A Decision Procedure for Computations of Finite Automata*

JOYCE FRIEDMAN†

The Mitre Corporation, Bedford, Massachusetts

Abstract. The "computed output sequence" of a finite automaton is defined as the sequence which results from the output sequence when all occurrences of a special output symbol X are deleted. A "computation pair" consists of an input sequence and the resultant computed output sequence, and the "computation" of an automaton is the set of all its computation pairs. The class of infinite computations is broader than the class of behaviors of finite automata. Burks has therefore raised the question of the existence of a decision procedure to determine if two automata have the same computation. In this paper, such a decision procedure is given.

I. *Introduction*

We consider a finite deterministic synchronous automaton S with n states $q_0, q_1, q_2, \cdots, q_{n-1}$, m input symbols $s_1, s_2, \cdots, s_m$, and $p + 1$ output symbols $a_1, a_2, \cdots, a_p, X$. For each of the $n \times m$ state and input pairs (q_i, s_t) there is a unique state and output pair (q_j, α) where α is either $a_1, a_2, \cdots, a_p$ or the special symbol X, such that, if S is in state q_i and receives input s_t, it will enter state q_j and output the symbol α. We write these "transitions" of S as

$$(q_i, s_t) \rightarrow (q_j, \alpha). \tag{1}$$

By the output sequence of S for a given input sequence we mean the sequence of output symbols produced including X. By the *computed output sequence* of S we mean the sequence of output symbols after all occurrences of the special symbol X have been deleted.

II. *The Decision Problem*

PROBLEM[1]. Can we decide whether or not two automata S and T have the same computed output sequence for every infinite input sequence, i.e. are computationally equivalent?[2]

SOLUTION. We answer this question by providing an effective procedure which determines whether or not two automata are computationally equivalent. In the course of the procedure we make use of the transitions (1), i.e. we assume

* Received November, 1961

† The author wishes to thank A W Burks and R. F. Shackford for reading the manuscript and offering many constructive suggestions.

[1] This problem came to the author's attention through a paper by A. W. Burks [1].

[2] The decision problem for *all* input sequences reduces to the problem of *all finite* input sequences. This in turn reduces to the problem of deciding whether two automata are behaviorally equivalent This problem has previously been solved. See Burks and Wang [2], Friedman [4], Moore [5]

that the internal structure and behavior of the automaton or an equivalent automaton are known. However, it is possible to show that we could likewise proceed by behavioral tests only, provided that an upper bound on the number of (behaviorally) distinguishable states is known. For we could then observe the behavior of the automaton through a multiple experiment of finite length and then write the transitions for an equivalent automaton (Moore [5]).

III. *Definitions—Computed Output Sequences*

Before presenting the decision procedure we require definitions for computed output sequences, for states of an automaton and also some lemmas on sequences and on the behavior of a finite automaton.

Length of a Computed Output Sequence. By the *length* L of a computed output sequence we mean the number of occurrences of the output symbols a_1, $a_2, \cdots, a_p$. We note that for an input sequence of length T the corresponding computed output sequence will be of length $L \leqq T$.

Equality of Computed Output Sequences. If two computed output sequences are equal they must be both finite or both infinite. Two finite computed output sequences are equal if and only if they are of the same length L, and for every $k \leqq L$, the kth symbol of one is identical with the kth symbol of the other. Two infinite computed output sequences are equal if and only if for every k the kth symbol of one is identical with the kth symbol of the other.

Computational Equivalence. Two automata S and T are computationally equivalent ($S \equiv T$) if and only if for every infinite input sequence the computed output sequence of S is equal to the computed output sequence of T.

IV. *Definitions for States*

Connected States. State q_i *is connected to* q_j iff there exists an input sequence γ of length $\geqq 1$ such that if the automaton is in state q_i and is supplied with γ it will then be in state q_j. We say that q_i is connected to q_j by the input sequence γ.

Terminal State. A state q_i is *terminal* iff, starting with the automaton in state q_i, there exists at least one infinite input sequence γ such that there is no further computed output.

Finite State. A state q_i is *finite* iff it is terminal or is connected to some terminal state q_j. It follows from this definition that for any finite state q_i there exists at least one infinite input sequence such that the corresponding computed output sequence is of finite length.

Infinite State. A state q_i is *infinite* if it is not finite, that is if the automaton is in state q_i, then for every infinite input sequence the computed output sequence is infinite.

