Coordinated Science Laboratory

UNIVERSITY OF ILLINOIS – URBANA, ILLINOIS

AUTOMATA & SEQUENTIAL MACHINES, A SURVEY

Herbert Y. Chang

REPORT R-168 JUNE, 1963

COORDINATED SCIENCE LABORATORY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

Contract DA-36-039-TR US AMC 02208(E)
DA Project 3A-99-25-004

The research reported in this document was made possible by support extended to the University of Illinois, Coordinated Science Laboratory, jointly by the Department of the Army, Department of the Navy (Office of Naval Research), and the Department of the Air Force (Office of Scientific Research) under Department of Army Contract DA-36-039-TR US AMC 02208(E).

TABLE OF CONTENTS

1. INTRODUCTION

Automata theory is becoming the core of many modern physical sciences. The theory of switching and sequential circuits is merely a branch of this new field of science. In spite of the rapid growth of new concepts and ideas of automata theory in recent years, only a limited number of applications is seen in computer design practice. A partial reason may be due to an engineer's unfamiliarity with the abstract notions being employed in the literature of automata theory. An attempt is made therefore to study the important contributions of this new field of science to the practical problems encountered in the area of sequential machine theory. The investigation concerning design and analysis is taken from an engineer's viewpoint with emphasis on the established theoretical results and possible routes for future development. Familiarity with notions of Huffman-Moore-Mealy is assumed. In the first section, various models of sequential machines are defined. Methods of characterizing machines other than the usual flow table or state diagram approach are pointed out in the second section. Finally, a number of results and problems concerning analysis and synthesis are mentioned.

2. THE MODELS, DEFINITIONS AND NOTATIONS

Among various classes of automata - fixed and growing, discrete and continuous, synchronous and non-synchronous, deterministic and probabilistic, and finite-state and infinite-state[75] - models of sequential machines are usually classified as the class of fixed-discrete synchrouous deterministic finite-state automata, or simply finite automata in short. In this section we will introduce some of the common models that are discussed in the literature.

2.1 Moore's Model

E.F. Moore[79] was among one of the first to investigate the abstract properties of sequential circuits. According to his notion, a sequential machine is a machine which has a finite number of states, a finite number of possible input symbols, and a finite number of possible output symbols. The behavior of the machine is strictly deterministic in that the present state of the machine depends only on its previous input and previous state, and the present output depends only on the present state.

If we denote by $\sum$ the set of possible inputs whose elements are I_1, $I_2, \ldots, I_m$, $\triangle$ as the set of possible outputs whose elements are $Z_1, Z_2, \ldots, Z_p$; and Y as the set of (internal) states whose elements are $q_1, q_2, \ldots, q_n$; where n,m, and p are finite; and furthermore, if we call δ the next state function and λ the output function, then the description of Moore's model can be transformed into the mathematical expressions:

$$\begin{aligned} & q_j = \delta\,(I_i,\ q_i) \\ & Z_j = \lambda\,(q_j) \\ & q_i,\ q_j \in Y,\ I_i \in \sum \\ & Z_i \in \triangle \end{aligned} \qquad (2\text{-}1)$$

where subscript i denotes previous moment and j denotes present moment on the time scale. If δ is defined for all possible pairs of (I, q); $I \in \sum$, $q \in Y$, then we say the machine is complete then we say the machine is complete[*]; otherwise, the machine is called incomplete[*]. In other words, output and next state of an incomplete machine are not defined for every input.

Since the output depends only upon the state and not upon the input, $\lambda(I_1, q) = \lambda(I_2, q)$, for $q \in Y$, and any $I_1, I_2 \in \sum$, this model is also called input-independent machine.[**] We must not, however, be confused by the name "input-independent" and fail to recognize the fact that the next state does depend on the input, so that a sequence of outputs $\lambda(J, q)$ does depend on the input sequence J whenever J has a length greater than 1.

2.2 Mealy's Model

Mealy's[76] definition of sequential machine[†] is contained in the definition of what he called a switching circuit.

A switching circuit is a circuit with a finite number of inputs, outputs, and (internal) states. Its present output combination and next state are determined uniquely by the present input combination and the present state. If the circuit has one internal state, we call it a combinational circuit, otherwise, we call it a sequential circuit (machine).

Again, the definition can be characterized by the equations:

$$\begin{aligned} q_k &= \delta(I_j, q_j) \\ Z_j &= \lambda(I_j, q_j) \end{aligned} \qquad (2\text{-}2)$$

where subscripts j and k denote present moment and next moment, respectively, on time scale.

Moore's and Mealy's model have been shown by Cadden[22] to be equivalent in the sense that every system describable by one model is shown to be describable by the other also. More discussions on the behavior equivalence of these

[*] According to Huffman's terminology, the terms "complete" and "incomplete" are equivalent to the terms "completely specified" and "incompletely specified", respectively.

[†] Mealy actually used the word "circuit", the difference between a sequential machine and a sequential circuit does not seem to be in existence, from a mathematical point of view, see S. Seshu[96]

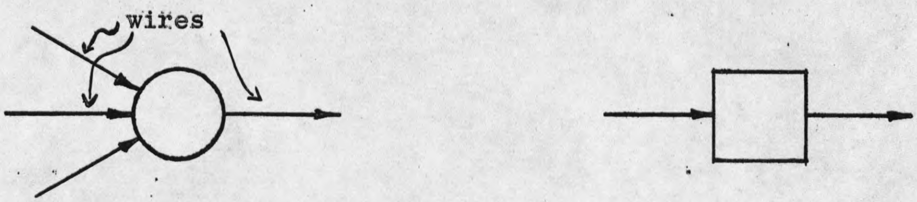
[**] See Section 2.2 for reference.

two models can be found in papers by S. Seshu[96] and A. Gill.[34]

In the field of automata studies, there has not been a unique definition of automaton that every writer agrees upon. The complexity of the problem often forces the writer to define new terms, and give new names on various occasions to suit his special purpose. It is, therefore, rather difficult to correlate all models of automata with Moore-Mealy's notion. We will only mention a few models which have the closest similarities related to sequential machines.

2.3 Finite-State Model-I :

A. Burks and H. Wang[19] discuss a model of finite automata which appears to be general enough to represent any other models of its kind. Before we introduce their model, we need a few definitions. By junctions we mean the ends of wires which do not impinge on a switching-element circle or a delay-element rectangular, (Figure 1).



Switching Element

Delay Element

Figure 1. Switching Element and Delay Element

A junction with no output wires attached to it is called an input junction. All others are called internal junctions.

An (finite) automaton is a fixed finite structure with a fixed number of input junctions and a fixed number of internal junctions such that: (a) each junction is capable of having two states, (b) the states of input junctions at every moment are arbitrary, (c) the states of the internal junctions at time zero are distinguished, (d) the internal state (i.e., the states of the internal junctions) at time t+1 is completely determined by the states of all junctions at time t and input junctions at time t+1 according to some pre-assigned law (which is embodied in the given structure).

Or in other words, this model can be characterized by two arbitrary effective transformations τ and λ from pairs of integers to integers. These integers are drawn from finite sets $\{I\}$, $\{S\}$, and $\{O\}$. Set $\{S\}$ contains a distinguished integer $\mathscr{S}_o$. Transformations are given:

$$\begin{aligned} S(0) &= \mathscr{S}_o \\ S(t+1) &= \tau[I(t), S(t)] \\ O(t) &= \lambda[I(t), S(t)] \end{aligned} \tag{2.3}$$

Apart from the fact that a particular initial state is specified, this model is essentially the same as Mealy's.

2.4 Finite-State Model-II

Closely related to Moore's model is the class of two-symbol finite automata defined by C.Y. Lee.[67]

A two-symbol finite automaton consists of

(a) A finite number of internal states, $q_o, q_1, \ldots, q_n$

(b) An alphabet of two symbols: $S_o=0$, $S_1=1$

(c) A map M whose domain and range are both subsets of the set of state-symbol pairs. If M is defined over state-symbol pair (q_i, S_j), then $M(q_i, S_j)$ is another pair (q_k, S_r). The symbol S_r is called an output symbol, and is completely determined by q_i; that is, S_r is independent of the input symbol S_j.

(d) An initial state q_o, which can reach every state q_i, $0 < i \leq n$ via some suitable input sequence of symbols.

This class of finite automata, as pointed out by Lee, can be regarded as a subfamily of Turing machines[102] and also a subfamily of W-machines.[103]

2.5 Non-Printing Tape Machine[89]

Intuitively, a non-printing tape machine may be regarded as a black box with a reading head and a one-dimensional tape. A yes-or-no question, interpreted as any arbitrary finite sequence of symbols from a finite alphabet, is represented by a tape. The reading head then reads the tape one symbol at a time. After each reading, it advances the tape one unit and reads the next symbol. The process stops when it runs out of tape. At this point, the machine answers the question by indicating a YES or a NO.

Now we are ready to give a formal definition of this model.

A <u>Non-Printing Tape Machine</u> over the alphabet E is a system $U=(S,M,s_o,F)$ where S is a finite non-empty set (the <u>internal</u> states of U), M is a function defined on the cartesian product S x E of all pairs of states and symbols with values in S (the table of transitions or "moves" of U), s_o is an element of S (the <u>initial</u> state of U), and F is a subset of S (the designated <u>final</u> states of U).

If Λ is denoted as the empty tape with no symbols; T as the class of all tapes, then we have the relations:

$$M(s,\Lambda) = s \text{ for } s \in S$$

$$M(s,x\sigma) = M(M(s,x),\sigma),\ \text{for } s \in S,\ x \in T, \text{ and } \sigma \in E \qquad (2\text{-}4)$$

$$M(s,x\,y) = M(M(s,x),y),\ \text{for } s \in S,\ x,y \in T$$

In addition to models mentioned above, there are still many other models whose structure and behavior bear close relations to that of Moore and Mealy models. Nerve nets[66],[72] and W-machines[68] (printing automata) are merely two of the many examples. A few abstract models are also discussed by Ginsburg in his papers.[47],[51]

To avoid ambiguity, we shall consider a sequential machine or simply a <u>machine</u> to be Mealy's model, unless otherwise specified. The terms <u>finite automaton</u> and machine are taken to be interchangable. In other words, a <u>machine</u> is characterized by the expressions:

$$\begin{aligned} q_k &= S(I_j,q_j) \\ Z_j &= \lambda(I_j,q_j) \end{aligned} \qquad (2\text{-}5)$$

which is the same as (1-2). We shall denote a machine S with n (internal states, m inputs, and p outputs as $S_{(n,m,p)}$. A machine S is said to be <u>strongly connected</u>[79] if for any ordered pair (q_i,q_j) of states of S, there exists a sequence of inputs which will take the machine from state q_i to state q_j.

3. CHARACTERIZATION OF MACHINES

The most important and most difficult part of synthesis of a sequential machine is the problem of stating exactly what this machine should do. The complexity of synthesis procedure depends, to a great extent, on the precision and simplicity of how the problem is specified. In this section, we intend to discuss some of the topics on methods to characterize machine actions that are related to concepts of automata theory. We will not, however, discuss the use of conventional "flow table" or a combination of state diagram and truth-table descriptions since these techniques are commonly known in practice. Interested readers may refer to papers by Huffman(61), Moore(79), and Mealy.(76)

3.1 Input-Output Signal Set Method

Let us think of a sequential machine as a black box with a finite number of input terminals and a finite number of output terminals (Figure 2).

