Linear Regular Languages. Part I[1]

By G. T. HERMAN

1. Introduction

In this paper we take a language to be any set of words (finite strings) ove a finite alphabet. We call those languages regular which can be described by a specia kind of finite expressions, the so called regular expressions. (Using a different terminology a language is regular if, and only if, it is right linear context free.)

Regular languages have a special significance from an engineering point of view, because they can be used to describe the behaviour of sequential circuits, which are often used in electrical engineering. Conversely, given a regular expression R one can design a binary sequential circuit which will in a certain sense accept amongst all the possible words over the alphabet only those which belong to the language described by R. In such a case we say that the circuit accepts the language described by R.

Certain subclasses of the class of regular languages are distinguished by the fact that the languages belonging to them can be accepted by some special kind of sequential circuits (e.g. feedback free).

In recent years there has been a great upsurge of interest in (binary) linear sequential circuits (i.e. circuits which use only unit delays and exclusive or gates). If a language is accepted by some linear sequential circuit, we call it a linear regular language.

In this paper we give a rigorous introduction to the concepts mentioned above and discuss the following problem: Given a regular expression R, how can we decide whether or not the language described by R is a linear regular language?

2. Regular Expressions

Definition 1. An *alphabet* S is a finite non-empty set of symbols.

Definition 2. Any finite string of symbols from an alphabet S is called a *word* over S. The set of all words over S is denoted by I_S. The empty word is denoted by e ($e \notin S$, $e \in I_S$).

[1] This paper has been presented at the International Colloquium on Mathematical Linguistics, Balatonszabadi, Hungary, Sept., 1968.

Definition 3. Any subset W of I_S is called a *language* over S. The set of languages over S is denoted by L_S.

Definition 4. The set E_S of *regular expressions* over the alphabet $S=\{s_1, s_2, \ldots s_l\}$ is defined as follows:

(i) $s_1, s_2, \ldots, s_l, e, o$ are regular expressions ($o \notin S$).
(ii) If P and Q are regular expressions, so are $(P+Q)$, (PQ) and P^*.
(iii) All regular expressions are words over the alphabet $S \cup \{e, o, (, +,), ^*\}$, but only those words are regular expressions which can be shown to be such using (i) and (ii).

Definition 5. An *interpretation* of regular expressions over S is a function I: $E_S \rightarrow L_S$, defined by (for simplicity we denote $I(R)$ by $|R|$):

(i) $|s_i| = \{s_i\} \qquad i=1, 2, \ldots, l,$

$|e| = \{e\},$

$|o| = \emptyset;$

(ii) $|(P+Q)| = |P| \cup |Q|,$

$|(PQ)| = \{pq;\ p \in |P| \ \&\ q \in |Q|\},$

$|P^*| = \bigcup_{n=0}^{\infty} \{p_1 p_2 \cdots p_n;\ p_i \in |P| \text{ for } 1 \leqq i \leqq n\}.$

Definition 6. A language W over S is called *regular* if, and only if, there exists a regular expression R over S such that

$$W = |R|.$$

Example 1. Let $S=\{a, b\}$. The set of all words which contains exactly two b's is regular. The regular expression which describes it is

$$(((a^*b)(a^*b))a^*).$$

Example 2. Let $S=\{a, b\}$. The set of all words which contain an even number (possibly none) of b's is regular. The regular expression which describes it is

$$(a^*((ba^*)(ba^*))^*).$$

3. Linear Sequential Circuits

Definition 7. The *clock* is a device which emits pulses at regular intervals to which we shall refer to as *units of time*.

A *delay* is a device with one input and one output wire such that it will emit a pulse on its output wire if, and only if, a pulse has been received at its input wire exactly one unit of time ago.

An *exclusive or gate* is a device with two input wires and one output wire, such that it will emit a pulse on its output wire if, and only if, it is receiving a pulse at exactly one of its input wires at the same time.

The existence of a pulse in a wire is denoted by 1, the lack of pulse by 0 ($0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$ and $1 \oplus 1 = 0$).

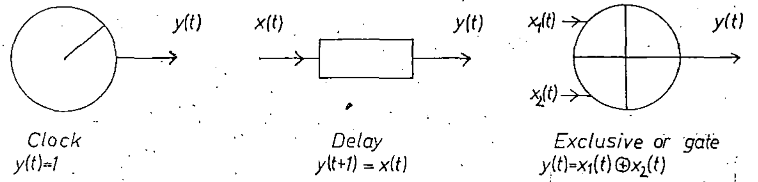

Fig. 1

Definition 8. A *linear sequential circuit* is a network formed from delays and exclusive or gates according to the following rules.