Initial State. The state q_0 is assumed to be the *initial* state of the automaton. That is if there have been no inputs, the automaton is in state q_0.

Admissible State. A state q_i is *admissible* iff either q_i is q_0 or q_0 is connected to q_i. Otherwise q_i is inadmissible.

V. *Sequences*

We shall have occasion to refer to sequences of input or output symbols. A few definitions and a lemma will be needed.

Null Sequence. The null sequence will be denoted by $\wedge$.

Concatenation of Sequences. If σ_1 is any finite sequence, and σ_2 is any sequence, we write $\sigma = \sigma_1\sigma_2$ to mean the sequence which consists of the sequence σ_1 followed by the sequence σ_2.

Ultimately Periodic Sequence. A sequence is ultimately periodic iff it is infinite and of the form $\sigma_1\sigma_2\sigma_2\sigma_2 \cdots$ where σ_1 and σ_2 are finite and σ_2 is not null. We write this sequence $\sigma_1\bar{\sigma}_2$. The *period* of $\sigma = \sigma_1\bar{\sigma}_2$ is the length of σ_2.

Reduced Form of an Ultimately Periodic Sequence. An ultimately periodic sequence $\sigma = \sigma_1\bar{\sigma}_2$ is in reduced form if $\sigma_1 = b_1b_2 \cdots b_k$ and

$$\sigma_2 = b_{k+1}b_{k+2} \cdots b_r,$$

where (1) $b_k \neq b_r$ and (2) σ_2 is a minimum period, i.e. σ_2 is not $\sigma_3\sigma_3 \cdots \sigma_3$ where σ_3 is shorter than σ_2.

LEMMA 1. *An ultimately periodic sequence has a unique reduced form.*

PROOF. Any ultimately periodic sequence can be put in reduced form. For suppose $\sigma = \sigma_1\bar{\sigma}_2$ where σ_2 is a repetition of some sequence σ_3. Then put $\sigma = \sigma_1\bar{\sigma}_3$. This can be repeated until the minimum period is obtained. If then

$$\sigma = b_1b_2 \cdots b_{k-1}b_k\overline{b_{k+1} \cdots b_{r-1}b_r}$$

where $b_k = b_r$, we can reduce to

$$\sigma = b_1b_2 \cdots b_{k-1}\overline{b_kb_{k+1} \cdots b_{r-1}}.$$

This absorption process can be repeated until we have either

$$\sigma = \sigma_4\bar{\sigma}_5 \quad \text{or} \quad \sigma = \bar{\sigma}_6,$$

where no further absorption is possible.

If any state $b_kb_{k+1} \cdots b_{r-1}$ is a repetition of a sequence $b_k \cdots b_l$, then at the previous stage $b_{k+1} \cdots b_r$ was a repetition of $b_{k+1} \cdots b_lb_k$. Thus, once a minimum period is obtained, the absorption process cannot affect the length of the period.

It is clear from the construction that the reduced form is unique.

VI. *Lemmas Concerning Behavior*

Lemmas 2, 3, and 4 are concerned primarily with behavior rather than computation. It is therefore likely that they have previously appeared in the literature in either this or similar form. See Burks and Wang [2], Burks and Wright [3], and Moore [5].

LEMMA 2. *If q_i is connected to q_j, then q_i is connected to q_j by an input sequence of length $\leqq n$, where n is the number of states.*

PROOF. We show that if q_i is connected to q_j by an input sequence γ_1, of length $L_1 > n$, it is also connected by γ_2 of length $L_2 < L_1$. It then will follow that it must be connected by an input sequence γ of length $L \leqq n$.

Consider the sequence of states corresponding to the input sequence γ_1. This sequence begins with q_i, ends with q_j and is of length $L_1 + 1$. Since $L_1 + 1 > n$ and there are only n states, some state q_k must occur more than once, say as the hth and $(h + l)$-th state $(l > 0)$, in the state sequence. Let $\gamma = \gamma_1\gamma_2\gamma_3$ where γ_1 is of length $h - 1$, γ_2 of length l, and γ_3 of length $L_1 - (l + h - 1)$. Then the input sequence $\gamma_1\gamma_3$ of length $L_2 = h - 1 + L_1 - (l + h - 1) = L_1 - l < L_1$ connects q_i to q_j Q.E.D.

