STRUCTURE OF AUTOMATA

Thesis for the Degree of Ph. D.

MICHIGAN STATE UNIVERSITY

Bruce Herbert Barnes

1960

This is to certify that the

thesis entitled

STRUCTURE OF AUTOMATA

presented by

Bruce Herbert Barnes

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Mathematics

[signature]
Major professor

Date November 22, 1960

O-169

STRUCTURE OF AUTOMATA

By

BRUCE HERBERT BARNES

AN ABSTRACT

Submitted to the School for Advanced Graduate Studies of Michigan State University of Agriculture and Applied Science in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

1960

Approved ______________________

ABSTRACT

This dissertation deals with the structure of finite automata and sequential machines. The first chapter is an extension of Rabin and Scott's work on finite automata. The sets of tapes acceptable to a finite automaton are characterized through the use of equivalence classes. This is done for four types of finite automata. These are:

(1) One initial state and not all states final.

(2) More than one initial state and not all states final.

(3) One initial state and all states final.

(4) All states both initial and final.

In the second chapter sets of input-output sequences that are acceptable to a sequential machine are characterized through the use of equivalence classes similar to those employed for finite automata. It is shown how the set of input-output sequences acceptable to a sequential machine can be obtained from the characterization of the set of tapes acceptable to a particular state of the sequential machine. This characterization is used to prove some theorems concerning the reduction of sequential machines to minimal state form.

In Chapter III the structure of finite automata and sequential machines is studied through the use of connection matrices. The properties of a positive connection matrix which distinguishes it from a strongly connected matrix are discussed. It is shown that all the powers of a positive connection matrix are strongly connected, but the same statement for strongly connected matrices is

not true. This condition is reduced to just the first n powers of an n x n connection matrix need be strongly connected to insure that a connection matrix is positive.

It is also pointed out that the theorems of Chapter III apply to primitive matrices of non-negative real numbers.

STRUCTURE OF AUTOMATA

By

BRUCE HERBERT BARNES

A THESIS

Submitted to the School for Advanced Graduate Studies of Michigan State University of Agriculture and Applied Science in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

1960

Approved ____________________

ACKNOWLEDGMENT

The author wishes to express his gratitude to Professor Gerard P. Weeg, for his guidance and encouragement throughout the writing of this paper and during most of his graduate study.

TABLE OF CONTENTS

INTRODUCTION

The investigation into the theory of sequential machines have had a very recent origin. It was in 1955 and 1956 that Mealy [22], Moore [23] and Huffman [17], published what are evidently the pioneering papers in the field. These authors, and most later ones, consider a sequential machine to be a device capable of assuming any one of a finite set of internal states, such that when an input symbol is presented to the machine, an output symbol is produced by the machine and the sequential machine assumes another state.

There are three major areas in the study of sequential machines. These are:

(1) The construction of electronic devices, which have properties similar to those mentioned above. Such devices are commonly called sequential transducers. Considerable work has been done in the area in recent years, especially since the advent of high speed digital computers and other types of process control apparatus where the process is sequential in nature. Huffman [17], Cadden [6], Unger [26] and others have done a considerable amount of work in this field.

(2) The synthesis and analysis of state diagrams (these are weighted graphs used to represent sequential machine) which are used to facilitate the construction of sequential machines. This area has been dealt with by Huffman [17], Mealy [22], Hohn [3], Aufenkamp [1, 3, 4], Ginsburg [10, 11, 12, 13], Bellman [5], Gillespie [1] and others.

(3) The study of sequential machines as a mathematical system. Kleene [19], Moore [23], Nerode [24], Rabin and Scott [25], Ginsburg [9], Weeg [18, 27] and several others have worked in this field.

This last area of sequential machines is the subject of this dissertation. Three separate but related topics will be covered in this dissertation. They are finite automata, sequential machines and connection matrices.

A finite automaton is a sequential machine without outputs. The set of sequences (called tapes) which are suitable for a finite automaton are characterized through the use of equivalence classes. Four types of finite automata are discussed in Chapter I.

The major result of the first chapter is that a set of tapes, U, is the acceptable set of tapes for a finite automaton, in which each state is an initial state and also each state is a final state, if and only if U is complete (that is, if x is any tape in U, then any portion of x is in U) and is the union of all but one of the equivalence classes of a particular right invariant equivalence relation of finite index.

In Chapter II, the set of sequences of input-output symbols (called I-0 sequences) which are suitable for a sequential machine are characterized by the use of equivalence classes. These equivalence classes are used to prove some theorems concerning special classes of sequential machines. The main theorem of this chapter is similar to that of Chapter I, except, due to the outputs, the conditions are more stringent.

For the purpose of study of the structure of sequential machines a connection matrix is often used. A special class of these is the class of strongly connected connection matrices. A subclass of the class of strongly connected sequential machines is that of positive machines. The properties which distinguish this subclass from the class of strongly connected connection matrices are discussed in Chapter III. The major distinction is that a positive connection matrix has all of its powers strongly connected. It is shown that this requirement can be reduced to the point that just the first n powers of an n x n connection matrix need be strongly connected to insure positiveness.

Chapter I

FINITE AUTOMATA

An automaton may be thought of as a black box which will accept tapes (questions). As the tape proceeds through the box the internal mechanisms of the box assume different configurations and when the tape is completely accepted by the black box an answer is given (yes or no). This answer depends on the configuration of the internal mechanism of the black box. The method of giving the answer might be by means of a light, which is on when the configuration of the internal mechanism corresponds to an answer of yes.

A tape (question) is called acceptable to a finite automaton if the answer corresponding to this question is yes. Several interesting questions arise concerning acceptable sets of tapes. Among these are:

(1) What are the properties of a set of acceptable tapes?

(2) For every set of tapes U is there an automaton having U as its set of acceptable tapes?

(3) Given a set of tapes, is there an effective procedure to ascertain if there exists an automaton accepting this set of tapes, and if so, can this automaton be produced?

These are the problems that are discussed in this chapter. Before these ideas are formalized, however, a few definitions will be needed. These definitions will largely be similar to those found in the literature.

Definition 1. A tape x is a finite sequence of elements from a non-empty finite set Σ. The elements of Σ are called tape symbols and Σ is called the alphabet. The null tape (i. e., the tape with no symbols) is denoted by λ.

Definition 2. The set of all finite tapes over the alphabet Σ is denoted by T. A tape x is written in the form $x = \sigma_0\sigma_1 \ . \ . \ . \ \sigma_n$ where σ_i are elements of Σ. If $x = \sigma_0 \ . \ . \ . \ \sigma_{n-1}$ and $y = a_0 \ . \ . \ . \ a_{m-1}$ then by xy is meant the tape $xy = \sigma_0 \ . \ . \ . \ \sigma_{n-1}a_0 \ . \ . \ . \ a_{m-1}$. A portion of tape x is denoted by ${}_ix_j$, which means that portion of the tape x beginning with the i^{th} position and continuing up to and including the j-1 position. By ${}_ix_i$ is meant the null tape.

Definition 3, [25, p. 116]. A finite automaton of type 1, also called an automaton when no confusion will result, over the alphabet Σ is a system $A^* = (S, M, s_0, F)$ where S is a finite non-empty set (called the set of internal states of A), M is a function defined on the Cartesian product $S \times \Sigma$ of all ordered pairs of states and input symbols with values in S (called the table of transitions or moves of A*), s_0 is an element of S (called the initial state of A*) and F is a non-empty subset of S (the set of designated final states of A*).

There are several methods for specifying a finite automaton. One common method is to give a table, called the table of moves, and a list of final states.

Example: Let $A^* = (S, M, s_0, F)$ be defined over the alphabet $\Sigma = \{a, b, c\}$ in the following way:

$S = \{s_0, s_1, s_2\}$, with s_0 the initial state.

$F = \{S_2\}$ the set containing only s_2.

M is defined by Table 1, with the following interpretation.

If s_i is the present state, σ_i the present input and s_n the next state, then $M(s_i, \sigma_j) = s_n$.

Table 1

Present State	Next State Present Input a	 b	 c
s_0	s_1	s_2	s_2
s_1	s_2	s_1	s_0
s_2	s_0	s_1	s_2

For the purposes of analysis of an automaton, a pictorial display is often useful. This is usually given in the form of a state diagram which parallels Moore's transition diagram of a sequential machine. A state diagram is a weighted directed graph with each vertex corresponding to a state of the automaton. The states are usually drawn as circles with an ordered pair (a, b) inside the circle, where a is the number of the state and b is 1, if this state is final, and 0, if this state is not a final state.

If $M(s_k, \sigma_i) = s_e$, then there is a directed line segment from vertex k to vertex e labelled σ_i. Figure 1 is the state diagram for the previous example.