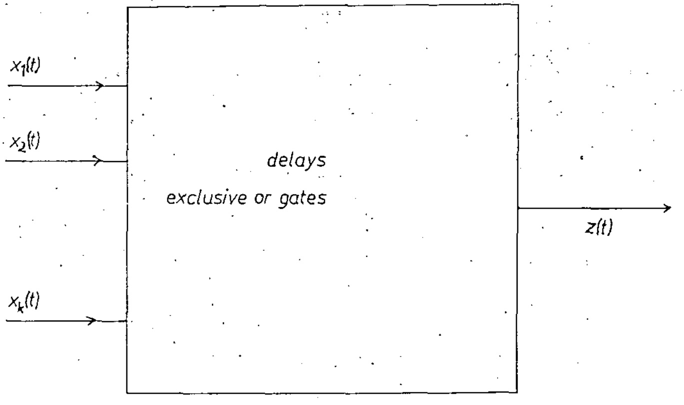

Fig. 2

The network will have a certain number (k, say) of external input wires and one external output wire. There is a clock, all wires may (or may not) carry pulses at the time when the clock emits a pulse, but will carry pulses at no other time. The propogation of pulses within wires is assumed to be instantaneous and hence the whole circuit is synchronized to the clock.

The external output wire and the input wires of any delay or exclusive or gate can be connected to any external input wire or the output wire of any delay or exclusive or gate provided only that the following restrictions are satisfied.

1. If there is a closed (feedback) path within the network (i.e. a point which we can get back to going along connected wires), it must go through at least one delay.

2. If there is a path from any external input wire to the external output wire, it must go through at least one delay.

Example 3. The following is a representation of a linear sequential circuit

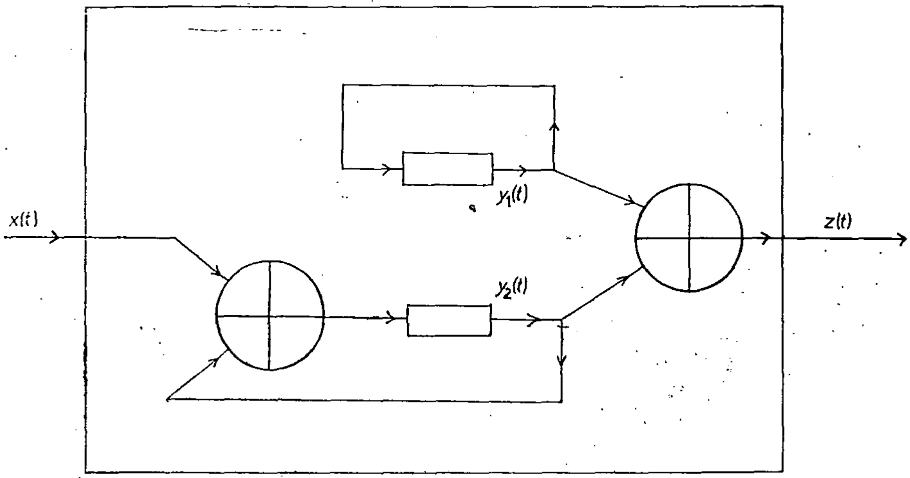

Fig. 3

Definition 9. Let W be a language over the alphabet S. W is called a *linear regular language* if there exists a positive integer k, a linear sequential circuit with k external inputs and a function f from S into k-tuples of 0's and 1's satisfying the following conditions.

For each symbol s of S we denote by s_j the jth element of $f(s)$ $(1 \leqq j \leqq k)$. This way each symbol s of S determines an external input condition for the linear sequential circuit, namely the jth external input wire carries a pulse if, and only if, $s_j = 1$.

A word $w = w_1 w_2 \ldots w_n$ belongs to W if, and only if, the following is true. If

(i) at time 1 the output of the first delay is 1 and of all other delays is 0,

and (ii) the external input condition to the circuit at time t is determined by w_t for $1 \leqq t \leqq n$,

then (iii) the external output at time $n+1$ is 1.

In such a case we say that the circuit *accepts* w.