LEMMA 3. *Terminal states can be identified.*

PROOF. Let the automaton be in state q_j, and consider all input sequences of length n and the corresponding state sequences (length $n + 1$) and computed output sequences (length $\leqq n$). Then state q_j is terminal if and only if at least one of the computed output sequences is null.

For suppose the input sequence γ of length n produces a null computed output sequence. Then since there are only n states, and the state sequence is of length $n + 1$, some state q_i must occur more than once, say as the kth and $(k + l)$-th state $(l > 0)$ in the state sequence. Write the input sequence γ as $\gamma_1\gamma_2\gamma_3$ where γ_1 is of length $(k - 1)$, γ_2 is of length l, and γ_3 is of length $(n - (k - 1) - l)$. Then the infinite (ultimately periodic) input sequence $\gamma_1\bar{\gamma}_2$ will produce a null computed output sequence.

The converse holds immediately, since any infinite input sequence must begin with one of the finite subsequences considered above.

LEMMA 4. *Finite states can be identified.*[3]

PROOF. Consider all input sequences of length n and the corresponding state sequences beginning with state q_j. Then q_j is finite iff a terminal state q_t occurs in at least one of the state sequences. For by lemma 2, q_j is connected to q_t iff it is connected by an input sequence of length $\leqq n$.

VII. *Computations of Finite Automata*

THEOREM 1. *Given an automaton S and an ultimately periodic sequence $\sigma = \sigma_1\bar{\sigma}_2$ of output symbols $a_1, a_2, \cdots, a_p$, there exists a procedure for deciding whether σ is computed for all infinite input sequences, if S starts in a given state q_{j_0}.*

PROOF[4]. By Lemma 4, we can determine if state q_{j_0} is finite. If so, then σ is not produced for all infinite input sequences.

[3] Burks and Wright [3] give an algorithm for deciding whether the computation of a fixed automaton is finite, infinite or mixed.

[4] This proof of Theorem 1 is based on a suggestion by A. W. Burks The proof may be visualized by drawing the state diagram and representing the transitions A by solid lines and those of B by dotted lines

Assume then that state q_{j_0} is infinite. Let the set of transitions from states admissible with respect to q_{j_0} of S be divided into two subsets A and B such that all transitions of A produce outputs $a_1, a_2, \cdots, a_p$ and those of B produce only X. We now construct from S an automaton S' which is computationally equivalent to S and which has only transitions which produce $a_1, a_2, \cdots, a_p$.

Let the admissible states of S be $q_1, q_2, \cdots, q_{n-1}$. Starting with $q = q_1$, if there are in B one or more transitions to state q, $(q_i \rightarrow q, X)$, and there are transitions in A from q, $(q \rightarrow q_k, a)$, then form and add to A all the transitions $(q_i \rightarrow q_k, a)$. If there are also in B transitions from q, $(q \rightarrow q_l, X)$, add the transitions $(q_i \rightarrow q_l, X)$ to B. (This cannot be $(q_i \rightarrow q_i, X)$, else q_i is inadmissible or q_{j_0} is finite.) Then delete from B the transitions to q. The resulting transitions of A and B describe an automaton which is computationally equivalent to S. Continue for $q_2, \cdots, q_{n-1}$. Once the transitions in B to a state q have been deleted, no further transitions to q are ever introduced into B. Hence, when all q_i have been considered, B is empty and the resulting transitions in A describe a computationally equivalent finite automaton S', which has an output $a_1, a_2, \cdots$, or a_p at every transition. From finite automata theory it is known that any periodic *solution* must have a period and phase of total length no more than the number of states of the automaton. Consequently one can decide not only whether the given periodic sequence σ is produced by all input sequences to S', but also whether there is any ultimately periodic sequence produced by all input sequences. Q.E.D.

VIII. *Joint Automaton ST*

Let S and T be automata with n_s and n_t states respectively and having in common m input symbols and $p + 1$ output symbols $a_1, a_2, \cdots, a_p$ and X. Form the combined automaton ST whose $N = n_s \times n_t$ states q_i are pairs of states (q_i^s, q_i^t) of S and T and whose output is an ordered pair (α_i, α_j). The initial state q_0 of ST is (q_0^s, q_0^t). The transitions of ST are obtained from those of S and T as follows: If $(q_{i_s}, s_\mu) \rightarrow (q_{j_s}, \alpha_1)$ is a transition of S and $(q_{i_t}, s_\mu) \rightarrow (q_{j_t}, \alpha_2)$ is a transition of T then $(q_i, s_\mu) \rightarrow (q_j, (\alpha_1, \alpha_2))$ is a transition of ST, where $q_i = (q_{i_s}, q_{i_t})$ and $q_j = (q_{j_s}, q_{j_t})$.