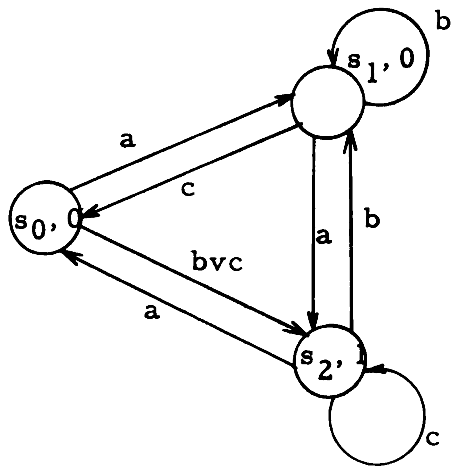

Figure 1

In the remainder of this chapter the given definition of an automaton will be modified and its properties compared with those given by Rabin and Scott [25]. In the Rabin and Scott type automaton $M(s_i, \sigma_j)$ is defined for all combinations of s_i in S and σ_j in Σ. This property will now be weakened by requiring that transitions be defined only for some state-input combinations, but not necessarily for all state-input combinations.

Definition 4. A finite automaton of type 2 over the alphabet Σ is a system $A = (S, M, s_0, F)$ where S is a finite non-empty set (called the internal states of A), M is a function defined on a subset L of the Cartesian product $S \times \Sigma$ of all pairs of states and input symbols into the set of internal states S, s_0 is an element of S (called the initial state) and F is a non-empty subset of S (called the set of final states of S).

Henceforth, the symbol A* will be used for a Rabin and Scott type automaton and A for a machine as defined according to Definition 4.

It is necessary for many of the proofs to extend the definition of M from L, a subset of $S \times \Sigma$, to H, a subset of $S \times T$. This is accomplished in a manner similar to that of Rabin and Scott [25]:

$$M(s_i, \lambda) = s_i$$
$$M(s_i, x\sigma_j) = M(M(s_i, x), \sigma_j)$$

for s_i in S, x in T, and σ_j in Σ. If $M(s_i, x)$ is not defined or $M(M(s_i, x), \sigma_j)$ is not defined, then $M(s_i, x\sigma_j)$ is not defined.

Definition 5. A tape x is called an acceptable tape for an automaton $A = (S, M, s_0, F)$ of type 2 if $M(s_0, x)$ is contained in F. The set of all tapes which are acceptable to an automaton A of type 2 is denoted by T(A), and is called the set of acceptable tapes for the automaton A.

Definition 6. The set $\mathcal{T}$ is the set of sets of tapes over the alphabet Σ such that U is in $\mathcal{T}$, if and only if U = T(A) for some automaton A over Σ, of type 2.

Theorem 1. If U is a set of tapes such that U = T(A), for some automaton A of type 2, then there exists an automaton A* of type 1 such that U = T(A*) and conversely.

Proof: If U = T(A), then the automaton A* is constructable from the automaton A. This is done in the following manner: Let $S^* = S \cup \{s'\}$ where S is the set of states for the automaton A and $\{s'\}$ is the set consisting of s' alone where s' is distinct from any state in S. Define:

$M^*(s_i, \sigma_j) = M(s_i, \sigma_j)$ if $M(s_i, \sigma_j)$ is defined.

$M^*(s_i, \sigma_j) = s'$ if $M(s_i, \sigma_i)$ is not defined and $M^*(s', \sigma_i) = s'$ for all σ_i contained in $\sum$.

$s^* = s_0$ and $F^* = F$.

For any tape x in T(A), x is also in T(A*), because $M^*(s_0, x) = M(s_0, x)$ and if $M(s_0, x)$ is in F, $M^*(s_0, x)$ is in F. Likewise, if any tape y is not in T(A), then it is also not in T(A*), for if $M(s_0, y)$ is defined, $M^*(s_0, y) = M(s_0, y)$ which is not in F and thus not in F*; while if $M(s_0, y)$ is not defined then $M^*(s_0, y) = s'$ which is not in F*.

One can also observe that an automaton A* of type 1 is also an automaton A of type 2. We have, therefore, shown that U = T(A) if and only if U = T(A*) for some A and A* of type 2 and 1 respectively.

The important difference between the two kinds of automata is that an automaton A probably would not require as many transitions and possibly even fewer states than the automaton A*. This can be seen from the following example.

Let U = $\{10101\}$. The automaton A of type 2 of figure 2 has U as its set of acceptable tapes.

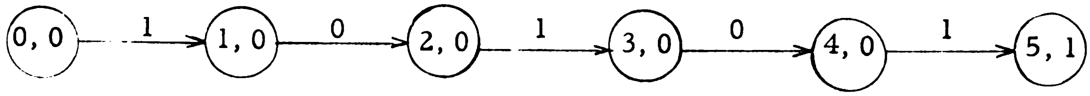

Figure 2

The Rabin and Scott type automaton A* of figure 3 also has U as its set of acceptable tapes. As can be seen the automaton A has one less state than the automaton A* and 9 less transitions.

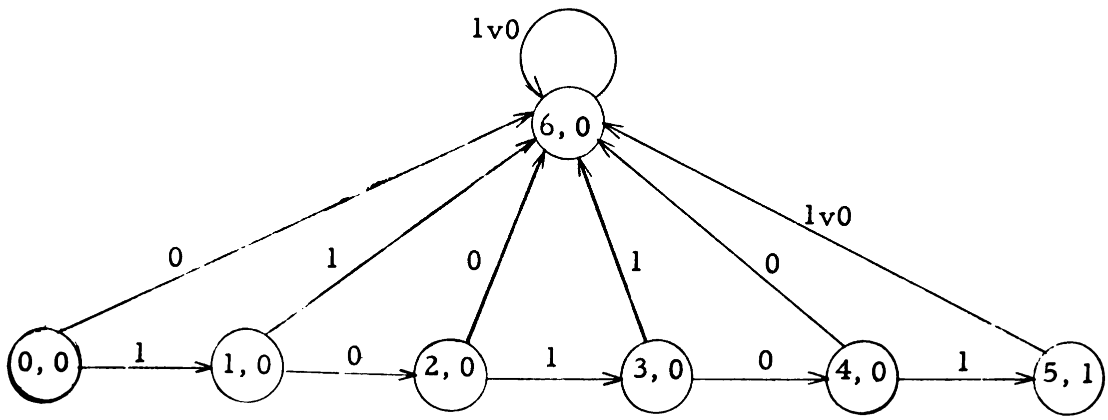

Figure 3

Not every set U of tapes is a set of acceptable tapes for some automaton. This has been shown by Rabin and Scott [25]. Their proof, however, produced a set of tapes not acceptable to any automaton A*, while the proof of Theorem 2 is an existence proof.

Lemma 1. The cardinal number of the set $\mathcal{T}$ is $\aleph_0$.

Proof: Let c denote the cardinality of Σ. If A = (S, M, s_0, F) has n states, then for each state s_i, $M(s_i, \sigma_j)$ can be any one of n states or not be defined. Thus, there are $(n+1)^c$ possible ways to define the function M for each state and for any fixed n there are at most $[(n+1)^c]^n$ possible ways to define the function M. Since the number of non-empty subsets F of S is 2^n-1, and s_0 can be any one of n states, there are $n(2^n-1)[(n+1)^c]^n$ possible machines with exactly n states. A countable number of sets, each of which contains all possible finite automata with a given number of states, can be formed. Hence, there are at most $\aleph_0$ finite automata and at most $\aleph_0$ sets of acceptable sets of tapes over the alphabet Σ. Let S be a set of n states. Let s_1 be the initial state and F the set containing only the state s_n. Let $M(s_i, \sigma_j) = s_{i+1}$ for $1 \leq i \leq n$ and let σ_i be undefined for all i except 1, then this automaton accepts the tape $\underbrace{\sigma_1\sigma_1 \cdots \sigma_1}_{n}$ and only this tape. In this way $\aleph_0$ distinct sets of acceptable tapes are produced. This shows that the cardinality of $\mathcal{T}$ is $\aleph_0$.

Theorem 2. There exist sets U of tapes, such that U is not the set of acceptable tapes for any automaton A, of type 2.

Proof: Since the cardinality of T is $\aleph_0$ there are $2^{\aleph_0} = \mathfrak{c}$ sets of sets of tapes over the alphabet Σ. Thus making use of Lemma 1, there exist sets of tapes which are not acceptable to any finite automaton.

Rabin and Scott [25] give several theorems concerning acceptable sets of tape, which are due to Myhill, Nerode, and themselves. We will now generalize these theorems. Many of these theorems make strong use of equivalence relations among the tapes of T.

Definition 7. An equivalence relation R over the set T of tapes is right invariant, if whenever xRy, then xzRyz for all z in T. There is also an analogous definition for a left invariant equivalence relation.

Definition 8. An equivalence relation over the set T is a congruence relation if it is both left and right invariant.

Definition 9. An equivalence relation over T is of finite index if there are only finitely many equivalence classes under the relation.

With these definitions available it is now possible to state the following theorem, which is due to Myhill [25, p. 117], and prove its applicability to an automaton A of type 2.

Theorem 3. Let U be a set of tapes over the alphabet Σ. The following three conditions are equivalent.

(1) U is in $\mathcal{T}$.

(2) U is the union of some of the equivalence classes of a congruence relation over T of finite index.

(3) The explicit congruence relation $\equiv$ defined by the condition that for all x, y in T, $x \equiv y$ if and only if for all z, w in T, whenever

zxw is in U, then zyw is in U, and conversely, is a congruence relation of finite index.

Proof: As pointed out in Theorem 1, for every automaton A of type 2, there is an automaton A* of type 1 which has the same set of acceptable tapes. Thus, if T(A) is in $\mathcal{T}$, so is T(A*) and both sets have the same properties.