Example 4. Let W be the language of Example 2. W is a linear regular language. This is true, because if we let $k=1$, the linear sequential circuit to be that of Example 3, and f be defined by

$$f(a)=0, \qquad f(b)=1,$$

then the conditions of Definition 9 are satisfied.

In order to see this let us mark the external input, the outputs of the delays and the external output at time t by $x(t)$, $y_1(t)$, $y_2(t)$ and $z(t)$, respectively. Then

$$z(t) = y_1(t) \oplus y_2(t) = 1 \oplus y_2(t)$$

(therefore $z(t)=1$ if, and only if, $y_2(t)=0$),

$$y_2(t+1) = x(t) \oplus y_2(t)$$

and $$y_2(1)=0.$$

So $y_2(t)=0$ initially and it changes value at time $t+1$ if, and only if, $x(t)=1$. So it accepts w if, and only if, $w \in W$.

4. The Relationship between Linear Regular and Regular Languages

Linear regular and regular languages have been defined in entirely different ways. In this section we show that every linear regular language is in fact regular, but there are regular languages which are not linear regular. In order to do this we need to establish some properties of linear sequential circuits.

Definition 10. Let C be a linear sequential circuit which has n delays. We number these delays from 1 to n, and let $y_i(t)$ denote the state of the output wire from the ith delay at time t ($y_i(t)=0$ means no pulse, $y_i(t)=1$ means pulse). Then the n-tuple $(y_1(t), y_2(t), \ldots, y_n(t))$ is said to be the *state* of the circuit C at time t.

Theorem 1. The operation of a linear sequential circuit C with k external input wires and n delays can be completely described by two matrix equations of the form

$$y(t+1) = y(t)\boldsymbol{A} \oplus x(t)\boldsymbol{B} \tag{1}$$

$$z(t) = y(t)\boldsymbol{C} \tag{2}$$

where $x(t)$ is the k-tuple of inputs at time t, $y(t)$ is the state at time t, $z(t)$ is the output at time t, $\boldsymbol{A}$ is a $n\times n$ matrix, $\boldsymbol{B}$ is a $k\times n$ matrix, and $\boldsymbol{C}$ is an $n\times 1$ matrix of 0's and 1's. All additions are to be performed modulo 2.

Proof. We deal with the second equation first. Starting from the external output wire we go backwards along all possible paths. When we come to an exclusive or gate the path splits into two. A path comes to an end when it reaches the output wire of a delay. Because of the restrictions in Definition 8 this must happen to every path. It is clear that the output is the number of paths (modulo 2) which finish at a wire which carries a pulse at the time. The output wire of a delay can modify this sum if, and only if, the number of paths leading to it is odd. So the construction of the circuit uniquely determines the value of $\boldsymbol{C}$.

Since the output of a delay is equal to its input at one unit of time ago, analogous argument shows that $\boldsymbol{A}$ and $\boldsymbol{B}$ are uniquely determined.

It is clear that the two equations completely describe what external output will result from a given sequence of inputs. In fact

$$z(t+1) = (1, 0, \ldots, 0)\boldsymbol{A}^t\boldsymbol{C} \oplus \sum_{i=1}^{t} x(i)\boldsymbol{B}\boldsymbol{A}^{t-i}\boldsymbol{C} \tag{3}$$

as can easily be shown by induction on t.

Example 5. The circuit in Example 3 can be characterized by the equations

$$(y_1(t+1), y_2(t+1)) = (y_1(t), y_2(t))\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \oplus x(t)[0, 1],$$

$$z(t) = (y_1(t), y_2(t))\begin{bmatrix}1\\ 1\end{bmatrix}.$$

Therefore,

$$z(t+1) = (1,0)\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}^t\begin{bmatrix}1\\1\end{bmatrix} \oplus \sum_{i=1}^{t} x(i)[0,1]\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}^{t-i}\begin{bmatrix}1\\1\end{bmatrix} = \sum_{i=1}^{t} x(i) \oplus 1,$$

showing again that this circuit accepts the language described in Example 2.

Theorem 2. Every linear regular language is regular.

Proof. (Although this theorem is the consequence of a well known theorem in automata theory, here we give a direct proof, which is conceptually considerably simpler than the proof of the general theorem and which also has interesting consequences.)