The computed output of ST is an ordered pair of sequences σ_s and σ_t of the symbols $a_1, a_2, \cdots, a_p$. We write this (σ_s, σ_t).

Value of the Computed Output of ST. Let the computed output (σ_s, σ_t) be expressed as $(\sigma\sigma_1, \sigma\sigma_2)$, where the leading symbol (if any) of σ_1 differs from the leading symbol (if any) of σ_2. Then the *value* v of the joint computation is (σ_1, σ_2). We may, of course, have $v = (\wedge, \wedge)$.

THEOREM 2. *If there is some finite input sequence γ to ST for which the value of the computed output is (σ_1, σ_2) where neither σ_1 nor σ_2 is null, then S and T are not equivalent.*

PROOF. If the value of the computed output is (σ_1, σ_2) then the computed output itself is $(\sigma\sigma_1, \sigma\sigma_2)$ where σ is a string of finite length $L \geqq 0$. Then the computed output sequences of S and T for γ differ in the $(L+1)$-th place and are thus not equal.

THEOREM 3. *If the initial state q_0 is connected to any state $q_j = (q_j^s, q_j^t)$ by several different input sequences for which the corresponding computed output pairs have two or more distinct values, then from these values we can determine either (1) that $S \not\equiv T$ or (2) that $S \equiv T$ only if S starting at state q_j^s computes an ultimately periodic sequence σ_s and T similarly computes σ_t, where σ_s and σ_t can be determined from the values.*

PROOF.

1. If any of the values are (σ_1, σ_2) where $\sigma_1 \neq \wedge$ and $\sigma_2 \neq \wedge$, then $S \equiv T$ by the previous theorem.

2. Consider any two values. They must then be either $(\wedge, \wedge)$ and $(\sigma, \wedge)$ or $(\sigma_1, \wedge)$ and $(\sigma_2, \wedge)$ or $(\sigma_1, \wedge)$ and $(\wedge, \sigma_2)$ or cases symmetric to these in S and T. We take each pair in turn.

(a) The values at q_j are $(\wedge, \wedge)$ and $(\sigma, \wedge)$, corresponding to γ_1 and γ_2 respectively, $(\sigma \neq \wedge)$. Then by construction there exist output strings σ_1 and σ_2 such that the computed outputs for γ_1 and γ_2 are respectively (σ_1, σ_1) and $(\sigma_2\sigma, \sigma_2)$. Let γ be any infinite input string. Then $\gamma_1\gamma$ produces a computation $(\sigma_1\sigma_s, \sigma_1\sigma_t)$ and $\gamma_2\gamma$ produces $(\sigma_2\sigma\sigma_s, \sigma_2\sigma_t)$. Then a necessary condition for equivalence of S and T is that $\sigma_1\sigma_s = \sigma_1\sigma_t$, hence $\sigma_s = \sigma_t$, and $\sigma_2\sigma\sigma_s = \sigma_2\sigma_t$, hence $\sigma\sigma_s = \sigma_t$. But this will be satisfied iff $\sigma_s = \sigma_t = \bar{\sigma}$.

(b) Similarly for $(\sigma_1, \wedge)$ and $(\sigma_2, \wedge)$ we have that for all input sequences γ starting at state q_j the sequences σ_s and σ_t must be produced where:

$$\sigma_1\sigma_s = \sigma_t$$

$$\sigma_2\sigma_s = \sigma_t .$$

If σ_1 and σ_2 are of the same length, S and T are not equivalent since σ_t must begin both with σ_1 and with σ_2. Let σ_1 be longer than σ_2. Then we must have $\sigma_1 = \sigma_2\sigma_3$ for some finite non-null σ_3. But this gives

$$\sigma_2\sigma_3\sigma_s = \sigma_t$$

$$\sigma_2\sigma_s = \sigma_t$$

which gives

$$\sigma_2\sigma_3\sigma_s = \sigma_2\sigma_s, \qquad \text{so that} \qquad \sigma_s = \bar{\sigma}_3$$

$$\sigma_t = \sigma_2\sigma_s, \qquad \text{so that} \qquad \sigma_t = \sigma_2\bar{\sigma}_3 .$$