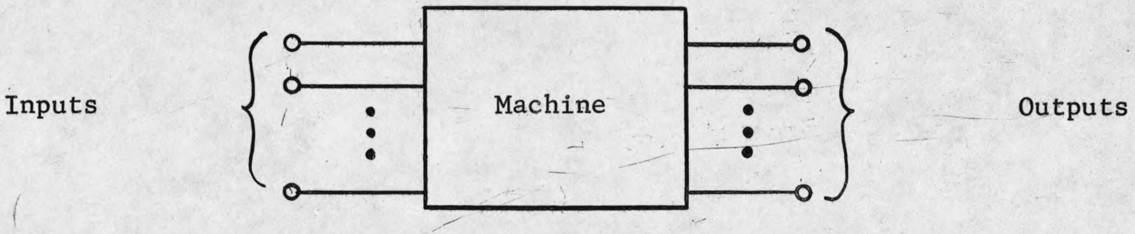


Figure 2. A 'Black Box' Machine

The black box is required to do certain things whenever a combination of input signals is present at input terminals so that a certain configuration of output signals will appear at output terminals. In other words, the black box acts like an input-output transducer subject to some given operating conditions. A natural way to characterize machine's action is therefore to specify the desired input-output configurations at these terminals. Machine's operation is synchronous. Signals are observed only at discrete moments of time, thus they can be applied in sequences according to some arbitrary time scale. We shall call this kind of input-output sequence characterization the input-output signal set[67] specification, or in more familiar words, the regular expression language[73] [13] specification.

A question arises here: What properties should a signal set possess in order to characterize a machine? The concepts of representation and realization of input-output signal sets are discussed by Kleene[66]; Copi, Elgot and Wright[23]; Rabin and Scott[89]; Lee[67]; Arden[4]; McNaughton and Yamada[73]; and many others (see bibliography).

Without loss of generality, let us represent the time scale by integers 0,1,2,...,t with 0 denoting the origin and t the present moment. At any time t, every signal at a terminal is assumed to take one of the two binary values, 0 or 1. An input (or output) configuration then becomes an ordered n-tuple of 0's and 1's. Along the time scale, sequences of input and output ordered n-tuples are called input sequence and output sequence, respectively. Among sets of input sequences, in order to distinguish one set that is applicable to a machine from those that are not, the notion of event is introduced.[66] An event is a subset of the set of all input sequences. It occurs when the actual input sequence belongs to this subset. An event E is said to be realized by a machine if and only if the signal at output terminal is 1 at the time of application of the last symbol of any input sequence denoted by E. Some events are realizable with combinational circuits but some others require circuits with memory (sequential machines).

Combinational circuits are often referred to as definite automata in the terminology of automata theory. More precisely, a definite automaton is a machine whose output at any time t is uniquely determined by the direct inputs at times $t-\ell+1$, $t-\ell$, ..., $t-1$, t (ℓ is finite). An event is definite if it is characterized by finite sets of sequences whose occurrences depend on the last $\ell(\geq 1)$ moments of time, $t-\ell+1$, $t-\ell$, ..., t. The realizability of definite events is discussed in papers by Kleene[66]; Copi, Elgot and Wright[23] with instantaneous logic, and by Arden[4] with both instantaneous logic and delay logic. The important result is that every definite event is realizable by a definite automaton using logical devices consisting of AND-gate, OR-gate, inverters and delays only, and every definite automaton represents a definite event.

A number of topics concerning definite automata and definite events have been discussed by Perles, Rabin and Shamir[88]; and by Brazozowski[13].

Events that characterize finite automata (or sequential machines) are so-called regular events. A few preliminary definitions are needed before we discuss the regularity of events. By concatenation of two events S and T, written as $S \circ T$, we mean the set of all sequences formed by a member of S followed by a member of T. Iterate of S and T, written as S^*T, means the smallest set of all sequences of S, $S \circ T$, $S \circ S \circ T$, $S \circ S \circ S \circ T$,..., etc. If T is empty, then S^*T, which can be written as S^*, means the smallest set of sequences of ϕ (empty set), S, $S \circ S$, $S \circ S \circ S$,...etc. Finally, the union of S and T, written as $S \vee T$, means the set of all sequences consisting of members of S and T.

Regular events are defined recursively as follows: (1) every event consisting of finite set of sequences is regular, (2) the concatenation of two regular events is regular, (3) the iterate of a regular event on a regular event is regular, (4) the union of two regular events is regular, (5) no event is regular unless its being so follows from (1), (2), (3), (4). It is apparent that definite events are regular, but regular events are not necessarily definite in general.

If we denote symbols in sequences by alphabet $\{0,1\}$, and let ϕ be the unit set of null sequences, i.e., sequence of zero length, and let Λ be the empty set of sequences, regular events then can be described by a language called regular expressions. Regular expression is defined recursively as follows: (1) a string consisting of a single 0, a single 1, a single ϕ or a single Λ is a regular expression, (2) if P and Q are regular expressions, then so are $P \vee Q$, $P \circ Q$ and P^*, (3) no other string of symbols is a regular expression unless its being so followed from (1) and (2). Thus an event is regular if and only if it can be described by a regular expression. Here we give an example showing how regular expression specifies an input signal set. Suppose the word description for a one-input and one-output machine is given as: "The output is 1 if and only if the binary value: of input sequence is equal to 2^n+1 ($n \geq 1$) since the last output 1 (disregarding the output at t=0)." In regular expression language, this specification is equivalent to $=[1(0)^*1,]$, which is the set consisting of sequences 11, 101, 1001, 10001,...,etc. A sample of this signal set is:

Input:	1	0	1	1	0	0	1	1	1	1	0	0	...
Output:	0	0	1	0	0	0	1	0	1	0	0	0	...
t =	0	1	2	3	4	5	6	7	8	9	10	11	---

Kleene indicates that any finite automaton can be characterized by a regular expression and every regular expression can be realized by such an automaton, and hence answers our question.

For machines having only one-input and one-output terminal, regular expression language appears to be an ideal way to specify machine's action. The language is precise, allows no ambiguity, and moreover, can be written out in one line in contrast to flow tables and state diagrams. It is better related to word description of a sequential machine. In spite of these advantages, up to now, the language is however not commonly adopted in design practice. Two of the main reasons are: (1) lack of thorough understanding of the language. Some important problems are still unsolved, such as algorithms for determining equivalence of regular expressions, i.e., whether they represent the same signal set; the search for the existence of "canonical forms" (minimal complexity) of regular expressions; and problems relating algebraic manipulations of regular expressions, (2) as number of input and output terminals increases, the number of regular expressions needed to complete a specification of a machine also increases. The complexity of synthesis procedure increases accordingly. Even for a machine of moderate size, say 8 input-output terminal pairs, the problem becomes so complicated that a designer would rather choose a flow table or state diagram approach.

3.2 W-Machine Program Method

Another way of characterizing a sequential machine is by means of a so-called W*-machine program.[67],[68] The idea of a programmable machine is introduced by H. Wang.[103] Every W-machine is made up of an exterior mechanism and an internal program structure. The exterior of a W-machine consists of (1) an alphabet of two symbols 0 and 1, (2) a one-way tape, potentially infinite to the right, which is divided into squares. Each square is either marked (symbol 1) or erased (symbol 0), (3) a read-write head which is capable of marking or erasing a square; and (4) a control mechanism to carry out instructions of the program structure. The internal program structure of

W-machine is an ordered list of instructions, called the program of W-machine which is executable by the control mechanism. The types of instructions are (1) e; erase the square under scan (by the read-write head), (2) m; mark the square under scan, (3) +; move the read-write head one square to the right, (4) -; move the read-write head one square to the left, (5) t(A); transfer to the instruction whose address is A if the scanned square is marked, otherwise transfer to next instruction in the program. C.Y. Lee[67] has shown that such a W-machine with S instructions is completely equivalent to a Turing machine with not more than S states and every Turing machine with n states is completely equivalent to a W-machine with not more than 10n+1 instructions. Here the term "completely equivalent" means what can be done with one can also be done with the other, no more and no less. For example, a 1-state Turing machine described by flow table:

State	Symbol	
	0	1
q	m,+,q	e,-,q

is completely equivalent to a W-machine described by a program:

Addresses	Instructions
1	t(3)
2	m,+,t(3),m,t(2)
3	e,-,t(3),m,t(2)

If the instruction "-" is deleted from the program of a W-machine, we obtain a W^*-machine. A W^*-machine is essentially a finite automaton which can be characterized by input-output sequences. The input sequence of symbols of such an automaton may be considered to be copied on the tape of W^*-machine, with the initial input symbol on the leftmost square of the tape. The operation of W^*-machine is such that it first scans the initial square, writes the output symbol (0 or 1) on the square being scanned, moves one square to the right; and then again scans the square under the read-write head, writes the output symbol, moves one square to the right, and so on to the next state. An output sequence is constituted with symbols in squares to the left of read-write head (Figure 3).

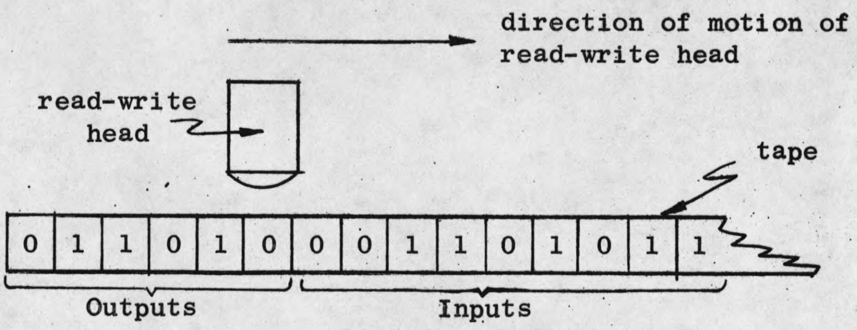


Figure 3. A Read-Write Head and a Tape

In other words, the "next state" and next output symbol are determined by the "present state" and present input symbol of the machine. Hence, any set of input signal set that characterizes a finite automaton also characterizes a W^*-machine and vice versa. Every finite automaton with s states is completely equivalent to a W^*-machine with not more than 10s+1 instructions, and every W^*-machine with b instructions differs from a finite automaton of not more than 2b+1 states by at most one unit of delay in the output. Like the situations encountered before in the discussion of regular expressions, the questions of equivalence of programs and that of finding an algorithm to minimize the number of individual instructions in any program also appear unanswered.

It is understandable that the techniques of characterizing sequential machines, by state diagrams or flow tables, by regular expressions (input-output sequences), or by programs of W^*-machine, are closely related to one another. These languages are readily translated from one form into another.[64],[57],[73] In synthesis, preference of selecting one over others depends on the type and size of machine and designer's taste. In view of this, it seems even at present moment, we are still lacking a sufficiently general and convenient method that would yield an adequate description of sequential machine specifications. In closing, let us illustrate the mentioned methods of characterizing machines by an example. Suppose the state diagram of a Mealy's model is given as shown in Figure 4.

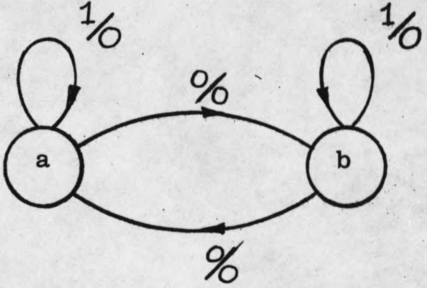


Figure 4. A Two-State Mealy's Model

The state diagram can be transformed into a regular expression.[73]: $1^*0(01^*0)^*1^*$ (state "a" is assumed to be the initial state). Likewise, it can be transformed into a flow table or a program as shown in Figure 5 and Figure 6.