The following theorem, which is a generalization of a theorem due to Nerode [24, p. 543], applies to automata A* of type 1, hence, to automata A of type 2.

Theorem 4. Let U be a set of tapes. The following three conditions are equivalent:

(1) U is in $\mathcal{T}$.

(2) U is the union of some of the equivalence classes of a right invariant equivalence relation over T of finite index.

(3) The explicit right invariant equivalence relation E defined by the condition that for all x, y in T, xEy if and only if for all z in T, whenever xz is in U, then yz is in U and conversely, is an equivalence relation of finite index.

Corollary 1. Let U be in $\mathcal{T}$. If the number of equivalence classes of T under the relation E is n, then the least number of states in any automaton having U as its set of acceptable tapes is n-p, where p is 1 if there exists an equivalence class [y], such that for any x in [y], xz is not in U for all z in T; otherwise p is 0.

Proof: Let each equivalence class denote a state. Let s_0 = [λ] (the equivalence class containing the null tape). Let F be the set of equivalence classes which contain a tape from U. Define M as follows:

$$M([x], \sigma_j) = [x\sigma_j].$$

Since $M([\lambda], x) = M(s_0, x) = [x]$, this automaton has U as its acceptable set of tapes. If one of the equivalence classes does not contain a tape y such that yz is in U, for some tape z in T, it is not necessary to have a state corresponding to this equivalence class in this automaton. Thus, this state and all the transitions emanating from it and terminating in it can be removed. This can be seen from the following argument.

If $x \not{E} y$ then $M(s_0, x) \neq M(s_0, y)$ for any automaton A of type 2. We have therefore, that for each equivalence class, which contains some tape x such that $M(s_0, x)$ is defined, a distinct internal state in the machine. If we define $M([x], \sigma) = [x\sigma]$ where $[x]$ and $[x\sigma]$ are equivalence classes containing a tape y such that $M(s_0, y)$ is defined, we produce an automaton which accepts the set of tapes U, and which has exactly as many states as there are equivalence classes under E which contain a tape for which $M(s_0, x)$ is defined.

The above corollary is a generalization of Nerode's theorem [25, p. 118]. However, it is interesting to note that with the generalized definition of an automaton there may be one less state needed to produce an automaton which will accept the set of tapes U.

The next three theorems by Rabin and Scott depend only on the equivalence relation and are, therefore, also true for the automata A of type 2.

Theorem 5. If x is in T, then $\{x\}$, the set consisting of x alone is in $\mathcal{T}$.

Definition 10. If x is the tape $\sigma_1\sigma_2 \ldots \sigma_n$, then x* is the tape $\sigma_n\sigma_{n-1} \ldots \sigma_2\sigma_1$. If U is a set of tapes then U* is the set of tapes such that x is in U if and only if x* is in U*.

Theorem 6. If U is in $\mathcal{T}$, then U* is in $\mathcal{T}$.

Theorem 7. The class $\mathcal{T}$ is a Boolean algebra of sets.

In the remainder of the chapter we will generalize the idea of an automaton and acceptability of tapes. Through an evolution of automata we will arrive at an automaton in which all states are initial states and all states are final states, with a tape being acceptable if it "reads" through the "reader". This corresponds to the more generally accepted idea of an automaton and also forms a good foundation for the next chapter on sequential machine.

Definition 11. An automaton B of type 3 over the alphabet Σ is a system B = (S, M, Q, F) in which M, S and F are defined as for an automaton A and Q, the set of initial states, is a non-empty subset of S.

Definition 12. A tape x is acceptable to an automaton B of type 3 if $M(s_i, x)$ is in F for some s_i in Q. The set of all acceptable tapes to an automaton B of type 3 is denoted by T(B) and is called the set of acceptable tapes for B.

Theorem 8. Let U be a set of tapes. U = T(B) for some B of type 3 if and only if U = T(A) for a suitable automaton A of type 2.

Proof: Let S' be the set of all subsets of S, s_0' the set Q, and let F' be the set of all subsets of S which contain at least one state in F. Define M' in the following manner: If $s_i' = \{s_{a_1}, \ldots, s_{a_n}\}$ then $M'(s_i', \sigma_j) = \bigcup_{i=1}^{n} \{M(s_{a_i}, \sigma_j)\}$. The definition of M' is extended to S' x T the same way in which the definition of M was extended. Let A = (S', M', s_0', F'). It can now be shown that the automaton A accepts the same set of tapes as the automaton B. This is accomplished in the following manner. Assume x is an acceptable tape for the automaton B. Then for some state s_i in Q, $M(s_i, x) = s_j$, which is in F. Let $x = \sigma_0\sigma_1 \ldots \sigma_{n-1}$. Consider $M'(Q, x) = s_a'$. Since $M(s_i, x) = s_j$ is a state in F and s_j is in s_a', s_a' is a state in F'.

Assume now that x is acceptable to the automaton A. Then M'(Q, x) = s_n' is in F'. Since s_n' is in F', s_n' contains some state s_n in S such that s_n is in F. There is some state, say s_j, such that s_j is in Q and $M(s_j, x) = s_n$. This can be seen by tracing the states of A back from s_n'. Let $s_0', \ldots, s_n'$ be the sequence of states of A which A assumes as A accepts the tape x. There much be some state s_{n-1} in s_{n-1}' such that $M(s_{n-1}, \sigma_n) = s_n$, otherwise s_n would not be in s_n'. Likewise, there is some state s_{n-2} such that s_{n-2} is contained in s_{n-2}' and $M(s_{n-2}, \sigma_{n-2}) = s_{n-1}$. In general there will be a state s_{n-j} in s_{n-j}' such that $M(s_{n-j}, \sigma_{n-j}) = s_{n-j+1}$. Thus, we see that there is a sequence of states beginning with a state Q and ending in s_n'. Since s_n is in F, x is acceptable to the automaton B.

If the set $Q = \{s_0\}$, a set containing only one state, then the automaton B of type 3 is essentially equivalent to an automaton A of type 2. Hence, a set of tapes is acceptable to an automaton B of type 3 if and only if it is the acceptable set of tapes to an automaton A of type 2.

Even though the definition of an automaton and, hence, the acceptability of tapes for an automaton has been changed, we have not changed the type of sets which are acceptable to an automaton. We will in many cases, however, be able to use a smaller number of states to produce the automaton accepting the set of tapes. If it is the case that Q = S, that is, all states are initial, the set T(B) for an automaton B of type 3 has the further property of terminal completeness. This means that if a tape x of length n is in T(B), then ${}_ix_n$ is in T(B) for all $0 \leq i \leq n$. This property of T(B) can be easily seen from the fact that if $M(s_0, x)$ is in F, then $M(M(s_0, {}_0x_i), {}_ix_n)$ is in F. Since $M(s_0, {}_0x_i)$ is in Q, ${}_ix_n$ is in T(B).

In some cases it might not be possible to ascertain whether or not the automaton stopped in a final state, although it would be possible to determine if a tape has been "read" in its entirety. This concept of acceptability corresponds to all states of an automaton being final. The definition of an automaton will now be changed to correspond to this new concept of acceptability.

Definition 13. An automaton C of type 4 is a system $C = (S, M, s_0)$, where S is the set of internal states and M is a mapping of a proper subset L of $S \times \Sigma$ into S and s_0 is an element of S, called the initial state. A tape x is acceptable if $M(s_0, x)$ is defined. (If L were not a proper subset of $S \times \Sigma$, then the automaton C would accept all tapes).

Definition 14. A set of tapes U is initially complete if it contains all its initial segments, that is if $x = \sigma_0\sigma_1 \cdot \cdot \cdot \sigma_{n-1}$ is in U than ${}_0x_h =$ $\sigma_0\sigma_1 \cdot \cdot \cdot \sigma_{h-1}$ is in U for $0 \leq h \leq n$.

Theorem 9. A set U of tapes, denoted by T(C), in T is the acceptable set of tapes for an automaton C of type 4 if and only if U is initially complete and U is the union of all but one of the equivalence classes of a right invariant equivalence relation E over T of finite index, unless U = T in which case U is composed of one equivalence class.

Proof: Let T(C) be the acceptable set of tapes for some automaton C of type 4. If $x = \sigma_0\sigma_1 \cdot \cdot \cdot \sigma_{n-1}$ is in T(C) then $M(s_0, x)$ is defined. This case occurs, however, if and only if $M(s_0, x)$ is defined for all ${}_0x_i$ in T(C). Thus, T(C) is initially complete. That T(C) is the union of all but one of the equivalence classes of a right invariant equivalence relation over T of finite index is a trivial consequence of the fact that C is an automaton of type 2 with F = S and all classes except the one containing the tapes x such that $M(s_0, x)$ is not defined are contained in T(C).

Let U be an initially complete set of tapes which is the union of all but one of the equivalence classes of a right invariant equivalence relation over T of finite index. Since U is the union of some of the equivalence classes of a right invariant equivalence relation over T of finite index, we can construct an automaton A of type 2 which has U as its set of acceptable tapes. If we choose some tape y not in U and remove $M(s_0, y)$ from A along with all transitions to and from this state we will again have an automaton. This new automaton has U as its set of acceptable tapes and is of the type 4.