(c) Similarly $(\sigma_1, \wedge)$ and $(\wedge, \sigma_2)$ gives

$$\sigma_1\sigma_s = \sigma_t$$

$$\sigma_s = \sigma_2\sigma_t$$

or

$$\sigma_s = \sigma_2\sigma_1\sigma_s$$

$$\sigma_t = \sigma_1\sigma_2\sigma_t$$

so that

$$\sigma_s = \overline{\sigma_2\sigma_1}$$

$$\sigma_t = \overline{\sigma_1\sigma_2}\,.$$

Thus, from any two of the values it can be determined that for all infinite input sequences starting at q_j the joint computed output of the form $(\sigma_1\bar{\sigma}_3, \sigma_2\bar{\sigma}_4)$ must be produced (or else $S \not\equiv T$).

3. Suppose now there are more than two values, that is, there is a finite input sequence γ connecting q_0 to q_j, such that the value of the corresponding joint computed output is (σ_5, σ_6).

Then a necessary and sufficient condition that no infinite input sequence beginning with γ will produce unequal computations for S and T is that

$$\sigma_5\sigma_1\bar{\sigma}_3 = \sigma_6\sigma_2\bar{\sigma}_4\,.$$

This is clearly decidable by putting both sides into the reduced form for ultimately periodic sequences. Q.E.D.

IX. *Decision Procedure*

In the decision procedure we construct for each state q_k of ST a set V_k of values and give three conditions which the sets V_k must satisfy. Each of these conditions is shown to be necessary. Finally, it will be proved that the conditions are sufficient: if each of the sets V_k meets all three conditions, then S and T are computationally equivalent.

Construction of the Sets V_k. We construct for each (admissible) state q_k of ST a set of values V_k such that if $v \in V_k$ then there exists a finite input sequence γ such that if ST starts in the initial state q_0 and γ is input, the resulting state is q_k and the value of the computed output sequence is v.

Let $v_0^0 = (\wedge, \wedge)$. For each admissible state q_k of ST, $k > 0$, let v_k^0 be the value of a joint computed output obtained by some input sequence γ_k, for which the corresponding state sequence starting at initial state q_0 ends in q_k. By Lemma 2 we can select some $v_j^0 = (\sigma_1, \sigma_2)$ where both σ_1 and σ_2 are of length $\leqq N$, where N is the number of states of ST.

Form the sets V_k as follows:

(1) For all k, $v_k^0 \in V_k$.

(2) If $[(q_j, s_r) \to (q_k, (\alpha_1, \alpha_2))]$ is a transition of ST, and $v_j^0 = (\beta_1, \beta_2)$, then the value of $(\beta_1\alpha_1, \beta_2\alpha_2)$ is in V_k.

Then it is clear that if $(\sigma_1, \sigma_2) \in V_k$ the value (σ_1, σ_2) occurs at q_k for either the input sequence γ_k of step (1) or $\gamma_j s_r$ for some j and r.

Likewise, it follows from the construction that if $(\sigma_1, \sigma_2) \in V_k$ then σ_1 and σ_2 are each of length $\leqq N + 1$, because v_j^0 is of length at most N

Necessary Conditions. Having constructed the sets V_k, we can now state three conditions for $S \equiv T$. These conditions will be given and each one proved to be necessary. It will then be shown that together they are sufficient. That is, $S \equiv T$ iff conditions 1, 2, and 3 are satisfied.

CONDITION 1. If for any k, V_k contains a value (σ_1, σ_2) in which neither σ_1 nor σ_2 is null, S is not equivalent to $T (S \not\equiv T)$. The necessity of this condition follows from Theorem 2.

CONDITION 2. If for any terminal state q (which can be identified by Lemma 3), V_k contains a value $(\sigma, \wedge)$ or $(\wedge, \sigma)$ where $\sigma \neq \wedge$, then $S \not\equiv T$. For there exists an infinite input sequence with no further computed output so that the computed output sequences of S and T are of unequal length and hence unequal.

Definition. Using the sets V_k, we distinguish two classes of states of ST: Class I states q_k for which V_k has only one distinct value and Class II states for which V_k has two or more distinct values. (The only states of ST not in either of these classes are inadmissible states, which can never be reached.)