State	Symbol	
	0	1
a	e,+,b	e,+,a
b	e,+,a	m,+,b

Figure 5. Flow Table of Machine of Figure 4

Addresses	Instructions
1	t(3)
2	e,+,t(4)
3	e,+,t(1),m,t(2)
4	m,+,t(4),m,t(1)

Figure 6. Program of Machine of Figure 4

4. ANALYSIS AND SYNTHESIS

4.1 The Notion of Experiments

In problems of analysis and synthesis of finite automata, we often wish to learn the answers to some questions about the nature of the machine. In particular, topics concerning the initial and final states of a machine, the equivalence of states, the reduction of a machine to its reduced form and the equivalence of machines are of special practical and theoretical interest. Information about the condition of a machine can often be obtained by applying sequence(s) of inputs and observing the resulting outputs.[79] We shall call such a sequence of inputs $I_1, I_2, \ldots, I_\ell$ an *experiment* (of *length* ℓ). Sometimes, an experiment (of length ℓ) is also called a *tape* (of *length* ℓ). For an incomplete machine, its output and next state are not defined for every input, we say that the machine has *input restrictions*.[8] Any experiment, which starting with the machine in state q_i, violates no input restriction of state q_i or any subsequent state, is called an *acceptable experiment for state* q_i. The *acceptable experiments for a machine* $S_{(n,m,p)}$ are the acceptable experiments for all n states of S. Clearly, any experiment is an acceptable experiment of a complete machine. Now we are ready to define some of the terms that were originated by E.F. Moore. A state q_i of machine S is said to be *indistinguishable* from state q_j of machine T if and only if (1) every experiment acceptable for state q_i is also acceptable for state q_j and vice versa, and (2) every acceptable experiment performed on S starting in state q_i produces the same output sequences as it would starting in q_j. State q_i is also said to be *equivalent* or *compatible* to state q_j. Similarly, a state q_i of a machine S is *distinguishable* (*non-compatible*) from a state q_j of a machine T if there exists an experiment of which the output sequence starting with machine S in state q_i differs from the output sequence starting with machine T in state q_j. S and T, of course, may be the same machine. Furthermore, we said a machine S is *distinguished* if no two states in S are indistinguishable. Two machines S and T are said to be *indistinguishable* (or equivalent) if and only if for each state q_i of S, there exists at least one equivalent state q_j of T, and for each state q_k of T, there exists at least one equivalent state q_e of S. Corresponding to each strongly connected machine S, there is a machine T which has the following properties: (1) T is indistinguishable from S (2) T has a minimal number of states (3) no two states in T are equivalent, and (4) T is unique within permutations of relabeling of states. T is then called the <u>reduced form</u> of S.

The problem concerning length of experiments for determining the state is initiated by E.F. Moore. Although Moore dealt with input-independent complete machines, the results sometimes are applicable to the whole class of machines with minor changes.[76], [22]

Moore investigated the lengths of experiments to distinguish states of sequential machine(s), and concluded that: (1) if $S_{(n,m,p)}$ is an input independent distinguished complete machine, then any two states of S can be distinguished by an experiment of length n-1, and this bound can not be lowered, (2) if $S_{(n,m,p)}$ and $T_{(n,m,p)}$ are input-independent complete machines and some state q_i of S can be distinguished from state q_j of T, then they can be distinguished by an experiment of length 2n-1, which is also the lowest bound possible. C.C. Elgot and J.D. Rutledge[27] generalized Moore's result, (2) for machines having different number of internal states. That is, if S has n states and T has n' states, they showed that the bound in (2) becomes n'+n-2. This bound, however, does not appear to be the best one as one can readily check by letting n'=n.

Another kind of experiment of interest is the one which determines the (final) state of a machine upon receiving a sequence of inputs. This is also sometimes called <u>terminal state experiment</u>[56] or <u>homing experiment</u>.[39] Moore has shown that for any input-independent machine $S_{(n,m,p)}$ in which all n states are distinguishable, an experiment of length no more than $\frac{n(n-1)}{2}$ is sufficient to determine the state of S at the end of the experiment. S. Ginsburg[41] considered a more general case. He defined that any experiment E, which is independent of the choice of an unknown state from a set A of admissible states of the machine as the initial state, to be a <u>uniform experiment</u> with respect to A. Then he estimated the length of uniform terminal state experiment to be $\frac{1}{2}[n(k^2-k) + \frac{2k-k^3}{6}]$ for any set A of $k \leq n$ states, and if k=n, the length is at most $\frac{n(2n-1)(n-1)}{6}$. At first, it seems rather important for us to notice that Moore assumed the machine initially was in an unknown initial state and the experiment performed was depending

depending upon that particular state whereas Ginsburg considered experiments applied to a set of states. Consequently, we may raise the question: "Is it necessarily true that bounds on lengths of experiments be lower than that of uniform experiments?" To one's surprise, the answer is NO. As was shown by T.N. Hibbard,[56] the least upper bounds on the lengths of the two kinds of experiments were proved to be equal.

Hibbard defined the terms minimal experiment and minimal uniform experiment. If $S_{(n,m,p)}$ is a machine and A is a subset of the set of states of S, then an experiment E for A is meant an experiment applied to each state in A. An experiment E for A is said to be minimal for A if the length of E is not greater than the length of any other experiment for A. A uniform experiment U for A is said to be minimal if the length of U is not greater than the length of any other uniform experiment for A. Hibbard concluded that for each distinguished complete machine $S_{(n,m,p)}$ and for each set A of k states of S, $1 < k \leq n$, if e(S,A) denotes the length of minimal terminal experiment for A and u(S,A) denotes the length of minimal uniform experiment for A, then

$$\text{Max}\left\{e(S,A) \mid \text{all } S,A\right\} = \text{Max}\left\{u(S,A) \mid \text{all } S,A\right\} = \frac{(2n-k)\ (k-1)}{2} \qquad (4\text{-}1)$$

This result indicates that when k=n, the bound $\frac{n(n-1)}{2}$ set up by Moore is in fact the best possible for complete machines in general.

In the case where $S_{(n,m,p)}$ is input-independent, the bound is even lower:

$$\text{Max}\left\{e(S,A) \mid \text{all } S,A\right\} = \text{Max}\left\{u(S,A) \mid \text{all } S,A\right\} = \frac{(2n-k)\ (k-1)}{2} - (k-2) \qquad (4\text{-}2)$$

Another problem of interest for sequential machine designers is to find the reduced form of a given machine. In the process of reducing the number of states of a machine to its minimum, compatible states play a pre-eminent role of removing redundancies.[61],[76] One of the necessary conditions for two states q_i and q_j to be compatible requires an inspection of all experiments that are acceptable to both q_i and q_j. This situation is indeed rather undesirable on account of the amount of labor involved. A possible way for improvement is to replace the phrase "inspection of all experiments" by "inspection of all experiments of length $t \leq k$" where k is a positive integer. We are, then, dealing with the problem of finding the smallest integer k.

Ginsburg[42] pointed out that for two machines S and T with n and m states respectively, state q_i in S is compatible with state q_j of T if for each experiment E of length $t \leq mn$, which is acceptable to both q_i and q_j, the output sequences are identical, i.e., $\lambda_S(I_x, q_i) = \lambda_T(I_x, q_j)$, for each $E = I_1, I_2, \ldots, I_x \ldots I_t$ with $1 \leq x \leq t$ and $t \leq mn$. Furthermore, this number mn can not be lowered. If S and T are the same machine, then mn is replaced by $\frac{n(n-1)}{2}$ which is the same least upper bound on a state terminal experiment for a complete machine.

In the case where S and T are input-independent, C.Y. Lee[67] pointed out that the number t is upper bounded by mn-1 but can not be lowered below mn-min. (m,n)+1. C.C. Elgot and J.D. Rutledge[27] indicated that the number mn-min (n,m)+1 is in fact the least upper bound. The result was further strengthened by S. Ginsburg[50] who assumed that the number of states n of S is equal to or less than the number of states m of T, and proved the bound to be n(m-1)+1. A similar conclusion is also reached by M.O. Rabin and D. Scott.[89] They specified two machines S and T to be equivalent if the set of all tapes (experiments) accepted by S is equal to the set of all tapes accepted by T. An immediate result concerns the length of experiments which distinguishes two machines. If S and T are two machines with n and m states, S is not equivalent to T if and only if there exists an experiment of length $t \leq mn$ which is accepted by S but not by T or vise versa.

Ginsburg[50] also indicated that the bound $\frac{n(n-1)}{2}$ on the length of experiments for two states q_i and q_j of $S_{(n,m,p)}$ to be compatible can be improved further if S is input-independent, that is, if $S_{(n,m,p)}$ is input-independent, q_i and q_j in S are compatible if and only if for every experiment E, which is acceptable by both q_i and q_j, of length $t \leq k$; where $k = \frac{n^2-2n}{4} + 1$ if n is odd, and $k = \frac{n^2-2n+1}{4} + 1$, if n is even; the output sequences are identical.

As to the actual procedures for constructing experiments, a thorough discussion is taken up by Gill in his book Introduction to the Theory of Finite State Machines.[39] These results concerning bounds on length of experiments may not yield instant application for machine simulation but the intrinsic concept of experiment performing certainly sets a guide to the analysis of properties of sequential machines. Some of the work still remains to be done in this area. Procedures for designing experiments to distinguish initial state or final state in a completely enumerational fashion and experiments to characterize and identify an unknown machine, and the possibility of applying concepts in this area to diagnosis of machines all deserve full attention.

4.2 Decomposition of Machines

In view of the difficulties encountered in analysis and synthesis of complex finite state sequential machines, many properties of the decomposition of machines have been investigated.[37],[57],[108] If a machine can be decomposed into a number of smaller units, each of which is made of a simpler structure than the whole, the procedure of design, testing, troubleshooting and replacement can be greatly simplified. Decompositions are generally classified into three types of construction: cascade, parallel, and loop-free.

A machine M is called a cascaded finite-state machine[37] if it is constructed by cascading a finite number of machines $M_1, M_2, \ldots, M_n$ in series, (Figure 7).

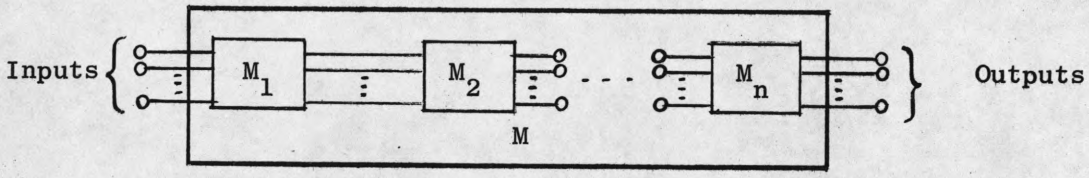


Figure 7. Cascaded Finite-State Machine

Inputs of M are inputs of M_1, and outputs of M are that of M_n's. The interconnections are such that outputs of M_i are identified with inputs of M_{i+1}, for $1 \leq i \leq n-1$. Several properties are apparent from this composition. If

$M_1, M_2, \ldots, M_n$ are machines with $n_1, n_2, \ldots, n_n$ states, respectively, then M is a machine with $n_1 \cdot n_2 \cdot n_3 \cdot \ldots \cdot n_n$ states. A machine with p states, where p is a prime number, is therefore not decomposable into simpler units unless some redundant states are added. If M is in reduced form, then every unit M_i must necessarily be in reduced form also. Moreover, if one of the M_i's is not strongly connected, then M is not strongly connected. The decomposibility of a machine into two stages in cascade is discussed by Gill[37] and by Yoeli.[108] Using connection matrix as a tool of analysis, Gill first defines a so-called n-m partition. The connection matrix [M] of a machine M with $n \cdot m$ states is said to have a n-m partition if it can be partitioned into n^2 $m \times m$ submatrices. Then the necessary condition for M to be decomposable into a m-state machine and an n-state machine in cascade is that an n-m partition must exist such that each row contains the same set of input symbols. This condition, however, is also sufficient as was pointed out by Hartmanis.[37] Yoeli uses the notion of transition graph (the conventional state diagram relative to a single input symbol) and properties of homomorphism between transition graphs to establish a necessary and sufficient condition for the decomposibility of M into two. A partition of state set of M is said to be admissible if it has the property that if any two states belong to the same class of partition, then their successors, relative to any input symbol, are again in the same class. In addition, an admissible partition is uniform if all classes contain the same number of states. A machine M then is said to be decomposible if and only if there exists a uniform admissible partition.