Since the state which was removed was not final and all but one of the states of the automaton were final, all of the states of the new automaton are final. Since U is complete, no state which is necessary for $M(s_0, x)$ to be defined for any x contained in U was removed, thus the new automaton C is of the type 4 and accepts the set U.

An automaton will now be defined in the form most useful to the notion of a sequential machine, the subject of the next chapter. The notion of acceptability is now, "Is there some state such that if the automaton is in this state and a tape x is presented to it, will the automaton read the entire tape?"

Definition 15. An automaton D of type 5 over the alphabet Σ is a system $D = (S, M)$ where S is a non-empty set (called the set of internal states) and a mapping M of a non-empty subset L of $S \times \Sigma$ into S.

Definition 16. A tape $x = \sigma_0\sigma_1 \ldots \sigma_{n-1}$ is acceptable to the automaton D of type 5 if there exists some s_i in S such that $M(s_i, x)$ is defined. The set T(D) is the set of all those tapes and only those tapes acceptable to D.

Theorem 10. For every automaton D of type 5 there is an automaton C of type 4 such that $T(D) = T(C)$.

Proof: Since the automaton D of type 5 is also of type 3, according to Theorem 3, an automaton A of type 2 can be found which accepts T(D). Since each state of the automaton D is final, T(D) is initially complete and the union of all but one of the equivalence classes of a right invariant equivalence relation over T of finite index. This then gives us that the automaton A is also of type 4, and we have an automaton C of type 4 which accepts T(D).

Since the automaton D has each of its states as initial states the set T(D) is terminally complete. Likewise, since each of its states is final, the set T(D) is initially complete. This leads to the interesting property of an automaton of the type 5, that of total completeness.

Definition 17. A set of tapes U is totally complete, if whenever a tape x of length n is in U, then ${}_ix_j$ is in U for $0 \leq i \leq j \leq n$. That is, if the tape x is in U then any contiguous portion of it is in U.

Theorem 11. A set of tapes U is totally complete if and only if it is both initially and terminally complete.

Proof: If the set of tapes U is totally complete then by definition it is both initially and terminally complete.

Assume the set of tapes U is both initially and terminally complete. We wish to show that if x is in U then ${}_ix_j$ is in U, for $0 \leq i \leq j \leq n$. Since U is initially complete ${}_0x_j$ is in U. If we now apply the condition of terminal completeness to ${}_0x_j$ we have that ${}_ix_j$ is in U. It has, therefore, been shown that the set of tapes acceptable to an automaton of type D is not only initially and terminally complete but is also totally complete.

We have now arrived at one of the goals of this chapter, that is, we have characterized the set of tapes acceptable to an automaton D of type 5 which is similar to a sequential machine except that no outputs have been associated with this automaton. In the next chapter we will consider sequential machines, which have outputs associated with them, and point out in which ways these theorems will have to be changed to be applicable to sequential machines.

Chapter II

SEQUENTIAL MACHINES

The idea of sequential machines has come into use in recent years in many fields of study. It has been employed by McCulloch and Pitts [20] and others in the representation of nerve nets. Kleene [19] used this idea for his work on the representation of events. Huffman [17], Mealy [22], Hohn [2], Aufenkamp [2, 3], and Ginsburg [10, 11, 12, 13] have dealt extensively with the synthesis and analysis of sequential machines to be used in the design and construction of computers and other types of process control equipment, where the process is essentially sequential in nature.

In this chapter we will characterize the sets of input-output sequences which are acceptable to a sequential machine. A sequential machine can be thought of as a set of states (possibly internal configurations of a device) "accepting" input sequences and "producing" output sequences, such that if the device is in a state and is given an input, the internal configuration changes to a new state (possibly the same state again) and an output is given.

There are two commonly used models of sequential machines. One, the Moore [23] model, associates outputs with states. The other associates outputs with transitions of one state to another with a given input. Sequential machines of this type are known as the Mealy [22] model, and are the type that will be primarily discussed in this chapter. A Mealy model sequential machine is defined formally in the following way.

Definition 1. A sequential machine with input alphabet Σ and output alphabet 0 is a system **A** = (S, P) where S is a non-empty finite set (the internal states of A), and P is a function defined as a non-empty (not necessarily proper) subset J of the Cartesian product S x 0. Specifically if s_i is in S and σ_j is in Σ, $P(s_i, \sigma_j) = (M(s_i, \sigma_j), N(s_i, \sigma_j))$ where M is defined as in Chapter I and N is a function of the subset L of $S \times \Sigma$ into 0.

An input-output sequence x is a finite sequence of ordered pairs $(\sigma_i, 0_j)$ where σ_i is an input symbol and 0_j is an output symbol. x will be written as $\frac{x^I}{x^0}$ where x^I is the input sequence and x^0 is the output sequence.

In order to discuss the idea of input-output sequences, it is necessary to extend the definition of P. Let θ be the set of all finite sequences of input symbols, and λ the null sequence of θ. Let Φ be the set of all finite sequences of output symbols with μ the null sequence of output symbols. The function P is extended from $S \times \Sigma$ to $S \times \theta$ by extending the definitions of the two functions M and N. M is extended just as in Chapter I, while N is extended to $S \times \theta$ in the following way:

$$N(s_j, \lambda) = \mu$$

$$N(s_j, \sigma_0\sigma_1 \ldots \sigma_{n-1}) = N(s_j, \sigma_0)\, N(M(s_j, \sigma_0), \sigma_1 \ldots \sigma_{n-1}),$$

where s_j is any state in S and $\sigma_0\sigma_1 \ldots \sigma_{n-1}$ is any tape in Σ such that $M(s_j, \sigma_0)$ is defined. The function P is extended from $S \times \Sigma$ to $S \times \theta$ by

$$P(s_i, x^I) = (M(s_i, x^I), N(s_i, x^I)),$$

where x^I is an input sequence.

There are several methods for specifying a sequential machine. Some of these are as follows:

(1) Transition and output tables. This is a table, called a next state-output matrix by Ginsburg [12], which has a list of states in a vertical column and a list of inputs in a horizontal row. In the position corresponding to the state s_i and the input σ_j is the ordered pair (s_p, o_q) corresponding to the next state and the output which occur when the machine is in the state s_i and input σ_j is given, that is $P(s_i, \sigma_j) = (s_p, 0_q)$. For example, let Σ and 0 be the sets $\Sigma = \{a, b\}$ and $0 = \{\alpha, \beta\}$. Then let A be the sequential machine with $S = \{0, 1, 2, 3, 4, 5\}$ and with P defined by Table 1 with the interpretation, that if the state is s_i and the input σ_j, then the next state and output pair is $P(s_i, \sigma_j)$.

Table 1

State	Next State and Output	Next State and Output
	Input a	Input b
0	(0, α)	(1, α)
1	(0, α)	(2, α)
2	(0, α)	(3, β)
3	(4, β)	(3, β)
4	(5, β)	(3, β)
5	(0, α)	(3, β)

(2) A pictorial method for displaying a sequential machine is the state diagram. This is a directed graph in which the vertices represent the states and the directed line segments represent transitions. If $P(s_i, \sigma_p) = (s_j, o_q)$, then the directed line segment from state s_i to state s_j is assigned the ordered pair (σ_p, o_q). The state diagram for the previous example is:

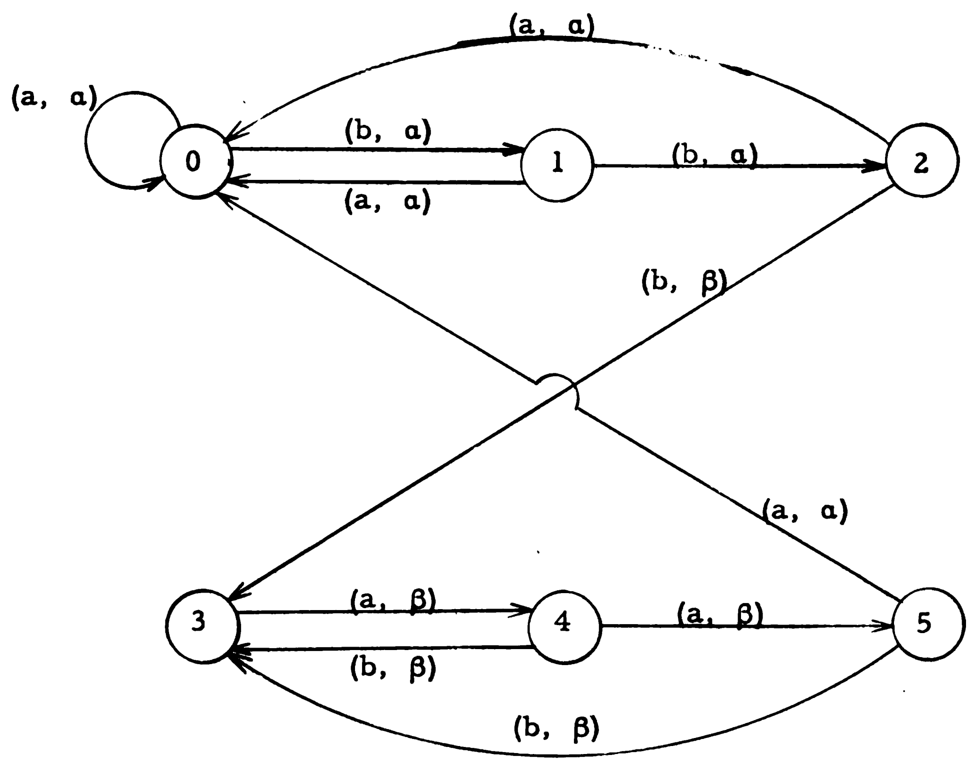

Figure 1

(3) A useful tool for the analysis of sequential machines is the connection matrix [2]. If $P(s_i, \sigma_p) = (s_j, o_q)$ then the i, j th position is the ordered pair $(\sigma_p, 0_q)$ or the formal sum of such pairs. If there is no input σ_p such that $M(s_i, \sigma_p) = s_j$ then the i, j th position is zero.

