + m_{1n}u_n \\
v_2 = m_21u_1 + m_22u_2 + \cdots + m_{2n}u_n \\
\vdots \\
v_n = m_{n1}u_1 + m_{n2}u_2 + \cdots + m_{nn}u_n \\

or 

[v_i] = [m_{ij}][u_j].

It is then stated without qualification that \([m_{ij}]\) is nonsingular. It might have been mentioned that this followed from the linear independence of the elements of a set of basis vectors. It is inferred that, having chosen two sets of basis vectors, the elements of the matrix \(M\) may be obtained from the solution of the above set of linear equations.

It is then shown that if the coordinates of a vector \(v\) are \(c_i\) with respect to a basis set \(v_i\), \(i = 1, 2, \cdots, n\), the vector may be written as 

\[ v = [c_i][v_i]. \]

It is then said to be seen that the coordinates \((r_j)\) or \(v\) with respect to the basis vectors \(\{u_j\}\), \(i = 1, 2, \cdots, n\) are given by the equation 

\[ [r_j] = [c_i][u_j]. \]

This might have been more immediately apparent if the equation 

\[ v = [r_i][u_i] \]

had been included. The development would then have been 

\[ v = [c_i][v_i] \]
\[ v_i = [m_{ij}][u_i] \]
\[ v = [c_i][m_{ij}][u_j] \]
\[ v = [r_j][u_j] \]

Linear sequential machines are treated in Chapter VIII following mathematical formulation in Chapters II and VII. There are good arguments for connecting this material with the chapter on regular expressions, and for this reason additional sections concerning the synthesis of linear (or polynomial) machines from regular expressions could be included. The material concerning the relationships between the rank of the \(K\) matrix and minimal state machines is quite simple for those readers having a basic knowledge of matrix theory; however, the rules applied are not explained, at least not explicitly, in the referenced chapter.

The first portion of Chapter IX gives a "mechanical" description of a Turing machine, showing the basic operation, and then goes into a formal mathematical definition. The next two sections present algorithmic approaches to accomplish large computational problems. Several very elementary submachines are first given and a method is given to accomplish the necessary interconnect for larger submachines that may be used to perform basic information-processing operations. Then, machines are formed to perform such operations as addition, subtraction, and multiplication. It is also shown that composition, primitive recursion, and minimalization, which can be performed by such machines, can be used to generate a large variety of more complex functions. The next two sections are devoted to a detailed treatment of compatibility. The last portion of the chapter is devoted to special Turing machines, with multiple tapes for separate storage and processing of information, or with special rules governing how information can be taken from and put into the external memory.

In Chapter X, regular expression is again a topic, but is now viewed from a different (and broader) perspective. The discussion of artificial languages in just one chapter must, of course, be limited. Chapter X deals with phrase-structure languages because they are closely related to the systems covered in preceding chapters. The chapter is divided into four sections: 1) definitions and terminology, 2) language-machine relationships, 3) operations on languages, and 4) decision problems. The first two sections establish what an artificial language is and how a language is associated with a particular machine. The last two sections extend further the mathematical basis of artificial languages.

Chapters XI and XII are devoted to random sequences and random processes in sequential machines.

A substantial list of references is included at the end of each chapter and challenging problems (with solutions in Appendix IV) are given at the end of each section. There are also problems given at the end of most of the chapters to point out the salient fundamentals included in the chapters.

The material in the book is organized so that it can be used for a single course, or for a two-course sequence without loss of contingency. It is, however, unfortunate that so many printing errors were made in the first printing.

The author has, in the opinion of this reviewer, accomplished a very commendable contribution.

Chester C. Carroll
Dept. of Elec. Engrg.
Auburn University
Auburn, Ala. 36830

B. EXECUTIVE PROGRAM DEVELOPMENT


Innumerable papers by Russian authors using probabilistic models for physical systems are prefaced by a comment explaining that the results obtained must follow from dialectic materialism, and therefore the same results could in principle be derived deterministically as well as statistically, save for mathematical convenience. A similarly political motivation may unconsciously underlie many papers in the computer field which explain that ever more elaborate executive programs are motivated exclusively by a desire to increase hardware efficiency and improve system economy.

Thus this paper, an otherwise excellent review of the array of problems encountered in organizing an executive program, weakens its semi-historical development by presenting each elaboration of executive responsibility as motivated only by improved hardware utilization. In the most serious example of the distortion caused by this viewpoint, the paper suggests that the motivation for paging as a scheme for allocating primary memory is to sandwich additional programs into memory in the holes left between other programs. Whether or not paging increases or decreases the number of words of primary memory which can be utilized probably depends on the statistics of the programs involved. The more serious issues (and the ones which raise the most emotional arguments) are the abilities of a paged machine to present the programmer with an interface which does not change every time a new box of memory is added, and to relieve the programmer of explicit overlay organization. The author mentions these latter issues as though they were virtually side effects of the effort to increase hardware utilization.

Aside from this single significant complaint, this paper is a welcome relief from the long series of "here is how we implemented a multi — system on the ——— computer." The author has taken the trouble to identify many of the sources of difficulty and complexity in an executive program. This is one of the few papers one can point to which recognize the importance of error handling and program snapshot/restart procedures in their contribution to the complexity of executive programs.

A discussion of solutions as comprehensive as the author's discussion of problems would require at least a book. The author has taken what appeared to be an easy shortening scheme — he indicates one or two techniques of attacking each problem. Unfortunately, he becomes unintentionally dogmatic when he presents the technique he is most familiar with and fails to indicate that there may exist several other techniques to solve the same problem. This comment applies, for example, to his discussions of priority schemes, error handling, and interrupt handling.