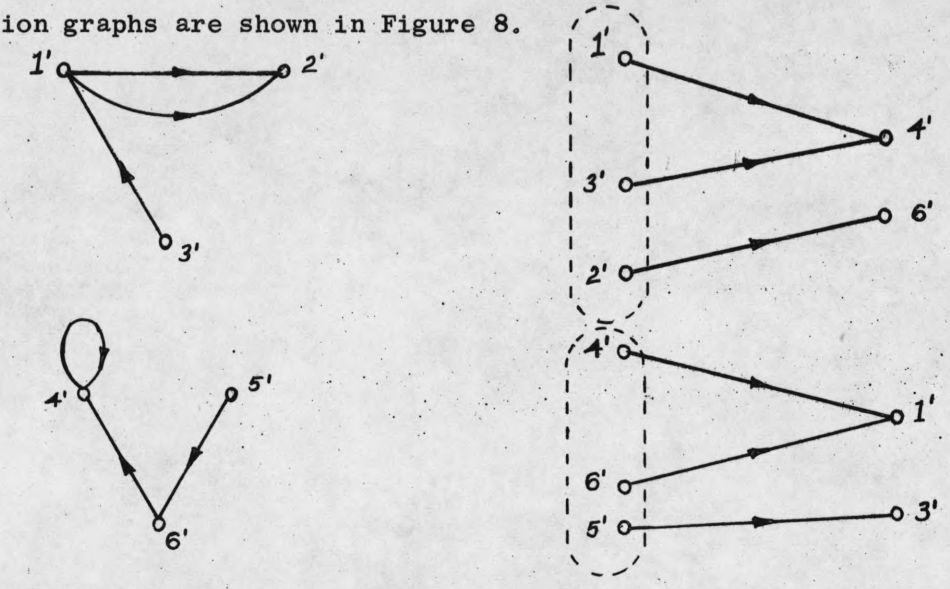
For a machine with $m \cdot n$ states, there exists $m!n!$ possible partitions, the test for decomposibility is usually a laborious task. As an example, let us consider a 6-state machine[37] described by a connection matrix [M_6] as:

$$[M_6] = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \left[\begin{array}{cccccc} 0 & 0 & (0/1) & 0 & 0 & (1/1) \\ (0/1) & 0 & (1/1) & 0 & 0 & 0 \\ (0/1) & 0 & (1/1) & 0 & 0 & 0 \\ (1/1) & 0 & (0/1) & 0 & 0 & 0 \\ 0 & (1/0) & 0 & (0/0) & 0 & 0 \\ (1/1) & (0/0) & 0 & 0 & 0 & 0 \end{array}\right] \end{array}$$

At first glance, whether $[M_6]$ is decomposable is by no means obvious. After relabelling, $[M_6]$ becomes $[M_6']$ which is decomposable:

$$[M_6'] = \begin{array}{r|ccc|ccc} & 1' & 2' & 3' & 4' & 5' & 6' \\ \hline 1'(=1) & 0 & (1/1) & 0 & (0/1) & 0 & 0 \\ 2'(=6) & (1/1) & 0 & 0 & 0 & 0 & (0/0) \\ 3'(=4) & (0/1) & 0 & 0 & (0/1) & 0 & 0 \\ \hline 4'(=3) & (0/1) & 0 & 0 & (1/1) & 0 & 0 \\ 5'(=5) & 0 & 0 & (0/0) & 0 & 0 & (1/0) \\ 6'(=2) & (0/1) & 0 & 0 & (1/1) & 0 & 0 \end{array}$$

The transition graphs are shown in Figure 8.



Under Input 1

Under Input 0

Figure 8. Transition Graphs of M_6'

The partition states of M_6' into classes $\{4',5',6'\}$ is clearly both admissible and uniform.

The decomposition of a sequential machine into several simpler ones operating in parallel was discussed by Hartmanis.[52] Each of the simpler machines operate on differently reduced information and the combination of these individual operations yields the desired results, (Figure 9). The decomposition of a complex operation into simpler operations carried out in parallel is also connected with the notion of partition. A partition Π on the set

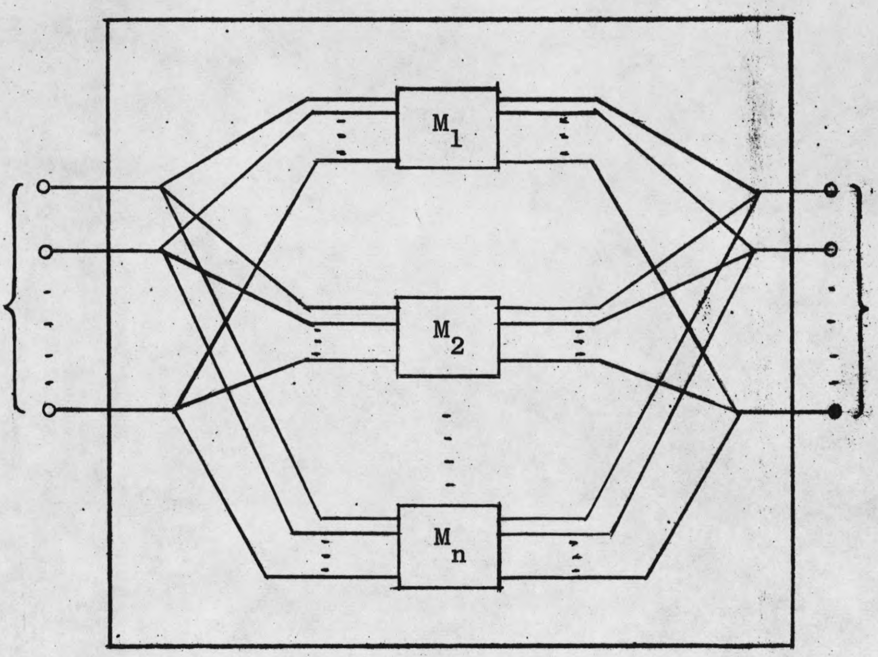


Figure 9. A Parallel Operating Machine

S is defined as a collection of disjoint subsets $\{S_\alpha\}$ of S such that their union is S. Symbol 0 denotes the smallest partition on S in which each subset contains exactly a single element. Symbol I denotes the largest partition on S where the only subset is the set S itself. Suppose an operation * is defined on the set S that a*b is closed for any a, b $\in$ S, we say a partition Π has the substitution property with respect to operation * if A,B,C are disjoint subsets of S and a $\in$ A, b $\in$ B, a*b=c $\in$ C implies any other element a' $\in$ A, b' $\in$ B, then a'*b' is also in C. The direct product of machine M_1 (with input set I_1, state set S_1) and machine M_2 (with input set I_2, state set S_2) is the machine M with input set equal to the cartesian product $I_1 \times I_2$ and set of states $S_1 \times S_2$. Moreover, for given two partitions Π_1 and Π_2 on S, the partition $\Pi_1 \cdot \Pi_2$ means the set of all intersecting subsets due to Π_1 and Π_2. Two sets U and V are chain connected if there exists a sequence of sets $U=X_1, X_2, \ldots, X_n=V$ such that the intersection of X_i and X_{i+1} is not empty for $1 \leq i \leq n-1$. If A is a subset of partition of Π_1, by partition $\Pi_1+\Pi_2$ we mean the collection of subsets each of which contains A is the set union of all subsets of Π_1 and Π_2 which are chain connected to A. For example, if $\Pi_1=\{x;y;\ zw\}$, $\Pi_2 = \{xy,z\}$ then

$\Pi_1 \cdot \Pi_2 = \{x;y;z\}$ and $\Pi_1 + \Pi_2 = \{xy;z;w\}$. The necessary and sufficient condition for a machine M to be decomposable into two machines M_1 and M_2 whose direct product is isomorphic to M is then the existence of two nontrivial *permutable partitions Π_1 and Π_2 with substitution properties for M such that: $\Pi_1 + \Pi_2 = I$ and $\Pi_1 \cdot \Pi_2 = 0$. In another paper[54], Hartmanis indicates that every loop-free realization of a sequential machine from n smaller machines corresponds to a set of n partitions $\{\Pi_1 \cdot \Pi_2, \ldots, \Pi_n\}$ with the substitution property whose product is the zero partition 0; i.e. $\prod_{i=1}^{n} \Pi = 0$. By loop-free we mean that in a set of interconnected machines, no subset consisting of two or more machines forms a loop. Conversely, every such set of n partitions corresponds to a realization of the given machine from n smaller machines.

The significance of these results are shadowed by the fact that the authors use a state diagram as their starting point; nevertheless the concepts of decomposition may serve as stepping stones toward our ultimate goal of synthesis of sequential machines by means of interconnecting machines of less complexity and smaller sizes. Many problems remain unsolved, some even have not been formulated. The test of decomposability of course deserves improvement. An algorithmic procedure for the actual decomposition operation which is programmable on digital computers is also an immediate need. Problems of interconnecting machines to form various configurations, such as series-parallel, loop with feedback (Figures 10,11) and the timing problem among each unit's inputs and outputs and their related properties likewise require investigation.

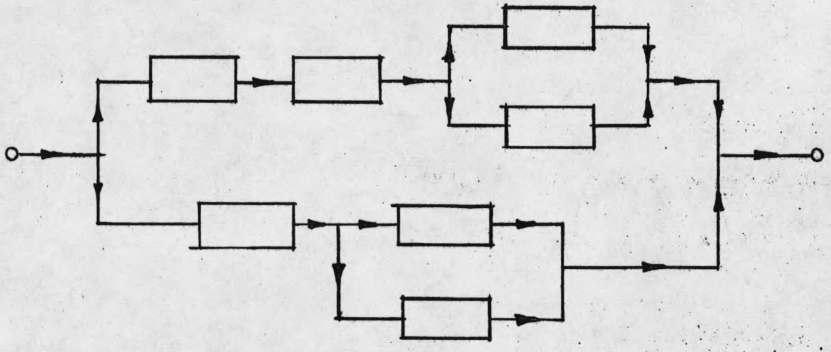


Figure 10. An Example of Series-Parallel Interconnection of Machines

*Π_1 and Π_2 are permutable if any two blocks of A of Π_1 and B of Π_2 which are contained in the same block of $\Pi_1+\Pi_2$ have a nonempty intersection.

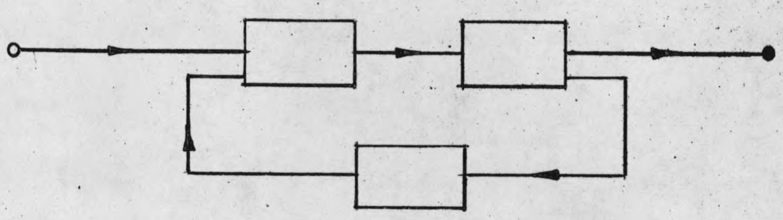


Figure 11. An Example of Interconnection of Machines with Loop Feedback

4.3 On Machine Synthesis Techniques

Almost in every field of engineering science the task of design appears to be the most difficult one. A good engineering design often depends upon a careful application of formal procedures together with the ingenuity and experience of design engineers. This is of course no exception in the case of sequential machine synthesis. While a formal and compact procedure is still lacking, the present art of design relies heavily on the intuition and cleverness of the designer.

As pointed out previously, the first stumbling block encountered in sequential machine synthesis is the lack of a precise, formal and useful mathematical language to describe machine's desired function. Huffman's flow-table method,[61] Moore-Mealy's state diagram approach or the use of regular expressions prove to be powerful tools for small scale machines. With the design goal being emphasized on achieving reduced form machines, numerous problems on state-reduction[44] and state-assignment[53],[101] have been investigated. Aizeman, Gusev, Rozonoer, Smirnova and Tal[2] studied the reduction problem from a viewpoint of imposing restrictions on inputs. The most general case is the same as in Huffman's procedure in which no input restrictions are assumed, i.e., all input sequences are permitted. If, however, a machine is only allowed to accept tapes in which successive repetitions of any input symbol do not occur in the input sequences, the number of states needed can be reduced relative to that when these input restrictions are removed. The machine is considered to consist of a "fast" automaton F which operates over fast cycles between input changes and an output converter, (Figure 12). If we further restrict the successive inputs to the machine to be ordered pairs of the input alphabet $\{I, I_\tau\}$,

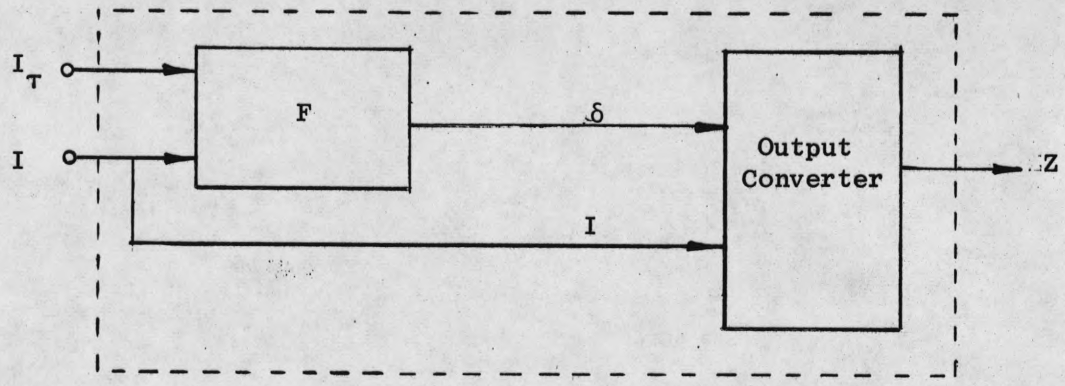


Figure 12. A Machine Consisting of a Fast Automaton and an Output Converter

where I_τ repeats the values of the I with a lag of τ (τ is the period of F), then the minimum number of states required is equal to the largest of α_i, where α_i is the number of distinct states Z_j into which each I_i transforms the automaton.