The connection matrix for the previous example is:

$$\begin{pmatrix} (a, \alpha) & (b, \alpha) & 0 & 0 & 0 & 0 \\ (a, \alpha) & 0 & (b, \alpha) & 0 & 0 & 0 \\ (a, \alpha) & 0 & 0 & (b, \beta) & 0 & 0 \\ 0 & 0 & 0 & (b, \beta) & (a, \alpha) & 0 \\ 0 & 0 & 0 & (b, \beta) & 0 & (a, \beta) \\ (a, \alpha) & 0 & 0 & (b, \beta) & 0 & 0 \end{pmatrix}$$

Figure 2

The work done by Rabin and Scott [25], together with the results of Chapter I, when considered in the light of sequential machine theory, leads us to the investigation of acceptability of sets of I-0 sequences to sequential machines. This will be the subject of the present chapter.

Definition 2. An input-output sequence $x = \frac{x^I}{x^0}$ is acceptable to a state s_j of a machine A if $N(s_j, x^I) = x^0$.

Definition 3. A set, to be denoted by $T(A_i)$, of I-0 sequences is the set of acceptable I-0 sequences for the state s_i of the sequential machine A, if all the I-0 sequences in $T(A_i)$ are acceptable to the state s_i of the sequential machine A and no other I-0 sequence is acceptable to the state s_i of the sequential machine A.

Definition 4. A partial segment ${}_i x_j$ of an I-0 sequence $x = y_0 y_1 \cdot \cdot \cdot y_{n-1}$ is the sequence $y_i y_{i+1} \cdot \cdot \cdot y_{j-1}$ where the y_i are input-output pairs. By ${}_i x_i$ is meant the I-0 sequence $\frac{\lambda}{\mu}$.

Definition 5. The number of input-output pairs that make up the I-0 sequence x is called the length of x.

Definition 6. The particular partial segment ${}_0 x_i$ is called an initial segment and the partial segment ${}_i x_n$, where x is of length n, is called a terminal segment.

Definition 7. A set of I-0 sequences is initially complete if, whenever an I-0 sequence x is in U, then all initial segments of x are in U. That is if x is in U, then ${}_0 x_i$ is in U for all $i \leq n$, where n is the length of x.

Definition 8. A set U of I-0 sequences is consistent if, whenever two I-0 sequences are in U, which have an initial segment ${}_0 x_i^I$ of their input sequences in common, then they both have the same output sequences ${}_0 x_i^0$.

Definition 9. The set of all finite input-output sequences is denoted by T.

Lemma 1. Let U be a consistent set of I-0 sequences. Define the relation E by xEy, if and only if for all z in T, if xz is in U, then yz is in U and conversely. If E is an equivalence relation of finite index and U is the union of all but one of these equivalence classes, then U is initially complete.

Proof: Since U is consistent one of the equivalence classes must contain those I-0 sequences which are inconsistent with the I-0 sequences in U. Thus, if x is in U, ${}_0x_i$ must also be in U, since ${}_0x_i\,{}_ix_n$ is in U, and must be in one of the equivalence classes different from the one containing inconsistent I-0 sequences. Since U is the union of all but one equivalence class ${}_0x_i$ is in U.

Theorem 1. A necessary and sufficient condition that a set U of I-0 sequences be the acceptable set of I-0 sequences for a state s_i of a sequential machine A is that U be consistent and the union of all but one of the equivalence classes of the explicit right invariant equivalence relation E of finite index.

Proof: Assume that U is the acceptable set of I-0 sequences for state s_i of the sequential machine A, that is $U = T(A_i)$. Since $N(s_i, x^I\sigma) = N(s_i, x^I)\, N(M(s_i, x^I), \sigma)$, the set U is consistent.

Since U is consistent, there is only one output sequence associated with any input sequence, thus only the input sequences x^I need be considered. Let x and y be any two I-0 sequences contained in U. If $M(s_i, x^I) = M(s_i, y^I)$ then x and y are equivalent, for if z is any I-0 sequence contained in T, such that xz is in U, then $M(s_i, x^Iz^I) = M(M(s_i, x^I), z^I) = M(M(s_i, y^I), z^I) = M(s_i, y^Iz^I)$ and x and y are equivalent. Since $M(s_i, x^I)$ is unique, the set of I-0 sequences such that $M(s_i, x) = s_q$ is either disjoint from the set of I-0 sequences such that $M(s_i, x) = s_p$ for $p \neq q$ or both sets are in the same equivalence class. In the latter case s_p and s_q are called indistinguishable [23, p. 136].

Likewise all non-acceptable I-0 sequences are in the same equivalence class because if x is not in U, xz is not in U for any z in T. Thus, if A is an n state sequential machine, $U = T(A_i)$ is the union of at most n equivalence classes.

Assume a set U of I-0 sequences is consistent and that the given equivalence relation has a finite index and U is the union of all but one of these classes. Since the set U is consistent, U might possibly be the set of acceptable I-0 sequences for some state of some sequential machine.

A sequential machine A = (S, P), which accepts U, can be constructed from the equivalence classes. First, let S be the set of all equivalence classes $[x_1]$, $[x_2]$, . . . $[x_{n-1}]$ of T under E except for that class which contains I-0 sequences inconsistent with I-0 sequences of U. The function P is defined as follows. Let x be any I-0 sequence in one of the equivalence classes contained in U and let (σ_i, o_i) be any input-output pairs. If $x\frac{\sigma_i}{o_i}$ is in an equivalence class contained in U, then $P([x], \sigma_i) = ([x\frac{\sigma_i}{o_i}], o_i)$. Repeating this for all equivalence classes in U and all input-output pairs completely defines P.

To show that U is accepted by A, we proceed as follows:

Denote by s_i the state corresponding to the equivalence class $\frac{\lambda}{\mu}$ containing the null input-output sequence. Let x be any I-0 sequence in U. According to Lemma 1, U is initially complete and ${}_0x_i$ is in U. The equivalence class $[{}_0x_1]$ containing ${}_0x_1$ is one of the states of A.

Define: $P(s_i, {}_0x_1) = ([{}_0x_1], {}_0x_1^0),$
$P([{}_0x_1], {}_1x_2) = ([{}_0x_2], {}_1x_2^0), \ldots,$
$P([{}_0x_{n-1}], {}_{n-1}x_n) = ([{}_0x_n], {}_{n-1}x_n^0).$

We see that ${}_0x_1, {}_0x_2, \ldots, {}_0x_n = x$ are acceptable to the state s_i of the sequential machine A. Since this is true for any x in U, the sequential machine A accepts the set U of I-0 sequences.'

Assume that x is not in U. Then there exists some initial segment ${}_0x_i$ for $0 \leq i \leq n$ such that ${}_0x_i$ is not in U, but ${}_0x_{i-1}$ is in U. Thus, $N(s_i, {}_0x_i^I) \neq {}_0x_i^0$ and ${}_0x_i$ is not acceptable to state s_i of the sequential machine A. Since U is complete x cannot be acceptable to state s_i of the sequential machine A. This shows that state s_i accepts U and only U, and that $U = T(A_i)$.

As previously mentioned we have been dealing with Mealy's model of a sequential machine. It is interesting to note the differences in the sets of acceptable I-0 sequences to a state of a Mealy model sequential machine and those acceptable to a state of a Moore's model sequential machine. Nerode [24, p. 542] has shown that in order for a set U of I-0 sequences to be acceptable to a Moore model sequential machine it must be "causal". A set of I-0 sequences is causal if for any x and y in U, if x and y have some initial segment in common then the outputs associated with the next inputs even if the next inputs are different must be the same. That is, if x and y are two I-0 sequences such that ${}_0x_i^I = {}_0y_i^I$, then ${}_0x_{i+1}^0 = {}_0y_{i+1}^0$. Using this fact it can be shown that Mealy's model is more general than Moore's model. Consider the set U of I-0 sequences x over the alphabet $\Sigma = \{a, b\}$ with output alphabet $0 = \{\alpha, \beta\}$ such that x is in U if and only if x always has α associated with a and β with b. This set is consistent and the equivalence relation E is a right invariant equivalence relation of index two. Thus a Mealy model sequential machine can be constructed to accept the set U of I-0 sequences. In particular, the sequential machine in figure 3 accepts the set U.