Let W be the linear regular language over S and C be the linear sequential circuit which accepts it using the function f from S into k-tuples of of 0's and 1's. For C there will be matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ as described in Theorem 1. In particular equation (3) will hold.

$\boldsymbol{A}$ is an $n\times n$ matrix with elements 0 or 1. There are only finitely many such matrices and there must be positive integers u and v such that $u<v$ and $\boldsymbol{A}^u=\boldsymbol{A}^v$. Choose u and v the smallest such integers and let $d = v-u$. If $t = qd+u+r$ where $0\leqq r<d$, then

$$\begin{aligned} z(t+1) = {} & \sum_{i=1}^{d} x(i)\boldsymbol{B}\boldsymbol{A}^{v+r-i}\boldsymbol{C} \oplus \\ & \oplus \sum_{i=1}^{d} x(d+i)\boldsymbol{B}\boldsymbol{A}^{v+r-i}\boldsymbol{C} \oplus \\ & \dots \\ & \oplus \sum_{i=1}^{d} x((q-1)d+i)\boldsymbol{B}\boldsymbol{A}^{v+r-i}\boldsymbol{C} \oplus \\ & \oplus \sum_{i=1}^{u+r} x(qd+i)\boldsymbol{B}\boldsymbol{A}^{u+r-i}\boldsymbol{C} \oplus (1,0,\dots,0)\boldsymbol{A}^{u+r}\boldsymbol{C}. \end{aligned}$$

Let P_0 be o if $c_1=0$ and P_0 be e if $c_1=1$. Let $P_1, P_2, \dots, P_p$ be all the regular expressions over S of the form $(\dots(w_1 w_2)\dots w_t)$ $(w_i\in S)$ for which $t<u$ and

$$\sum_{i=1}^{t} f(w_i)\boldsymbol{B}\boldsymbol{A}^{t-i}\boldsymbol{C} \oplus (1,0,\dots,0)\boldsymbol{A}^t\boldsymbol{C} = 1.$$

It is in principle easy to enumerate all such expressions. Let P be the regular expression

$$(\dots((P_0+P_1)+P_2)+\dots+P_p).$$

For $r=0,1,\dots,d-1$ let $X_{r1},\dots,X_{rx_r}$ be all regular expressions of the form $(\dots(w_1w_2)\dots w_{u+r})$ for which

$$\sum_{i=1}^{u+r} f(w_i)\boldsymbol{B}\boldsymbol{A}^{u+r-i}\boldsymbol{C} \oplus (1,0,\dots,0)\boldsymbol{A}^{u+r}\boldsymbol{C}$$

is equal to 0, and $Y_{r1},\ldots, Y_{ry_r}$ be all such expressions for which the same sum is equal to 1.

Let $F_{r1},\ldots, F_{rf_r}$ be all regular expressions of the form $(\ldots(w_1 w_2)\ldots w_d)$ for which

$$\sum_{i=1}^{d} f(w_i)\boldsymbol{B}\boldsymbol{A}^{v+r-i}\boldsymbol{C}$$

is equal to 0, and $G_{r1},\ldots, G_{rg_r}$ be all such expressions for which the same sum is eqalu to 1.

Let X_r be the regular expression

$$(\ldots(X_{r1}+X_{r2})+\ldots+X_{rx_r})$$

or o if $x_r=0$, and let Y_r, F_r and G_r be similarly defined.

Let D_r be the regular expression

$$(((F_r^*((G_rF_r^*)(G_rF_r^*))^*)(G_rF_r^*))X_r)$$

and E_r be the regular expression

$$((F_r^*((G_rF_r^*)(G_rF_r^*))^*)Y_r).$$

Then the regular expression

$$(P+(\ldots(D_0+E_0)+\ldots+(D_{d-1}+E_{d-1})))$$

describes W.

Example 6. For the circuit of Example 3, $u=1$, $v=2$, $d=1$. Looking at Example 5 we see that

P_0 is e,	$p=0$,	P is e;	
	$x_0=1$,	X_{01} is b,	X_0 is b;
	$y_0=1$,	Y_{01} is a,	Y_0 is a;
	$f_0=1$,	F_{01} is a,	F_0 is a;
	$g_0=1$,	G_{01} is b,	G_0 is b;

$$D_0 \text{ is } (((a^*((ba^*)(ba^*))^*)(ba^*))b),$$

$$E_0 \text{ is } ((a^*((ba^*)(ba^*))^*)a).$$

Hence a regular expression which describes the language accepted by this circuit is

$$(e+((((a^*((ba^*)(ba^*))^*)(ba^*))b)+((a^*((ba^*)(ba^*))^*)a))).$$

A much simpler expression which describes the same language has been given in Example 2.