Generally speaking, the purpose of state-reduction is to remove redundant states in a primitive flow table or a state diagram. The well-known Huffman's state-merging technique is applicable only to complete machines. For machines that are not completely specified, the reduction problem is an exceedingly difficult one. At the present time, no automatic procedure which generates the desired results without any enumeration is available. A partially enumerative procedure is given by Paull and Unger.[87] Although the procedure depends on the complexity of particular examples, it seems to be able to handle problems of moderate size. A more powerful technique has been developed by Ginsburg.[42],[44] Part of his process is automatic and part of it depends on enumerating. The main disadvantage of Ginsburg's approach is that it is too "lengthy" to be useful to attack any practical problems since it is not programmable on a computer. Notion of state-reduction also bears a close relationship with the structure of a sequential machine. Sometimes the simplest realizations of a sequential machine can be destroyed by state reduction.[55] The necessary information flow needed for realization of a machine from smaller units may be disrupted by merging states. Hence a careful analysis on the structure of the machine should be made before we apply any reduction procedure.

By state-assignment we mean the assignment of codes to the internal states of sequential machine so as to obtain an economical logical realization. A practical and useful algorithm of state-assignment, which is applicable to both complete and incomplete machines, is described by Armstrong.[5] Although in some cases Armstrong's procedure will not attain truly economic realizations which are known to exist, it, however, has the superiority of being able to handle large problems, say machines with 100 states and 30 input symbols, with the aid of a digital computer. A somewhat different approach is taken by Hartmanis[53] and Stearns.[101] Their approach stresses upon the concept of assignments with reduced dependencies, i.e., each internal state variable at any time t depends only on a small subset of these variables at previous moment t-1 and on the input variables. The procedure may not always produce the most economical realization but the notions of substitution property and partition pairs, as defined in their papers, certainly play an important role in machine decompositions.

Apart from the flow-table method, there is Ginsburg's[45] technique by which a machine is synthesized from a set of finite input-output sequences. If input and output symbols are denoted by I and Z, respectively, then the set $\sum = \left\{ I^i_1, \ldots, I^i_{k(i)};\ Z^i_1, \ldots, Z^i_{k(i)};\ 1 \leq i \leq S,\ S \text{ is finite} \right\}$ is called a finite set of input-output sequences where superscripts denote the sequence number and subscripts denote the individual terms of the i^{th} sequence. The procedure begins with finding a lower bound n on the number of states of a machine which can have the response required by certain input-output sequences. Then consider all such machines with n states. If none accepts all sequences in $\sum$, then consider all machines with n+1 states, and then, if necessary, machines with n+2 states, ..., etc. Once such a machine is found, we are assured to have obtained a minimal machine. The result is profound but the amount of work that one has to do in carrying out this procedure is also tremendous if not impossible in a finite time.

The synthesis technique of sequential machines from regular-expression languages was first described by McNaughton and Yamada.[73] Algorithms are given for finding a state diagram corresponding to a regular expression and vice versa. A similar approach was discussed by Lee.[67] Lee shows that if the descriptions of flow-table (or its equivalent state diagram) for the machine realizing the regular expressions α and β are known, then the flow-tables corresponding to the regular expressions $\alpha \vee \beta$, $\alpha \circ \beta$ and α^* can be constructed. Ultimately, the flow-tables of any regular expressions which are formed by applying operations "v", "∘" and "*" a finite number of times on α and β can also be constructed. Ott and Feinstein,[86] however, considered the problem from a somewhat different view point. They use the notion of so-called improper state diagram (ISD), which is simply the usual state diagram of a nondeterministic machine,[89] i.e., a machine that can be in any finite number of states simultaneously. They indicate that the occurrence of an event of any regular expressions can always be realized by a nondeterministic machine whose behavior is described by an improper state diagram; and for any improper state diagram, there exists an equivalent Mealy-type state diagram. Their construction method thus assures a realization for any given regular expression. The above mentioned results concerning synthesis techniques are, at this moment, only seen applications to small machines. For machines of reasonable size, say 20 feedback loops, the procedures become too complicated to handle.

5. CONCLUSION

A brief study of the application of concepts of automata theory to analysis and synthesis of sequential machines has been made. The investigation was taken from an engineer's viewpoint and was by no means complete. After defining models of sequential machines, two ways of characterizing a machine, namely, the input-output signal set (or regular expression) method and the computer program method, were indicated. The discussion on analysis of machines was concentrated on experiment-performing, a way to examine a machine's structure by merely applying inputs and observing outputs; and machine decompositions. Lastly, some of the results and difficulties related to state-reduction and state-assignment in machine synthesis procedures were mentioned.

REFERENCES

1. Aizerman, M.A., Gusev, L.A., Rozoner, L.I., Smirnova, I.M., and Tal, A.A., "Finite Automata", Automation and Remote Control, Part I, vol. 21, pp. 156-164, Oct.; Part 2, vol. 21, pp. 248-255, Nov. 1960. Review: IRE Trans. on Elec. Comp.,vol. EC-10, p. 544, Sept. 1961.

   (Some basic content and problems of theory of finite automata and possible ways of realizing finite automata are discussed.)

2. Aizeman, M.A., Gusev, L.A., Rozoner, L.I., Smirnova, I.M. and Tal, A.A., "On Methods for Realizing a Finite Automaton Whose Cyclical Nature is Determined by the Variation of the Input State", Automation and Remote Control (Automatika i Telemekhanika), vol. 21, pp. 1576-1594; 1960. Review: IRE Trans. on Elec. Comp., vol. EC-11, no. 3; June 1962.

   (The paper studies 3 methods for an economical (in the sense of number of states) realization of a finite automaton specified by an equation (table) and operating in a cyclic manner determined by the variation in the states of its inputs.)

3. Aribib, M., "Turing Machines, Finite Automata and Neural Nets", J. Assoc. Comp. Mach., vol. 8, no. 4, pp. 467-475; Oct 1961.

   (The author compares the notions of Turing machine, finite automaton and neural net and discusses theorems which are related to papers of Copi, Elgot and Wright; Rabin and Scott; and McCulloch and Pitts.)

4. Arden, D.N., "Delayed-Logic and Finite-State Machines", in Switching Circuit Theory and Logical Design, Proc. of 2nd Annual Symp., Detroit, Mich., Oct.1961 and papers from the 1st Annual Symp., Chicago, Ill., Oct.1960, (published by AIEE, New York, N.Y.)

   (An extension of Kleene's theory of neural nets and finite automata to the realization of sequential machines with delay elements. Two main results concerned with the relationships between the computational rate of devices of which the circuit is comprised, the complixity of the computation being performed by the circuit, and the delay in the circuit from the inputs to the outputs are described.)

5. Armstrong, D.B., "A Programmed Algorithm for Assigning Internal Codes to Sequential Machines", IRE Trans. on Elec. Comp., vol EC-11, no. 4, pp. 466-473, Aug. 1962.

   (A procedure for assigning codes to the internal states of a sequential machine which is applicable to both completely and incompletely specified states tables.)

6. Ashenhurst, R.L., "The Decomposition of Switching Functions", Proc. Intnatl. Symp. Theory of Switching, Harvard Univ., Cambridge, Mass., Part 1, pp. 74-116; April 1957.

7. Aufenkamp, D.D., "Analysis of Sequential Machines II", IRE Trans. on Elec. Comp., vol. EC-7, pp. 299-306; Dec. 1958.

8. Aufenkamp, D.D. and Hohn, F.E., "Analysis of Sequential Machines", IRE Trans. on Elec. Comp., vol. EC-6, pp. 276-285; Dec. 1957.

9. Bartee, T.C., "Computer Design of Multiple-Output Logical Networks", IRE Trans. on Elec.Comp., vol. EC-10, no. 1, March 1961, pp. 21-30. Review: IRE Trans. on Elec. Comp., vol. EC-10, no. 3, Sept. 1961.

(Use of computer program to derive a set of minimal Boolean expressions describing a logical network with multiple outputs.)

10. Bellman, R., "Sequential Machines, Ambiguity and Dynamic Programming", J. Assoc. Comp. Mach., vol. 7, pp. 24-28, 1960.

11. Blokh, A. Sh., "Equivalent Transformations of Sequential Machines", (Avtomatika i Telemekhanika, vol. 21, no. 11, pp. 1490-1497, Nov. 1960) (translation: Automation and Remote Control, vol. 21, no. 11, May,1961) Review: IRE Trans. on Elec. Comp., vol. EC-11, p. 94, Feb. 1962.

(The paper defines certain equivalent transformations of sequential machines and the corresponding structures of the machines.)

12. Brozozowski, J.A. and McClusky, E.J., "Signal Flow Graph Techniques for Sequential Circuit State Diagrams", U.S. Govt. Res. Rept., vol. 37, p. S-35(A); May 20, 1962.

(The application of signal flow graph techniques to the problem of characterizing sequential circuit state diagrams by regular expressions is discussed.)

13. Brozozowski, J.A., "A Survey of Regular Expressions and Their Applications", IRE Trans. on Elec. Comp., vol. EC-11, pp. 324-335; June 1962.

(The relation of regular expressions to sequential circuits is studied. Methods for constructing sequential circuits (state diagrams) from regular expressions and methods for obtaining a regular expression from a state diagram of a sequential circuit are discussed. Also mentioned are algebraic properties of regular expressions, and characteristics of improper state diagrams.)

14. Buchi, J.R., Elgot, C.C. and Holland, J.H., "Research in Automata Structure, Behavior and Design", College of Lit. and Arts, Michigan Univ., Ann Arbor, Mich., Final Report No. 2794-15-F; Nov. 1960.

(A summary of research on automata design algorithms in formal system of logic and theoretic computers with iterative structure and parallel action.)

15. Burks, A.W., "The Logic of Fixed and Growing Automata", Proc. Intnatl. Symp. on the Theory of Switching, April 2-5, 1957, in "The Annals of the Comput. Lab.", Harvard Univ., Cambridge, Mass., vol. 29, pp. 147-188; 1959. Review: IRE Trans on Elec. Comp., vol. EC-9, no. 4.; Dec. 1960.

(The main concern is discrete synchronous and deterministic computers.)

16. Burks, A.W., "Computation, Behavior, and Structure in Fixed and Growing Automata", Behavioral Sci., vol. 6, pp. 5-21; Jan. 1961.

(The theory of automata in relation to its internal behavior and the possibility of its being able to self-reproduce is discussed.)

17. Burks, A.W., Buchi, J.R., Elgot, C.C. and Wright, J.B., "Design Algorithms in Automata Languages", Dept. of Philosophy, Logic of Computers Group, Univ. of Michigan, Ann Arbor, Report No. 2755-11-F; July 1960.

(Reports on the design algorithms for automata languages.)

18. Burks, A.W. and Copi, I.M., "The Logical Design of an Idealized General Purpose Computer", J. Franklin Inst., vol. 261, pp. 299-314 and 421-436; 1956.