CONDITION 3. If V_k is a class II state, then by Theorem 3 it is a necessary condition for $S \equiv T$ that certain specified ultimately periodic sequences $\sigma_s = \sigma_1\bar{\sigma}_2$ and $\sigma_t = \sigma_3\bar{\sigma}_4$ be produced by every infinite input sequence from state $q_k = (q_k^s, q_k^t)$. Theorem 1 provides a method for determining if this condition is met. Since ultimately periodic sequences are by definition infinite, it follows that if Condition 3 is met, then all Class II states are infinite for both S and T.

One further theorem is needed before the sufficiency of these conditions can be proved.

THEOREM 4. *Consider an arbitrary finite input sequence γ and suppose that having started initially in state q_0, state q_j has been reached through the state sequence $q_0 \cdots q_i q_j$ and that the value of the computed output sequence is v. Then either $v \in V_j$ or the state sequence $q_0 \cdots q_i$ has included at least one Class* II *state.*

PROOF BY INDUCTION. Either q_0 is of Class II or it has only one value v_0. Suppose that after a finite number t of inputs ST is in q_l with a value $v \in V_l$. Let the $(t + 1)$-th input be s_r so that the transition is $(q_l, s_r) \rightarrow (q_\mu, \alpha)$ for some μ. If q_l is in Class I, then $v = v_l$ and hence by the construction of V_μ, the resulting value is in V_μ. Otherwise q_l is in Class II.

PROOF OF SUFFICIENCY. We now prove that the procedure is sufficient, that is if conditions 1, 2, and 3 are met, S and T are computationally equivalent. To do this, we must show that if the conditions are met, the computations of S and T are both finite or both infinite and that they are equal.

1. *There is no infinite input sequence for which the computed output sequence of S is finite and that of T infinite or vice versa.*

For suppose there were an input sequence γ such that after t inputs the complete finite computed output sequence of S had been produced and ST was in state q_i. Clearly q_i is terminal for S. Suppose the value of the joint computed output is v. Then $v \in V_i$, for by Theorem 4 the computed output sequence at q_i can have no values other than those of V_i unless the state sequence has included a Class II state. But all Class II states are infinite for both S and T and therefore cannot be connected to the state q_i which is a terminal state of S.

Now since the computed output sequence of T is infinite for the input sequence γ, after some number t_2 of inputs its length will exceed its length after t inputs by $2(N + 1)$. Thus, after t_2 inputs, ST will be in some state q_j with a value $(\wedge, \sigma)$ where σ is of length $>N + 1$. But all values in V_j are of length $\leqq N + 1$, so that $v \notin V_j$ and thus a Class II state must have occurred in the state sequence. But this is impossible since all Class II states are infinite for both S and T.

2. *If for some infinite input sequence the computed output sequences are both finite, they are equal.*

For after the computed output sequences are complete ST is always in some state q_i which is terminal for both S and T. Thus, no Class II state was included in the state sequence, and hence, by Theorem 4, the value of the joint computed output must be in V_i. But by Condition 2, V_i contains only $(\wedge, \wedge)$. So the joint computed output is (σ, σ).

3. *If for some infinite input sequence the computed output sequences are both infinite, they are equal.*

If the state sequence for some input sequence contains any Class II states, a first Class II state q_i is reached by a sub-sequence γ_1, with a value $v \in V_i$. Then by Condition 3 it has been determined that for any infinite input sequence beginning with $\gamma_1\gamma$ the computed output sequences are the same.

If, on the other hand, the state sequence contains only Class I states, the computed output sequences are again equal. For if not equal, a value $v = (\sigma_1, \sigma_2)$ where neither σ_1 nor σ_2 is null would occur. But by Condition 1, this v is not in any V_i and by Theorem 4 no new values are ever introduced by a sequence of Class I states only. Q.E.D.

REFERENCES

1. BURKS, ARTHUR W Computation, behavior, and structure in fixed and growing automata. *Behavioral Sci 6* (1961), 5–22.
2. BURKS, ARTHUR W , AND HAO WANG. The logic of automata. *J. ACM 4* (1957), 193–218 and 279–297.
3. BURKS, ARTHUR W., AND WRIGHT, JESSE B. Sequence generators and digital computors. University of Michigan, Ann Arbor (Feb. 1961)
4. FRIEDMAN, JOYCE. Some results in Church's restricted recursive arithmetic, *J. Symbolic Logic 22*, (1957), 337–342.
5. MOORE, EDWARD F. Gedanken-experiments on sequential machines. In C. E. Shannon and J. McCarthy (Eds), *Automata Studies*, pp. 129–153, Princeton Univ. Press, 1956.