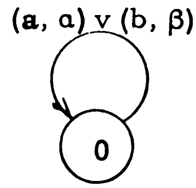

Figure 3

This set of I-0 sequences, however, is not causal. This can be seen by considering the I-0 sequences $\frac{a,\ b}{\alpha,\ \beta}$ and $\frac{a,\ a}{\alpha,\ \alpha}$. These have the initial segment $\frac{a}{\alpha}$ in common but the next outputs are different. Thus there is no Moore model sequential machine which will accept the set U.

Any set of I-0 sequences acceptable to a Moore model is, however, also acceptable to a Mealy model. The following algorithm, in fact, gives a method of converting a Moore model sequential machine to a Mealy model sequential machine:

Remove all outputs from the states and associate the output which was associated with a state with all input symbols emanating from this state. Figure 4 is a Moore model sequential machine and figure 5 is its equivalent Mealy model sequential machine. We have shown that any set of I-0 sequences acceptable to a Moore model sequential machine is also acceptable to a Mealy model sequential machine, but the converse is not true. Thus, Mealy's model of a sequential machine is the more general of the two.

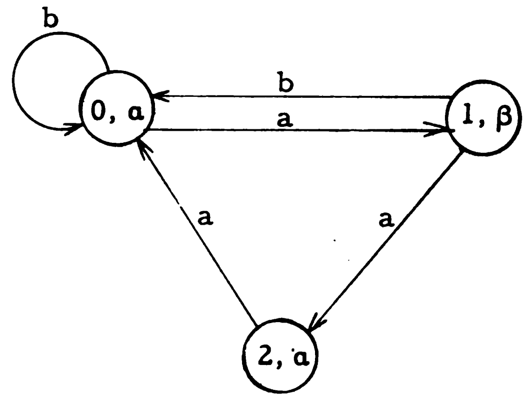

Figure 4

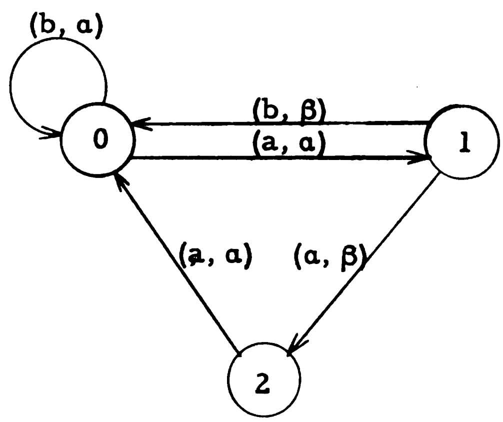

Figure 5

It is now appropriate to consider the set of I-0 sequences that are acceptable to a sequential machine A, if the sequential machine is allowed to start in any state. It will be assumed that each sequential machine A has some state say s_0, called the connecting state, such that for each s_i there is an I-0 sequence x such that $M(s_0, x^I) = s_i$. Such a machine is called a connected sequential machine. We will also assume that the sequential machine A is in reduced form [3, p. 282].

Definition 10. An I-0 sequence x is acceptable to a sequential machine A if there exists some state s_i in S such that $N(s_i, x^I) = x^0$.

Definition 11. The set of I-0 sequences composed of all and only those I-0 sequences acceptable to A is called the acceptable set of I-0 sequences for A and is denoted by T(A).

Let $T(A_0)$ be the set of I-0 sequences acceptable to the state s_0. This is the union of all but one of the equivalence classes of the given equivalence relation of finite index. Let us now look at one of these equivalence classes. It represents the set of I-0 sequences that terminate in a particular state.

Thus, if from each I-0 sequence in $T(A_0)$, which has an initial segment which is identical with an I-0 sequence of the equivalence class corresponding to state s_i, the initial segment ${}_0x_i$ is removed, the terminal segment ${}_ix_n$ will be an I-0 sequence acceptable to the state s_i of the sequential machine A. The set of all such terminal sequences is the set of acceptable I-0 sequences for state s_i of the sequential machine A.

Call this set $T(A_i)$. This set, of course, satisfies the given right invariant equivalence relation over T. If we repeat this process for all states we produce the set of I-0 sequences acceptable to each state. Thus, the set T(A) of I-0 sequences acceptable to the machine A is $\bigcup_{0}^{n-1} T(A_i)$. These ideas are summarized in Theorem 2.

Definition 12. The difference of an equivalence class of I-0 sequences [x] from a set of I-0 sequences U, written U - [x], is a set of I-0 sequences, such that ${}_ix_n$ is in U - [x] if and only if ${}_0x_i\, {}_ix_n$ is in U and ${}_0x_i$ is in [x].

Theorem 2. The set of acceptable I-0 sequences for a sequential machine A is $T(A) = \bigcup_{i=1}^{n} T(A_i)$, where each $T(A_i)$ satisfies the conditions of Theorem 1. Also $T(A_i) = T(A_0) - [x]$ where s_0 is the connecting state and [x] is an equivalence class of $T(A_0)$.

Theorem 3. Let U be a set of I-0 sequences. Assume there exists a subset U_0 of U and that the given right invariant relation E separates U_0 into a finite number of equivalence classes. Assume also that the differences $U_i = U_0 - [x]_i$, where $[x]_0, \ldots, [x]_{n-1}$ are distinct equivalence classes contained in U, are divided into a finite number of equivalence classes by the given equivalence relation. Then if $U = \bigcup_{i=0}^{n-1} U_i$ and each U_i is consistent then a sequential machine can be constructed which will accept U.

Proof: The sequential machine A is formed by constructing the sequential machine which has some state say s_0, which will accept the set U_0. This machine will then have for its set of acceptable I-0 sequences for its other states s_i the sets $T(A_i)$. But each $T(A_i) = U_0 - [x]_i$. Hence A accepts the set U.

Theorem 4. Let U be a set of I-0 sequences. U is the set of acceptable I-0 sequences for a sequential machine A, if and only if the following three conditions are true.

(1) The relation R defined by xRy if for all z and w in T, if zxw is in U, then zyw is in U, and conversely, is a congruence relation of finite index.

(2) U is the union of all but 1 of the congruence classes of the relation R.

(3) Each of the congruence classes contained in U is consistent.

Proof: Let A be a sequential machine. Let R be a relation such that xRy, if $M(s_i, x^I) = M(s_i, y^I)$ for all s_i in S whenever both $M(s_i, x^I)$ and $M(s_i, y^I)$ are defined, or both are not defined for the same s_i. Then R is a congruence relation. The proof of this follows that of the similar theorem for finite automata. If xRy, then x and y must be in the same equivalence class of the right invariant equivalence relation E, and according to Theorem 1, the congruence classes are consistent. If there are r internal states in A then for a fixed I-0 sequence x, $M(s_i, x^I)$ can be any one of r states or be undefined. Thus, the relation R separates T into at most $(r+1)^r$ equivalence classes and consequently R is of finite index.

Assume that U is the union of all but one of the congruence classes of the given congruence relation and that each equivalence class contained in U is consistent. Let [x] denote a congruence class and let each congruence class contained in U denote a state. Define P as follows: if [x] and

[y] are two congruence classes, such that [x] contains an initial segment $_0y_{n-1}$ of some I-0 sequence $_0y_n$ of [y], and if $_{n-1}y_n$ is $\frac{\sigma_i}{o_i}$, then $P([x], \sigma_i) = ([y], o_i)$. This, then, completely defines a sequential machine A.

The proof that U = T(A) is analogous to that of Theorem 1.

To this point we have been studying as much of the structure of general sequential machines as could be distinguished by considering only input-output sequences. However, most sequential machines which will be useful in actual switching circuits design have the property that for any pair of states s_i and s_j there is an I-0 sequence x such that $M(s_i, x^I) = s_j$. Such machines are said to be strongly connected. In the remainder of this chapter we will make use of the characterization of the set of acceptable I-0 sequences to prove some interesting properties of strongly connected sequential machines and a certain subclass of such machines, positive sequential machines.

Definition 13. Two states s_i and s_j of a connected sequential machine A are equivalent if there exist two I-0 sequences x and y such that $M(s_0, x^I) = s_i$ and $M(s_0, y^I) = s_j$ and both x and y are in the same equivalence class of T under the given right invariant equivalence relation E. This definition compares with that given by Hohn and Aufenkamp [3].

Definition 14. A connected sequential machine is in reduced form if it does not possess any pair of equivalent states.

Theorem 5. A connected sequential machine with n states is in reduced form if and only if $T(A_0)$, when s_0 is the connecting state, is the union of n equivalence classes of the given right invariant equivalence relation E over T.

Proof: Assume A is a reduced connected sequential machine with n states. Since A is connected, for any i there exists an I-0 sequence x such that $M(s_0, x^I) = s_i$. Since A is reduced, if x and y are two I-0 sequences such that $M(s_0, x^I) \neq M(s_0, y^I)$, then x and y are in different equivalence classes. It has been shown that each state corresponds to an equivalence class and that there are n equivalence classes.

Assume $T(A_0)$ is the union of n equivalence classes. Since there are only n states $M(s_0, x^I) \neq M(s_0, y^I$ if $[x]$ is not equivalent to $[y]$. Thus the sequential machine A is in reduced form.