19. Burks, A.W. and Wang, H., "The Logic of Automata", J. Assoc. Comput. Mach., vol. 4, no. 2, pp. 193-218, no. 3, pp. 279-297; 1957.

(The use of logical systems and techniques in analysis of the structure and behavior of automata.)

20. Burks, A.W. and J.B. Wright, "Theory of Logical Nets", Proc. IRE, vol. 41, no. 10, pp. 1357-1365, Oct. 1953.

(One of the first papers to apply rigorously two-valued logic to digital computer circuits. The idea of logical net along with "stroke element" and "delay element" are introduced. Sets of equations associated with various types - well-behaved, well-formed deterministic - of nets are discussed.)

21. Burks, A.W. and Wright, J.B., "Sequence Generators and Digital Computers", Univ. of Michigan, Rept. 03105-19-T, Feb. 1961.

(A generalization of the concepts of digital computers, finite automaton, logical net and other information-processing systems.)

22. Cadden, W.J., "Equivalent Sequential Circuits", IRE Trans. on Circuit Theory, vol. CT-6, pp. 30-34; 1959.

(The functional behavior of Moore's model and Mealy's model and Huffman's asynchronous machine are equivalent in the sense that one can be transformed into any other.)

23. Copi, I.M., Elgot, C.L. and Wright, J.B., "Realization of Events by Logical Nets", J. Assoc. Comp. Mach., vol. 5, pp. 181-196; April 1958.

(New formulation and new proofs of Kleene's work are presented. See Kleene (66).)

24. Curtis, H.A., "Multiple Reduction of Variable Dependency of Sequential Machines", J. Assoc. Comp. Mach., vol. 9, no. 3, pp. 324-345, July 1962.

25. Davis, A.S., "Markov Chains as Random Input Automata", Am. Math. Monthly, pp. 264-267, 1961.

26. Elgot, C.C., "Decision Problems of Finite Automata Design and Related Arithmetics", Trans. Am. Math. Soc., vol. 98, pp. 21-51; Jan. 1961.

(A class of problems concerning the existence of automata which produce specified mappings from input sets into output sets is considered in a formal manner.)

27. Elgot, C.C. and Rutledge, J.D., "Operations on Finite Automata", in Switching Circuit Theory and Logical Design", Proc. of Second Annual AIEE Symp., pp. 129-132; Sept. 1961.

(Unification of various points of view that have been adopted in the treatment of the synthesis of finite Automata.)

28. Elspas, B., "The Theory of Autonomous Linear Sequential Networks", IRE Trans. on Circuit Theory, vol. CT-6, pp. 45-60, 1959.

(The relation between network logical structure and network sequential behavior is investigated.)

29. Fitch, F.B., "Representation of Sequential Circuits in Combinatory Logic", Phil. Sci., vol. 25, pp. 263-279, 1958.

30. Fleck, A.C., "Isomorphism Groups of Automata", J. Assoc. Comp. Mach., vol. 9, no. 4, pp. 469-476; Oct. 1962.

(Certain algebraic properties of automata and their relationship to the structure (properties of the next state function) of automata are discussed.)

31. Friedland, B., "Linear Modular Sequential Circuits", IRE Trans. on Circuit Theory, vol. CT-6, pp. 61-68, 1959.

(An extension of Huffman's concepts of synthesis and analysis of sequential circuits to the case when circuits comprise unit delays, modulo-p (p=prime) adders, and ideal amplifiers with integral values of gain < p.)

32. Friedman, J., "A Decision Procedure for Computations of Finite Automata", J. Assoc. Comp. Mach., vol. 9, no. 3, pp. 315-323; July 1962.

(A decision procedure to determine if two automata have the same computation.)

33. Ghiron, H., "Rules to Manipulate Regular Expressions of Finite Automata", IRE Trans. on Elec. Comp., vol. EC-11, no. 4, pp. 574-575, Aug. 1962.

(Some algebraic rules to manipulate regular expressions are presented.)

34. Gill, A., "Comparison of Finite-State Models", IRE Trans. on Circuit Theory, vol. CT-7, no. 2, pp. 178-179, June 1960.

(This note shows the "Moore Machine" and "Mealy Machine" are equivalent. Another paper is by Cadden, "Equivalent Sequential Machines".)

35. Gill, A., "Characterizing Experiments for Finite-Memory Binary Automata", *IRE Trans. on Elec. Comp.*, vol. EC-9, no. 4, pp. 469-471; Dec. 1960.

(Discuss properties and construction of "experiments" which determine the characteristics of a discrete automata with a finite memory.

36. Gill, A., "State-Identification Experiments in Finite Automata", *Information and Control*, vol. 4, pp. 132-154, 1961. Review: *IRE Trans. on Elec. Comp.*, vol. EC-11, p. 584, Aug. 1962.

(A procedure for the construction of experiments to determine (1) initial state (2) terminal state of a machine.)

37. Gill, A., "Cascaded Finite-State Machines", *IRE Trans. on Elec. Comp.*, vol. EC-10, pp. 366-370, Sept. 1961. Review: *IRE Trans. on Elec. Comp.*, vol. EC-11, p. 94, Feb. 1962.

(A necessary condition is established for a machine to be decomposable into two machines in cascade.)

38. Gill, A., "A Note on Moore's Distinguishability Theorem", *IRE Trans. on Elec. Comp.*, vol. EC-10, pp. 290-291, 1961.

(The Moore's theorem on length of minimal experiment which distinghishes states of two machines is improved.)

39. Gill, A., *Introduction to the Theory of Finite-State Machines*, McGraw-Hill Book Company, Inc., 1962.

40. Gillespie, R.G. and Aufenkamp, D.D., "On the Analysis of Sequential Machines", *IRE Trans. on Electronic Computers*, vol. EC-7, pp. 119-122; June 1958.

41. Ginsburg, S., "On the Length of the Smallest Uniform Experiment which Distinghishes the Terminal States of a Machine", *J. Assoc. Comp. Mach.* vol. 5, no. 3, pp. 266-280; July 1958.

(A bound is established on the length of minimal uniform experiment which determines the terminal state of a machine.)

42. Ginsburg, S., "On the Reduction of Superflous States in a Sequential Machine", *J. Assoc. Comp. Mach.*, vol. 6, pp. 259-282; 1959.

43. Ginsburg, S., "A Synthesis Technique for Minimal-State Sequential Machines", *IRE Trans. on Elec. Comp.*, vol. EC-8, pp. 13-24; 1959.

(A synthesis technique for minimal state sequential machines from a set of input-output sequences.)

44. Ginsburg, S., "A Technique for the Reduction of a Given Machine to a Minimal-State Machine", *IRE Trans. on Elec. Comp.*, vol. EC-8, pp. 346-355; 1959.

45. Ginsburg, S., "Synthesis of Minimal-State Machines", *IRE Trans. on Elec. Comp.*, vol. EC-8, pp. 441-449; 1959.

46. Ginsburg, S., "Connective Properties Preserved in Minimal-State Machines", J. Assoc. Comp. Mach., vol. 7, pp. 311-325; 1960.

(The design of a sequential machine and the properties of the connectives in the minimal state machine are discussed.)

47. Ginsburg, S., "Some Remarks on Abstract Machines", Trans. Am. Math. Soc., vol. 96, pp. 400-444; Sept. 1960.

(The abstract concept of Quasi-machine and related properties are introduced.)

48. Ginsburg, S., "Sets of Tapes Accepted by Different Types of Automata", J. Assoc. Comp. Mach., vol. 8, pp. 81-86; Jan. 1961.

(A number of different types of one-way automata and their family relationship are presented.)

49. Ginsburg, S., "A Comparison of the Work Done by Generalized Sequential Machines and Turing Machines", Tech. Memo. TM-604, pp. 1-16; March 1961.

50. Ginsburg, S., "Compatibility of States in Input-Independent Machines", J. Assoc. Comp.,Mach., vol. 8, no. 3, pp. 400-404; July 1961.

(To find, for a certain class of machines, the smallest integer K having the property that, if two given states do non-contradictory work for all tapes of length less than or equal to K, these two states do non-contradictory work for all tapes for any length.)

51. Ginsburg, S., "Examples of Abstract Machines", IRE Trans. on Elec. Comp., vol. EC-11, pp. 132-135; April 1962. Review: IRE Trans. on Elec. Comp., vol. EC-11, pp. 721-722; Oct. 1962.

52. Hartmanis, J., "Symbolic Analysis of a Decomposition of Information Processing Machines", Information and Control, vol. 3, pp. 154-178; 1960.

(The study of decomposing (replacing) a complex finite state sequential machine by several simpler ones which operate in parallel and yield the same result.)

53. Hartmanis, J., "On the State Assignment Problem for Sequential Machines, I", IRE Trans. on Elec. Comp., vol. EC-10, no. 2, pp. 157-164; June 1961.

(A method for assigning internal states of a sequential machine is presented.)

54. Hartmanis, J., "Loop-Free Structure of Sequential Machines", Information and Control, vol. 5, pp. 25-43; 1962.

(With the properties of partitions with substitution and partition pairs, the author studies the realization of sequential machines from several smaller machines.)

55. Hartmanis, J. and Stearns, R.E., "Some Dangers in State Reduction of Sequential Machines", Information and Control, vol. 5, pp. 252-260; Sept. 1962.

(Reduction by merging states sometimes disrupted the systematic information flow in the unreduced machines and can destroy the simplest realizations of a sequential machine.

56. Hibbard, T.N., "Least Upper Bounds on Minimal Terminal State Experiments for Two Classes of Sequential Machines", J. Assoc. Compt. Mach., vol. 8, no. 4, pp. 601-612; Oct. 1961.

(For a machine of n distinguished states, an experiment of length $\frac{n(n-1)}{2}$ is the best possible bound to determine the state of this machine. For input independent machines, the bound is even lowered.)

57. Hohn, F.E., Seshu, S., and Aufenkamp, D.D., "The Theory of Nets", IRE Trans. on Elec. Comp., vol. EC-6, pp. 154-161; Sept. 1957.

(A unifying concept of a net is used as background for the theory of sequential machines.)

58. Holland, J.H., "Iterative Circuit Computers", Proc. Western Joint Conf., San Francisco, Calif., May 3-5, 1960; pp. 259-266.

(Mathematical characterizations for a given condition are related to a program which will establish a theory of adaptive system towards a concept of automaton generators.)

59. Holland, J.H., "A Universal Computer Capable of Executing an Arbitrary Number of Sub-Programs Simultaneously", Proc. 1959 Eastern Joint Computer Conf., Boston, Mass., Dec. 1-3, 1959; pp. 108-113.

(A universal computer which has iterative circuits and whose structure and behavior can be formulated in a manner which will provide a formal basis for a theoretical study of automata with changing structure.)

60. Holland, J.H., "Cycles in Logical Nets", J. of Franklin Institute, vol. 270, no. 3, pp. 202-226; Sept. 1960. Review: IRE Trans. on Elec. Comp., vol. EC-10, no. 2; June 1961.

(The influence of cycles in a logical net upon the complexity of its behavior is investigated.)

61. Huffman, D.A., "Synthesis of Sequential Switching Circuits", J. Franklin Inst., vol. 257, no. 3, pp. 161-190; March 1954; no. 4, pp. 275-303; April 1954. J. Symbolic Logic, vol. 20, pp. 69-70; March 1955. Reviewed by R.J. Nelson.

(A general method for synthesis of switching circuits.)

62. Jeffrey, R.C., "Some Recent Simplifications of the Theory of Finite Automata", TR 219, Research Laboratory of Electronics, M.I.T., May 1959.

(Kleene's theory of representation of events by finite automata is presented in a simplified and strengthened form largely due to Medvedev, and Rabin and Scott.)

63. Karatsuba, A.A., "Solution of a Problem from the Theory of Finite Automata", Uspekhi Matematiki Nauk, vol. 15, pp. 157-159; 1960 (English translation No. K-218, Morris D. Friedman, Inc., West Newton, Mass.)

64. Karp. R.M., "A Note on the Application of Graph Theory to Digital Computer Programming", Information and Control, vol. 3, pp. 179-190, 1960.