Theorem 3. Not every regular language is linear regular.

Proof. The language W described in Example 1 is regular, but it is not linear regular.

To see this we have to look at the proof of Theorem 2.

Suppose W is accepted by some linear sequential circuit. Let w be the word

$$bba^v;$$

then $w \in W$. Since

$$W = |(P+(\ldots(D_0+E_0)+\ldots+(D_{d-1}+E_{d-1})))|,$$

$w \in |P|$ or $w \in |D_r|$ or $w \in |E_r|$ for some r.

However, $w \notin |P|$, since it is too long.

If $w \in |D_r|$, then

$$bba^{v-r} \in |(((F_r^*(G_rF_r^*)(G_rF_r^*))^*)(G_rF_r^*))|$$

and so b must occur in some word belonging to $|G_r|$ or $|F_r|$. But then there must be words belonging to W in which b appears more than twice, contradicting the definition of W.

The case $w \in |E_r|$ is analogous.

5. Characterization of Linear Regular Languages

In the last section we have shown that every linear regular language can be described by a regular expression of a certain type. Now we consider the problem: Given a regular expression R, how can we decide whether or not the language described by R is a linear regular language?

Unfortunately the decision method used is quite complicated, it will form Part II of this paper. Its essence is the checking the linearity of the languages over k-tuples of 0's and 1's which are the images of $|R|$ under some functions f mapping S into such k-tuples.

The method is not only indirect, but is needs the introduction of a number of new ideas. It seems likely that there is a more direct solution, something along the lines of the proof of Theorems 2 and 3. The kind of question which we ought to try to answer is: What do regular expressions which describe linear regular languages look like?

It was mentioned in the introduction that a language is regular if, and only if, it can be described by a right linear context free (Chomsky) grammar. A similar question to the above is: What are the characteristics of those grammars which describe linear regular languages?

6. Final Comments and Acknowledgements

There is very little in this first part of the paper which has not already appeared elsewhere is one form or another. However this is the first time that the results have been discussed from the point of view of linear regular languages, and the novel approach bore fruit in the proofs of Theorems 2 and 3.

The algorithm given in Theorem 2 for analyzing linear sequential circuits works faster than any other algorithm published so far for this purpose. This is not really surprising since other published algorithms can be used to analyse sequential circuits in general, while our algorithm makes essential use of the linearity of the sequential circuit under consideration.

Because the paper is introductory, all definitions and results have been fully explained and references to original sources (sometimes difficult to ascertain) have been omitted. The bulk of the author's knowledge is based upon lecture courses given at the University of California by Professors J. A. BRZOZOWSKI and M. A. HARRISON. His indebtedness to both these gentlemen is great.

A comprehensive list of references is given, which includes many original papers on the topic under discussion including those which have been used for Part II. The author apologizes for the possible omission of relevant papers, and hopes that they are referenced in the literature cited below.

Most of these papers refer to the closely related problem of the linearity of sequential machines. A good critical review of the work done on this problem can be found in DAVIS [10].

The author wishes to invite any comments relevant to the problems discussed here.

IBM (UK) LTD.,
EDUCATION CENTRE
SUDBURY TOWERS, GREENFORD RD.
GREENFORD, MIDDX., ENGLAND.

PRESENT ADDRES:
DEPT. OF COMPUTER SCIENCE
STATE UNIVERSTY OF NEW YORK
AT BUFFALO.