Definition 15. A sequential machine A with n states is strongly connected if for each i and j $(i, j \leq n)$ there exist some input-output sequence x such that $M(s_i, x^I) = s_j$.

Theorem 6. Let A be an n state sequential machine. If for $i = 1, 2, \ldots, n$, $T(A_i)$ is the union of n equivalence classes of the given right invariant equivalence relation E, then A is strongly connected.

Proof: Each equivalence class of $T(A_i)$ consists of the I-0 sequences that terminate in distinct states. That is, distinct I-0 sequences in different classes of $T(A_i)$ end in distinct states. Thus, if there are n states and $T(A_i)$ is the union of n classes then there is an I-0 sequence x such that

$M(s_i, x) = s_j$ for $j = 1, 2, \ldots, n$. Since this is true for all i, A is strongly connected.

This condition is not necessary, for two I-0 sequences may terminate in different states and still be in the same class, that is if these states are equivalent. For a reduced machine, however, the conditions of the previous theorem are also necessary.

Theorem 7. The reduced form of a strongly connected sequential machine is a strongly connected sequential machine.

Proof: Let A be a strongly connected sequential machine and let s_i and s_j be any two states of the reduced form of the sequential machine A. If s_i and s_j are not states which were merged in the formation of a reduced sequential machine then there still exists x such that $M(s_i, x^I) = s_j$. Let either s_i or s_j be a state of the reduced machine produced by the merging of two states of A. Since there exists an I-0 sequence x such that the function M of one of the states forming s_i and x^I is in one of the states forming s_j, then $M(s_i, x^I) = s_j$.

Definition 15. A sequential machine A is positive if there exists some r such that for all i and j ($i, j \leq n$) there exists an I-0 sequence x of length r such that $M(s_i, x^I) = s_j$.

Theorem 8. The reduced form of a positive sequential machine is a positive sequential machine.

Proof: The proof of this theorem is similar to that of the previous theorem.

We have completed one of the major objectives of this study, that of characterizing the sets of acceptable I-0 sequences for a sequential machine and have also shown the usefulness of this characterization in proving some theorems concerning the structure of sequential machines.

Chapter III

INDECOMPOSABLE MATRICES AND THE STRUCTURE OF AUTOMATA

The definition of a strongly connected sequential machine was introduced by Moore [23, p. 140] and was used by him mostly for the determinations of the minimal length experiment necessary to distinguish one machine from another. Weeg and Kateley [18] used this idea of strongly connected machines to prove equivalence of a certain class of sequential machines. Seshu, Miller and Metze [21] studied strongly connected machines as such. They made use of connection matrices, which are similar to the connections matrices to be used in this chapter.

The concept of a positive machine was introduced by Weeg [27]. Its connection matrix corresponds very closely to the idea of a primitive matrix studied by Frobenius [7], Herstein [15], Holladay and Varga [16] and others.

Various types of matrix representations of sequential machines have been employed in the study of sequential machines. Hohn and Aufenkamp [3] use a connection matrix which has the formal sum of all I-0 pairs of the sort (a, α) in the i j position if $P(s_i, a) = (s_j, \alpha)$. They make use of this representation to reduce a sequential machine to minimal form.

Seshu, Miller, and Metze [21] employ another kind of connection matrix called a transition matrix. Corresponding to each input symbol σ_i a transition matrix T^i is defined. If $M(s_i, \sigma_i) = s_j$ then there is a one in the i, j th position of T^i. If there is no such transition then the i, j th position is 0. If the alphabet Σ contains exactly the symbols $\sigma_1, \sigma_2, \ldots, \sigma_n$ then the matrix C as used in [3] is $C = \sum_{i=1}^{n} \sigma_i T^i$, where addition signifies Boolean inclusive or. Gill [8] assigns a prime number to each transition and inserts this number in the i, j th position if the transition carries state s_i to state s_j.

Weeg [27] used the following definition for a connection matrix which is similar to the transition matrices of Seshu, Metze, and Miller [21]. If for some σ_p in Σ, $M(s_i, \sigma_p) = s_j$ then there is a one in the i, j th position of the connection matrix C. Otherwise the i, j th position is 0. This is the definition that will be employed in this chapter. The connection matrix for the example in figure 1 of Chapter II is:

$$C = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Figure 1

By C^r is meant the rth power of the connection matrix C, where multiplication is the normal matrix multiplication with Boolean arithmetic. The sum of two connection matrices, written $A \cup B$, is the elementwise addition with Boolean arithmetic. It is to be noted that the definition of a connection matrix is independent of input symbols and output symbols, thus, the connection matrix and all the theorems of this chapter apply to both finite automata and both Mealy's and Moore's model sequential machine. They also apply to precedence matrices for directed graphs, as discussed by Harary [14] and others.

In common with current literature, the term path of length n from s_i to s_j will be used to denote that there exists some input sequence x of length n such that $M(s_i, x) = s_j$. It is to be noted that a 1 in the i, j th position of C^r means that there is a path of length r from s_i to s_j.

Definition 1. A sequential machine is strongly connected if there is a path from each state to each other state of the sequential machine.

Definition 2. A connection matrix is strongly connected if the sequential machine which it represents is strongly connected.

In the remainder of this chapter the terms connection matrix and the sequential machine which it represents will be used synonymously, and the rows and columns will be used synonymously with the states they represent.

A necessary condition that a connection matrix be strongly connected is that it have a non-diagonal 1 in each row and in each column. For suppose some row does not have an off diagonal 1. Then there is no path from this state to any other state. If there is a column without any off diagonal ones, then there is no path which terminates in this state, and thus there is no path to this state from any other state. This condition is not, however, sufficient. As can be seen from the connection matrix of figure 2, it is not possible to have a path from state s_4 or s_5 to s_1, s_2, or s_3.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 2

The connection matrix in figure 1 is strongly connected, as there is a path of length 5 passing through each state.

Definition 3. An n x n connection matrix C is indecomposable if there is no permutation matrix P for which

$$P\,C\,P^T = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

where A_{11} and A_{22} are square submatrices. (A_{12} may not be square).

Theorem 1. An n x n connection matrix is strongly connected if and only if it is indecomposable.

Proof: The transformation $P\,C\,P^T$ corresponds to a permuation of the numbers assigned to the states. But the property of strongly connectedness is independent of the numbering of the states. Now if C is decomposable as

$$P\,C\,P^T = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

then there can be no path from any state corresponding to the rows of A_{22} into any state corresponding to the rows of A_{11}. Hence C would not be strongly connected.

On the other hand, suppose that the n x n matrix C is not strongly connected. Then there is some set S' of $p < n$ states for which $M(s_i, \sigma_j)$ is in S' for all s_i in S' and all σ_j in Σ. If the permutations are performed which make these states correspond to the last columns of the connection matrix, the first n-p entries of the last p rows of the connection matrix will be zero. Hence, if P is that permutation matrix then

$$P\,C\,P^T = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

where A_{11} is of dimension (n-p) x (n-p), A_{12} is (n-p) x p, A_{22} is p x p, and 0 is p x (n-p). But then C is decomposable, so that we have proved that if C is indecomposable, then C is strongly connected.

Definition 4. An n x n connection matrix C is positive if there exists some positive integer r such that $C^r = U$, where U is the n x n matrix which has 1 in each position.

The connection matrix for figure 1 is positive; in fact $C^5 = U$.

Holladay and Varga [16, p. 631] give the following definition of a primitive matrix.

Definition 5. A matrix A on non-negative real numbers is primitive if there exists some positive integer r such that A^r has all positive entries.

It is to be noted that a positive real number in the powers of a primitive matrix corresponds to a one in the powers of a connection matrix. That is, to each non-negative matrix $A = (a_{ij})$ there corresponds a connection matrix $C = (c_{ij})$ such that if $a_{ij} \neq 0$ then $c_{ij} = 1$, while if $a_{ij} = 0$, $c_{ij} = 0$. Thus if one is only interested in the structure of a matrix, and not in the magnitude of the value of its elements, then one need deal only with the corresponding connection matrix. This approach to many of the problems concerned with primitive matrices appears to lead to simpler proofs.

As pointed out by Weeg [27] every positive connection matrix is strongly connected, but the converse statement is not true. Thus the class of positive connection matrices is a subclass of the class of strongly connected connection matrices. It is this subclass of positive connection matrices and the differences which distinguish this subclass from the class of strongly connected connection matrices that will be studied in this chapter.

As previously noted [27] all powers of a positive connection matrix are positive and thus strongly connected, whereas all the powers of a strongly connected connection matrix are not necessarily strongly connected. This can be seen from the example in figure 3. C is strongly connected but C^3 is not.

$$C = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad C^2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad C^3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure 3

In fact the class of positive connection matrices is exactly that class of strongly connected connection matrices all of whose powers are strongly connected, as is shown by the next theorem.

Theorem 2. An $n \times n$ connection matrix C is positive if and only if C^r is strongly connected for all integers $r \geq 1$.

Proof: As previously pointed out if C is positive, C^r is strongly connected for all integers $r \geq 1$.