(A method is given to transform a flow-chart of a computer program into a directed graph.)

65. Keller, H.B., "Finite Automata, Pattern Recognition, and Preceptions", J. Assoc. Comp. Mach, vol. 8, no. 1, pp. 1-20; 1961.

(The idea of the discrimination function of an automaton is emphasized. Two pattern recognition devices and mathematical characterization of a perception are also given.)

66. Kleene, S.C., "Representation of Events in Nerve Nets and Finite Automata", Automata Studies, (Annals of Math. Studies No. 34) Princeton Univ. Press, pp. 3-42; 1956.

(McCulloch-Pitts neuron nets and other representable stimuli are investigated. The notion of "regular events" and realization of finite automata are also discussed.)

67. Lee, C.Y., "Automata and Finite Automata", Bell Syst. Tech. J., no. 39, pp. 1267-1295; 1960. Review: IRE Trans. on Elec. Comp., vol. EC-10, no. 2; June 1961.

(The programming approach of characterizing an automaton is emphasized.)

68. Lee, C.Y., "Categorizing Automata by W-Machine Programs", J. Assoc. Comp. Mach., vol. 8, no. 3, pp. 384-399; July 1961.

(Algorithms showing conversion of internal descriptions of machines to classes of programs based on program structure introduced by Wang.)

69. McCluskey, E.J., Jr., and Unger, S.H., "A Note on the Number of Internal Variable Assignments for Sequential Switching Circuits", IRE Trans. on Elec. Comp., vol. EC-8, pp. 439-440; Dec. 1959.

(Gives a formula which indicates the number of different assignments which can be made for flow tables having a given number of rows.)

70. McCluskey, E.J., Jr., "A Comparison of Sequential and Iterative Circuits", AIEE Trans., Commun. and Electronics, no. 46, Part 1, vol. 78, pp. 1039-1044; Jan. 1960. Review: IRE Trans. on Elec. Comp., vol. EC-9, no. 3; Sept. 1960.

(By way of example, the author illustrates the design of an iterative network that will perform the desired operation in a parallel mode.)

71. McCluskey, E.J., Jr., and Bartee, T.C., A Survey of Switching Circuit Theory, McGraw-Hill Book Company, Inc., 1962.

72. McCulloch, W.S. and Pitts, W., "A Logical Calculus of the Ideas Immanent in Nervous Activity", Bull. Math. Biophys., vol. 5, pp. 115-133; 1943.

(One of the first papers to use the concept of symbolic logic for realization of automata. A model of neuron net whose structure is a reflection of the logical propositions describing the properties of the net is presented.)

73. McNaughton, R.F., and Yamada, H., "Regular Expressions and State Graphs for Automata", IRE Trans. on Elec. Comp., vol. EC-9, no. 1, pp. 39-48; March 1960.

(Algorithms are given for constructing state diagrams of sequential machines from given regular expressions and vice versa.)

74. McNaughton, R., "Symbolic Logic and Automata", U.S. Govt. Res. Repts. vol. 35, p. 601(A); May 16, 1961.

(Use of symbolic logic to describe the behavior of an automaton is discussed.)

75. McNaughton, R., "The Theory of Automata, a Survey", pp. 379-421 in Advances in Computers, F.L. Alt (ed), vol. 2, Academic Press, Inc., New York; 1961.

76. Mealy, G.H., "A Method for Synthesizing Sequential Circuits", Bell Syst. Tech. J., vol. 34, no. 5, pp. 1045-1079; Sept. 1955. Review: J. Symbolic Logic, vol. 22, pp. 334-335; Sept. 1957. Bell System Monograph No. 2458.

(A technique for synthesizing synchronous automata.)

77. Mezei, J.E., "Minimal Characterizing Experiments for Finite Memory Automata", IRE Trans. on Elec. Comp., vol. EC-10, p. 288; 1961.

(The relationship between finite memory automata and the corresponding minimal characterizing experiments is made.)

78. Minsky, M.L., "Some Universal Elements for Finite Automata", Automata Studies, (Annals of Math. Studies No. 34), Princeton Univ. Press, pp. 117-128; 1956.

(Some elements can be assembled into machines that can realize arbitrary functions within reasonable restrictions.)

79. Moore, E.F., "Gedanken Experiments on Sequential Machines", Automata Studies, (Annals of Math. Studies No. 34) Princeton Univ. Press, pp. 129-153; 1956.

(Investigation of the characteristics of sequential machines by means of experiments, (applying inputs and observing outputs).

80. Muller, D.E., "A Theory of Asynchronous Circuits", Proc. Internatl. Symp. Theory of Switching, Harvard Univ. Cambridge, Mass., Part 1, pp. 204-243; April 1957.

81. Myhill, J., "Finite Automata and Representation of Events", WADC Tech. Rept. 57-624; 1957.

(A certain class of devices known as finite automata is discussed with the aid of algebraic techniques.)

82. Myhill, J., "Linear Bounded Automata", U.S. Govt. Res. Repts., vol. 35, p. 324(A); March 10, 1961.

(A class of automata, linear bounded automata, which can do more than a finite automaton but less than a Turing machine, is presented.)

83. Narasimhan, R., "Minimizing Incompletely Specified Sequential Switching Functions", IRE Trans. on Elec. Comp., vol. EC-10, pp. 531-532; 1961.

84. Nerode, A., "Linear Automata Transformations", Proc. Am. Math. Soc., vol. 9, pp. 541-544; 1958.

85. Netherwood, D.B., "Minimal Sequential Machines", IRE Trans. on Elec. Comp., vol. EC-8, pp. 339-345; 1959.

86. Ott, G.H. and Feinstein, N.H., "Design of Sequential Machines from Their Regular Expressions", J. Assoc. Comp. Mach., vol. 8, pp. 585-600; Oct. 1961.

(The notion of improper state diagram is introduced. Algorithms are also given for the synthesis of such a state diagram from regular expressions.)

87. Paull, M.C. and Unger, S.H., "Minimizing the Number of States in Incompletely Specified Sequential Switching Functions", IRE Trans. on Elec. Comp., vol. EC-8, pp. 356-367; 1959.

(An analysis of the problem is presented and a partially enumerative solution is evolved.)

88. Perles, M., Rabin, M.O. and Shamir, E., "The Theory of Definite Automata", U.S.Govt. Res. Repts., vol. 37, pp. 35-36(A); Jan. 5, 1962.

(The notion of a definite event introduced by Kleene and the related concepts of definite automata and tables are discussed.)

89. Rabin, M.O., and Scott, D., "Finite Automata and Their Decision Problems", IBM J. Res. & Dev., vol. 3, pp. 114-125; April 1959.
Review: IRE Trans. on Elec. Comp., vol. EC-8, no. 3; Sept. 1959.

(The theory of finite automata, divided into one-tape, one-way, two-way and multi-tape, is discussed. Finite automata are treated as subclass of Turing machines with a finite number of internal states and finite tapes.)

90. Raney, G.N., "Sequential Functions". J. Assoc. Comp. Mach., vol. 5, no. 2, pp. 117-180; April 1958.

(Similar to Mealy's notation. Associate the concept of state with the function rather than with the structure of the sequential machine.)

91. Reed, I.S., "Mathematical Structure of Sequential Machines", in A Survey of Switching Circuit Theory, E.J. McCluskey and T.C. Bartee (eds.), pp. 187-196, McGraw-Hill Book Company, Inc., New York, 1962.

92. Rubinoff, M., "Remarks on Design of Sequential Circuits", Proc. Internatl. Symp. on the Theory of Switching, April 2-5, 1957, (Annals of the Computation Lab., Harvard Univ., Cambridge, Mass., vol. 30, pp. 241-280; 1959.) Review: IRE Trans. on Elec. Comp., vol. EC-9, no. 3, p. 380; 1960.

(Describes design of sequential circuits from the viewpoint of a Turing machine. The machine may write a symbol, it may change its state and then it may look at either the symbol it has just written or one of the neighboring symbols.)

93. Subert, E.J., "Matrix Algebra for Sequential Logic", AIEE Trans., vol. 78, Part 1, pp. 1074-1079; 1960.

94. Schützenberger, M.P., "A Remark on Finite Transducers", Information and Control, vol. 4, pp. 185-196; Sept. 1961. Review: IRE Trans. on Elec. Comp., vol. EC-11, p. 802; Dec. 1962.

(Few properties of transduction are discussed.)

95. Schützenberger, M.P., "On the Definition of a Family of Automata", Information and Control, vol. 4, pp. 245-270, 1961.

(The definition of a family A of automata derived from the family A_o of the finite one-way one-tape automata (Rabin & Scott).)

96. Seshu, S., "Mathematical Models for Sequential Machines", IRE Natl. Conv. Record, vol. 7, Part 2, pp. 4-16; 1959.

(A intuitive yet unified description of sequential machines.)

97. Seshu, S., Miller, R.E. and Metze, G., "Transition Matrices of Sequential Machines", IRE Trans. on Circuit Theory, vol. CT-6, no. 1; March 1959. Review: IRE Trans. on Elec. Comp., vol. EC-8, no. 4; Dec. 1959.

(A matrix formulation of sequential machines due to Moore's model.)

98. Shannon, C.E., "Computers and Automata", Proc. IRE, vol. 41, no. 10, pp. 1234-1241; Oct. 1953.

(A brief survey of development in the field of automata and nonnumerical computation.)

99. Shepherdson, J.C., "The Reduction of Two-Way Automata to One-Way Automata", IBM Jour. of Res. & Dev., vol. 3, pp. 198-200; April 1959.

(A shorter, more direct proof is presented of Rabin's result that finite 2-way automata are equivalent to one-way automata, as far as the classification of input tapes is concerned.)

100. Srinivasan, C.V. and Narasimhan, R., "On the Synthesis of Finite Sequential Machines", Proc. Indian Acad. Sci., vol. 50, pp. 68-82; 1959.

(A procedure is given to obtain a minimal machine behavior in terms of events and corresponding output states.)

101. Stearns, R.E. and Hartmanis, J., "On the State Assignment Problem for Sequential Machines, II", IRE Trans. on Electronic Comp., vol. EC-10, pp. 593-603; Dec. 1961.

102. Turing, A.M., "On Computable Numbers with an Application to the Entschiedungs Problem", Proc. London Math. Soc.Ser 2, vol. 42, pp. 230-265; 1936 and vol. 43, pp. 544-546; 1937.

(The idea of Turing machines is introduced.)

103. Wang, H., "A Variant to Turing's Theory of Computing Machines", J. Assoc. Comp. Mach, vol. 4, no. 1, pp. 63-92; 1957.

(A variation of Turing machine is discussed.)

104. Wang, H., "Circuit Synthesis by Solving Sequential Boolean Equations", Zeitschrift für Mathematische Logik und Grundlagen die Mathematik, vol. 5, pp. 291-322; 1959. Review: IRE Trans. Elec. Comp., vol. EC-10, no. 3; Sept. 1961.

105. Weeg, G.P., "The Structure of an Automaton and Its Operation-Preserving Transformation Group", J. Assoc. Comp. Mach., vol. 9, no. 3, pp. 345-349; July 1962.

(The group of operation-preserving transformations of a strongly connected automaton onto itself is isomorphic to a group of subsets of input sequences under a certain operation.)

106. Winett, J.M., "An Alpha-State Finite Automaton for Multiplication by Alpha", IRE Trans. on Elec. Comp., vol. EC-11, pp. 412-413; June 1962.

(Considers the construction of a finite automaton which, when presented with an arbitrary number N, will produce as the output αN where α is a given fixed number (integer).)

107. Yamada, H., "Disjunctively Linear Logic Nets", IRE Trans. on Elec. Comp., vol. EC-11, no. 5; pp. 623-638; Oct. 1962.

(A class of logic nets with disjunctively linear structure is discussed.)

108. Yoeli, M., "The Cascade Decomposition of Sequential Machines", IRE Trans. on Elec. Comp., vol. EC-10, pp. 587-592; 1961. Review: IRE Trans. Elec. Comp., vol. EC-11; April 1962.

(A necessary and sufficient condition is given for a machine to be decomposed into two smaller machines operating in cascade.)