7. References

[1] BRZOZOWSKI, J. A., A survey of regular expressions and their applications, *IRE Trans.*, v. EC-11, 1962, pp. 324—335.
[2] —, Derivatives of regular expressions, *J. Assoc. Comp. Mach.*, v. 11, 1964, pp. 481—494.
[3] —, Regular expressions for linear sequential circuits, *IEEE Trans.*, v. EC-13, 1964, pp. 148—156.
[4] — & W. A. DAVIS, On the linearity of autonomous sequential machines, *IEEE Trans.*, v. EC-13, 1964, pp. 673—679.
[5] — & —, On the linearity of sequential machines, *IEEE Trans.*, v. EC-15, 1966, pp. 21—29.
[6] COHN, M., A theorem on linear automata, *IEEE Trans.*, v. EC-13, 1964, p. 52.
[7] — & S. EVEN, Identification and minimization of linear machines, *IEEE Trans.*, v. EC-14, 1965, pp. 367—376.
[8] CURTIS, H. A., Polylinear realizations of finite automata, *IEEE Trans.*, v. CT-17, 1968, pp. 251—259.
[9] DAVIS, W. A. Linear realizations for permutation machines, *IFIP Congress 68*, Edinburgh, 1968. pp. A 140—A 145.
[10] —, The linearity of sequential machines: a critical review, *9th Annual Symposium on Switching & Automata Theory*, 1968, pp. 427—430.
[11] DOTY, K. & H. FRANK, A theorem on linearity, *IEEE Trans.*, v. CT-17, 1968, pp. 270—272.
[12] ELSPAS, B., The theory of autonomous linear sequential networks, *IRE Trans.*, v. CT-6, 1959, pp. 45—60. Also in [25].
[13] GALLAIRE, H., J. N. GRAY, M. A. HARRISON & G. T. HERMAN, Infinite linear sequential machines, *Journal of Computer and System Sciences*, v. 2, 1968, pp. 381—419.
[14] GILL, A., Analysis and synthesis of stable linear sequential circuits, *J. Assoc. Comp. Mach.* v. 12, 1965, pp. 141—149.
[15] —, The minimization of linear sequential circuits, *IEEE Trans.*, v. CT-12, 1965, pp. 292—294.
[16] —, *Linear sequential circuits*, McGraw Hill, N. Y. 1966.
[17] —, Synthesis of linear sequential circuits from input-output relations, *SIAM J. Appl. Math.* v. 16, 1968, pp. 216—227.
[18] —, Graphs of affine transformation, with applications to sequential circuits, *7th Annual Symposium on Switching and Automata Theory*, 1966, pp. 127—135.
[19] — & C. J. TAN, The factorization of linear cycle sets, *IEEE Trans.*, v. CT-12, 1965, pp. 630—632.

[20] Gray, J. N. & M. A. Harrison, The theory of sequential relations, *Inform. Control,* v. 9, 1966, pp. 435—468.
[21] Harrison, M. A., *Introduction to switching and automata theory,* McGraw-Hill, N. Y., 1965.
[22] Hartmanis, J., Two tests for the linearity of sequential machines. *IEEE Trans.,* v. EC-14, 1965, pp. 781—786.
[23] — & W. A. Davis, Homomorphic images of linear sequential machines, *Journal of Computer and System Sciences,* v. 1, 1965, pp. 155—165.
[24] — & R. E. Stearns, *Algebraic structure theory of sequential machines,* Prentice-Hall, N. J., 1966.
[25] Kautz, W. H. (ed), *Linear sequential switching circuits: selected technical papers,* Holden-Day, San Francisco, 1965.
[26] Kleene, S. C., Representation of events in nerve nets and finite automata, *Automata Studies,* Princeton, N. J., 1965, pp. 3—41.
[27] Marino, P. J., Identification and synthesis of linear sequential machines, *Bell. Syst. Tech. J.,* v. 47, 1968, pp. 343—384.
[28] Pugsley, J. H., Sequential functions and linear sequential machines, *IEEE Trans.,* v. EC-14, 1965, pp. 376—382.
[29] Rabin, M. O. & D. Scott, Finite automata and their decision problems, *IBM J. Res. Develop.,* v. 3, 1959, pp. 114—125.
[30] Reusch, B., Linear and partial-linear realization of automata, *IFIP Congress 68,* Edinburgh, 1968, pp. A 151—A 155.
[31] Srinivasan, C. V., State diagram of linear sequential machines, *J. Franklin Inst.,* v. 273, 1962, pp. 383—418. Also in [25].
[32] —, *The state diagram of linear sequential machines,* Doctor of Engineering Science Dissertation, Columbia University, 1963.
[33] Wang, K. C., Transition graphs of affine transformations on vector spaces over finite fields, *J. Franklin Inst.,* v. 283, 1967, pp. 55—72.
[34] Yau, S. S. & K. C. Wang, Linearity of sequential machines, *IEEE Trans.,* v. EC-15, 1966, pp. 337—354, corrections on p. 925.

(Received March 12, 1969)