Suppose that C^r is strongly connected for all integers $r \geq 1$. In particular, then, C is strongly connected, so that for each $i = 1, 2, \ldots, n$ there must exist an integer $m_i \geq 1$ for which $c_{ii}^{(m_i)} = 1$. (By $c_{ij}^{(r)}$ is meant the ij entry of C^r). Further, if $c_{ii}^{(m_i)} = 1$, so also does $c_{ii}^{(qm_i)} = 1$ for any integer $q \geq 1$. Hence, $c_{ii}^{(m_1 m_2 \cdots m_n)} = 1$ for $i = 1, 2, \ldots, n$, so that $C^{m_1 \cdots m_n}$ has a 1 for each diagonal entry. But Herstein [15, p. 20] has proved that if C is indecomposable and each diagonal entry is positive, then C is primitive. Since all powers of C are strongly connected $c^{m_1 \cdots m_n}$ is strongly connected, so that C is positive, as was to be proved.

If a connection matrix C is not positive then there is an integer r such that C^r is not strongly connected. It will be shown that if a strongly connected n x n matrix C has some power C^r which is not strongly connected, then in fact one of the first n powers is not strongly connected. This will be shown without recourse to Herstein's Lemma [15, p. 20].

Theorem 3. Let C be an n x n connection matrix. C^r is strongly connected for $1 \leq r \leq n$, if and only if C is positive.

Proof: If C is strongly connected, then for any i there is a path k of length $p \leq n$ which carries state s_i back to state s_i. For any pair i, j there is a path h of length pq, $q \leq n$ from state s_i to state s_j. This is true since C^p is strongly connected.

However, for any i and j there is a 1 in the i, j th position of $C^{(n!)^2}$. This follows from the fact that the path, which is comprised of $\frac{(n!)^2 - pq}{p}$ copies of the path k, followed by one copy of the path h, is of length $\frac{(n!^2 - pq)}{(p)}$ $p + pq = (n!)^2$ and carries state s_i into state s_j. Since this is true for all pairs i, j, $C^{(n!)^2}$ has a 1 in each position and the matrix C is positive. The proof of the converse is trivial.

As can be seen from the proof the conditions of the previous theorem are too strong. All that is needed is that if p_i is the minimum length of any path which carries state s_i back to state s_j then C^{p_i} is strongly connected. This gives the following:

Corollary 1. Let C be an n x n connection matrix and let $p_i \leq n$ be the least integer such that $c_{ii}^{(p_i)} = 1$. C is a positive connection matrix if and only if C^{p_i} is strongly connected for $i = 1, 2, \ldots, n$.

It is now a simple matter to prove Herstein's Lemma [15, p. 25], which was used in the proof of Theorem 2.

Corollary 2. (Herstein's Lemma [15, p. 20]). If C is an $n \times n$ strongly connected connection matrix and $c_{ii} = 1$ for $1 \leq i \leq n$, then C is a positive connection matrix.

Corollary 3. Let C be a strongly connected connection matrix. If p_i, $1 \leq i \leq n$, is an integer such that there is a path of length p_i which carries state s_i back to state s_j and if $C^{p_i \cdots p_n}$ is strongly connected, then C is a positive connection matrix.

Proof: Since $p_1 \cdots p_n$ is an integer multiple of p_i for each $1 \leq i \leq n$, $C^{p_1 \cdots p_n}$ has a 1 in each diagonal position and according to Corollary 1 C is positive.

Corollary 4. If C is a symmetric connection matrix, such that C and C^2 are strongly connected, then C is positive.

Proof: Since C is a symmetric matrix and strongly connected, C^2 has a 1 in each diagonal position. Thus according to Corollary 1 C is positive.

Weeg [27] has shown, that if C is strongly connected, then $I \cup \bigcup_{i=1}^{n-1} C^i = U$. This can be used to show another interesting property of strongly connected matrices.

Theorem 4. If C is strongly connected then $C \cup C^2$ is positive.

Proof: To show that $C \cup C^2$ is positive it is sufficient to produce the power of $C \cup C^2$ which is U. But

$$
\begin{aligned}
(C \cup C^2)^n &= C^n(I \cup C)^n \\
&= C^n(I \cup C \cup C^2 \cup \ldots \cup C^n) \\
&= C^n U.
\end{aligned}
$$

Since C is strongly connected there is a path of length n emanating from every state. Also there is a path of length n terminating at each state. Thus, there is a 1 in each row and column of C^n. This gives that $C^n U = U$ and $C \cup C^2$ is a positive connection matrix.

As one notices during the review of this chapter, the major difference between strongly connected connection matrices which are not positive, and those that are, is in the strongly connectedness of the powers of a positive connection matrix. The similarity of positive connection matrices and primitive matrices is also of interest. All of the theorems of this chapter, since they deal only in the structure of the matrix, apply to primitive matrices. It seems to be the case that certain results can be obtained more easily by resorting to the machines corresponding to a matrix than by analyzing the matrix itself. Thus, Herstein's Lemma is easily proved in this fashion though its original proof was not so simple.

BIBLIOGRAPHY

1. Aufenkamp, D. D., and R. G. Gillespie, "On the analysis of sequential machines, "IRE Transactions on Electronic Computers, Vol. EC-7 (1958) pp. 119-122.

2. Aufenkamp, D. D., F. E. Hohn, and S. Seshu, "The theory of nets," IRE Transactions on Electronic Computers, Vol. EC-6 (1957) pp. 154-161.

3. Aufenkamp, D. D., and F. E. Hohn, "Analysis of sequential machines," IRE Transactions on Electronic Computers, Vol. EC-6 (1957) pp. 276-285.

4. Aufenkamp, D. D., "Analysis of sequential machines II," IRE Transactions on Electronic Computers, Vol. EC-7 (1958) pp. 299-306.

5. Bellman, R., "Sequential machines, ambiguity and dynamic programming", Jour. Assoc. Computing Machinery, Vol. 7 (1960) pp. 24-28.

6. Cadden, W. J., "Equivalent sequential circuits," IRE Transactions on Circuit Theory, Vol. CT-6 (1959) pp. 30-34.

7. Frobenius, G., "Uber Matrizen aus nicht negativen Elementen," Sitzungsberichte der Preussischen Akademie dur Wissenschaften zu Berlin (1912) pp. 456-477.

8. Gill, A., "Analysis of nets by numerical methods," Jour. Assoc. Computing Machinery, Vol. 7 (1960) pp. 251-254.

9. Ginsburg,S., "On the length of the smallest uniform experiment which distinguishes the terminal states of a machine," Jour. Assoc. Computing Machinery, (1958) Vol. 5, pp. 266-280.

10. Ginsburg, S., "A sy thesis technique for minimal state sequential machines," IRE Transactions on Electronic Computers, Vol. EC-8 (1959) pp. 13-24.

11. Ginsburg, S., "On the reduction of superfluous states in sequential machines," Jour. Assoc. Computing Machinery, Vol. 6 (1959), pp. 259-282.

12. Ginsburg, S., "A technique for the reduction of a given machine to a minimal state machine," IRE Transactions on Electronic Computers, Vol. EC-8 (1959) pp. 346-355.

13. Ginsburg, S., "Synthesis of minimal-state machines," IRE Transactions on Electronic Computers, Vol. EC-8 (1959) pp. 441-449.

14. Harary, F., "On the consistency of precedence matrices," Jour. Assoc. Computing Machinery, Vol. 7 (1960) pp. 255-259.

15. Herstein, I. N., "A note on primitive matrices," Amer. Math. Monthly, Vol. 61 (1954) pp. 18-20.

16. Holladay, J., and R. Varga, "On powers of non-negative matrices," Proc. Amer. Math. Society, Vol. 9 (1958) pp. 631-634.

17. Huffman, D.A., "The synthesis of sequential switching circuits," Jour. Franklin Inst., Vol. 257 (1954) pp. 161-190, 275-303.

18. Kateley, J., and G. Weeg, "Some properties of strongly connected machines," Presented at the 15th National Conference of the Association for Computing Machinery, August 1960.

19. Kleene, S. C. "Representation of events in nerve nets on finite automata," Automata Studies, Princeton University Press, (1956), pp. 3-41.

20. McCulloch, W., and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bulletin of Mathematical Biophysics, Vol. 5 (1943) pp. 115-133.

21. Metze, G., R. Miller and S. Seshu, "Transition matrices of sequential machines," IRE Transactions on Circuit Theory, Vol. CT-6 (1959) pp. 5-12.

22. Mealy, G., "Method for synthesizing sequential circuits," Bell Systems Tech. Jour., Vol. 34 (1955) pp. 1054-1079.

23. Moore, E., "Gedanken-Experiments on sequential machines," Automata Studies, Princeton University press, Princeton N. J., Study 34, pp. 129-153 (1956).

24. Nerode, A., "Linear automaton transformations," Proc. Amer. Math. Society, Vol. 9 (1958) pp. 541-544.

25. Rabin, M. O., and D. Scott, "Finite automata and their decision problems," IBM Jour. Research and Development, Vol. 3 (1959) pp. 114-125.

26. Unger, S., "Hazards and delays in asynchronous sequential switching circuits," IRE Transactions on Circuits Theory, Vol. CT-6 (1959) pp. 12-25.

27. Weeg, G., "Finite automata and connection matrices," Presented at the 15th National Conference of the Association for Computing Machinery, August 1960.