Distribution list as of February 1, 1964

1 Director
Air University Library
Maxwell Air Force Base, Alabama
Attn: CR-4803a

1 Redstone Scientific Information Center
U.S. Army Missile Command
Redstone Arsenal, Alabama

1 Electronics Research Laboratory (Unclassified)
University of California
Berkeley 4, California

2 Hughes Aircraft Company
Florence and Teale
Culver City, California
Attn: N. E. Devereux
Technical Document Center

3 Autonetics (Unclassified)
9150 East Imperial Highway
Downey, California
Attn: Tech. Library, 3041-11

1 Dr. Arnold T. Nordsieck
General Motors Corporation
Defense Research Laboratories
6767 Hollister Avenue
Goleta, California

1 University of California (Unclassified)
Lawrence Radiation Laboratory
P. O. Box 808
Livermore, California

1 Mr. Thomas L. Hartwick
Aerospace Corporation
P. O. Box 95085
Los Angeles 45, California

1 Lt. Colonel Willard Levin
Aerospace Corporation
P. O. Box 95085
Los Angeles 45, California

1 Professor Zorab Kaprelian
University of Southern California
University Park
Los Angeles 7, California

1 Sylvania Electronic Systems - West
Electronic Defense Laboratories
P. O. Box 205
Mountain View, California
Attn: Documents Center

1 Varian Associates
611 Hansen Way
Palo Alto, California
Attn: Dr. Ira Weissman

1 Huston Denslow (Unclassified)
Library Supervisor
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

1 Professor Nicholas George
California Institute of Technology
Electrical Engineering Department
Pasadena, California

1 Space Technology Labs., Inc.
One Space Park
Redondo Beach, California
Attn: Acquisitions Group
STL Technical Library

2 Commanding Officer and Director
U.S. Naval Electronics Laboratory
San Diego 52, California
Attn: Code 2800, C. S. Manning

1 Commanding Officer and Director
U.S. Navy Electronics Laboratory
San Diego 52, California
Attn: Library

1 Office of Naval Research Branch Office
1000 Geary Street
San Francisco, California

1 The RAND Corporation
1700 Main Street
Santa Monica, California
Attn: Library

1 Stanford Electronics Laboratories (Unclassified)
Stanford University
Stanford, California
Attn: SEL Documents Librarian

1 Dr. L. F. Carter
Chief Scientist Air Force
Room 4E-324, Pentagon
Washington 25, D. C.

1 Mr. Robert L. Feik
Associate Director for Research
Research and Technology Division
AFSC
Bolling Air Force Base 25, D. C.

1 Captain Paul Johnson (USN-Ret)
National Aeronautics and Space
Administration
1520 H Street, N. W.
Washington 25, D. C.

1 Major Edwin M. Myers
Headquarters USAF (AFRDR)
Washington 25, D. C.

1 Dr. James Ward
Office of Deputy Director
(Research and Info)
Department of Defense
Washington 25, D. C.

1 Dr. Alan T. Waterman
Director, National Science Foundation
Washington 25, D. C.

1 Mr. G. D. Watson
Defense Research Member
Canadian Joint Staff
2450 Massachusetts Ave., N. W.
Washington 8, D. C.

1 Mr. Arthur G. Wimer
Chief Scientist
Air Force Systems Command
Andrews Air Force Base
Washington 25, D. C.

1 Director, Advanced Research
Projects Agency
Washington 25, D. C.

1 Air Force Office of Scientific Branch
Directorate of Engineering Sciences
Washington 25, D. C.
Attn: Electronics Division

1 Director of Science and Technology
Headquarters, USAF
Washington 25, D. C.
Attn: AFRST-EL/GU

1 AFRST - SC
Headquarters, USAF
Washington 25, D. C.

1 Headquarters, R & T Division (Unclassified)
Bolling Air Force Base
Washington 25, D. C.
Attn: RTHR

1 Headquarters, U. S. Army Material Command
Research Division, R & D Directorate
Washington 25, D. C.
Attn: Physics & Electronics Branch
Electronics Section

1 Commanding Officer
Diamond Ordnance Fuze Laboratories
Washington 25, D. C.
Attn: Librarian, Room 211, Bldg. 92

1 Operation Evaluation Group
Chief of Naval Operations (OP-03EG)
Department of Navy
Washington, D. C. 20350

1 Chief of Naval Operations (Code OP-ONT)
Department of the Navy
Washington, D. C. 20350

1 Commanding Officer
U. S. Army Personnel Research Office
Washington 25, D. C.

1 Commanding Officer & Director
Code 142 Library
David W. Taylor Model Basin
Washington, D. C. 20007

1 Chief, Bureau of Ships (Code 686)
Department of the Navy
Washington, D. C. 20360

1 Chief, Bureau of Ships (Code 732)
Department of the Navy
Washington, D. C. 20360

1 Chief, Bureau of Naval Weapons
Technical Library, DLI-3
Department of the Navy
Washington, D. C. 20360

1 Director, (Code 5140)
U. S. Naval Research Laboratory
Washington, D. C. 20390

1 Chief of Naval Research (Code 437)
Department of the Navy
Washington, D. C. 20360

1 Dr. H. Wallace Sinaiko (Unclassified)
Institute for Defense Analyses
Research & Engineering Support Division
1666 Connecticut Ave., N. W.
Washington 9, D. C.

1 Data Processing Systems Division
National Bureau of Standards
Conn. at Van Ness
Room 239, Bldg. 10
Washington 25, D. C.
Attn: A. K. Smilow

1 National Bureau of Standards (Unclassified)
Research Information Center &
Advisory Service on Information
Processing
Data Processing Systems Division
Washington 25, D. C.

1 Exchange and Gift Division (Unclassified)
The Library of Congress
Washington 25, D. C.

1 NASA Headquarters
Office of Applications
400 Maryland Avenue, S. W.
Washington 25, D. C.
Attn: Mr. A. M. Greg Andrus
Code FC

1 APGC (PGAPI)
Eglin Air Force Base
Florida

1 Martin Company
P. O. Box 5837
Orlando, Florida
Attn: Engineering Library
MP-30

1 Commanding Officer
Office of Naval Research, Chicago Branch
6th Floor, 230 North Michigan
Chicago 1, Illinois

1 Laboratories for Applied Sciences
University of Chicago
6220 South Drexel
Chicago 37, Illinois

1 Librarian (Unclassified)
School of Electrical Engineering
Purdue University
Lafayette, Indiana

1 Donald L. Epley (Unclassified)
Department of Electrical Engineering
State University of Iowa
Iowa City, Iowa

1 Commanding Officer
U. S. Army Medical Research Laboratory
Fort Knox, Kentucky

2 Keats A. Pullen, Jr.
Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland

1 Director
U. S. Army Human Engineering Laboratories
Aberdeen Proving Ground, Maryland

1 Mr. James Tippett
National Security Agency
Fort Meade, Maryland

1 Commander
Air Force Cambridge Research Laboratories
Laurence G. Hanscom Field
Bedford, Massachusetts

1 Dr. Lloyd Hollingsworth
Director, ERD
AFCRL
L. G. Hanscom Field
Bedford, Massachusetts

1 Data Sciences Laboratory
Air Force Cambridge Research Lab
Office of Aerospace Research, USAF
L. G. Hanscom Field
Bedford, Massachusetts
Attn: Lt. Stephen J. Kahne - CRB

1 Instrumentation Laboratory(Unclassified)
Massachusetts Institute of Technology
68 Albany Street
Cambridge 39, Massachusetts

1 Research Laboratory of Electronics(Unclassifi
Massachusetts Institute of Technology
Cambridge 39, Massachusetts
Attn: Document Room 26-327

1 Dr. Robert Kingston
Lincoln Laboratories
Lexington, Massachusetts

1 Lincoln Laboratory (Unclassified)
Massachusetts Institute of Technology
P. O. Box 73
Lexington 73, Massachusetts
Attn: Library, A-082

1 Sylvania Electric Products, Inc.
Electronic Systems
Waltham Labs. Library
100 First Avenue
Waltham 54, Massachusetts

1 Minneapolis-Honeywell Regulator Co.
Aeronautical Division (Unclassified)
2600 Ridgeway Road
Minneapolis 13, Minnesota
Attn: Dr. D. F. Elwell
Main Station: 625

1 Inspector of Naval Material
Bureau of Ships Technical Representative
1902 West Minnehaha Avenue
St. Paul 4, Minnesota

20 Activity Supply Officer, USAELRDL
Building 2504, Charles Wood Area
Fort Monmouth, New Jersey
For: Accountable Property Officer
Marked: For Inst. for Exploratory Research
Inspect at Destination
Order No. 576-PM-63-91

1 Commanding General
U. S. Army Electronic Command
Fort Monmouth, New Jersey
Attn: AMSEL-RE

1 Mr. A. A. Lundstrom
Bell Telephone Laboratories
Room 2E-127
Whippany Road
Whippany, New Jersey

1 AFMDC (MDSGP/Capt. Wright)
Holloman Air Force Base
New Mexico

1 Commanding General
White Sands Missile Range
New Mexico

1 Microwave Research Institute
Polytechnic Institute of Brooklyn
55 John Street (Unclassified)
Brooklyn 1, New York

1 Cornell Aeronautical Laboratory, Inc.
4455 Genesee Street
Buffalo 21, New York
Attn: J. P. Desmond, Librarian

1 Sperry Gyroscope Company (Unclassified)
Marine Division Library
155 Glen Cove Road
Carle Place, L. I., New York
Attn: Mrs. Barbara Judd

1 Major William Harris
RADC (RAWI)
Griffiss Air Force Base
New York

1 Rome Air Development Center
Griffiss Air Force Base
New York
Attn: Documents Library
RAALD

1 Library (Unclassified)
Light Military Electronics Department
General Electric Company
Armament & Control Products Section
Johnson City, New York

1 Columbia Radiation Laboratory
Columbia University (Unclassified)
538 West 120th Street
New York 57, New York

1 Mr. Alan Barnum
Rome Air Development Center
Griffiss Air Force Base
Rome, New York

1 Dr. E. Howard Holt (Unclassified)
Director
Plasma Research Laboratory
Rensselaer Polytechnic Institute
Troy, New York

3 Commanding Officer
U.S. Army Research Office (Durham)
Box CM, Duke Station
Durham, North Carolina
Attn: CRD-AA-1P, Mr. Ulsh

1 Battelle-DEFENDER
Battelle Memorial Institute
505 King Avenue
Columbus 1, Ohio

1 Aeronautical Systems Division
Navigation and Guidance Laboratory
Wright-Patterson Air Force Base
Ohio

1 Aeronautical Systems Division
Directorate of Systems Dynamic Analysis
Wright-Patterson Air Force Base
Ohio

1 Commander
Research & Technology Div.
Wright-Patterson Air Force Base
Ohio 45433
Attn: MAYT (Mr. Evans)

1 Commanding Officer (AD-5)
U.S. Naval Air Development Center
Johnsville, Pennsylvania
Attn: NADC Library

2 Commanding Officer
Frankford Arsenal
Philadelphia 37, Pennsylvania
Attn: SMUFA-1300

1 H. E. Cochran
Oak Ridge National Laboratory
P. O. Box X
Oak Ridge, Tennessee

1 U.S. Atomic Energy Commission
Office of Technical Information Extension
P. O. Box 62
Oak Ridge, Tennessee

1 President
U.S. Army Air Defense Board
Fort Bliss, Texas

1 U.S. Air Force Security Service
San Antonio, Texas
Attn: ODC-R

1 Director
Human Resources Research Office
The George Washington University
300 North Washington Street
Alexandria, Virginia

20 Defense Documentation Center
Cameron Station
Alexandria, Virginia
22314

1 Commander
U. S. Army Research Office
Highland Building
3045 Columbia Pike
Arlington 4, Virginia

1 U.S. Naval Weapons Laboratory
Computation and Analysis Laboratory
Dahlgren, Virginia
Attn: Mr. Ralph A. Niemann

2 Army Material Command
Research Division
R & D Directorate
Bldg. T-7
Gravelley Point, Virginia