$$a = .x_1x_2x_3 \cdots = .y_1y_2 \cdots y_kz_1z_2 \cdots z_lz_1z_2 \cdots z_l \cdots,$$

namely, the base-σ expansion of a has period l. A tape $\xi_1\xi_2 \cdots \xi_n$ is in U_a if and only if there exists a non-negative integer $m \leq n$ such that

$$x_1x_2 \cdots x_{m-1} = \xi_1\xi_2 \cdots \xi_{m-1},$$

and

$$x_m > \xi_m,$$

or

$$x_1x_2 \cdots x_n = \xi_1\xi_2 \cdots \xi_n.$$

Utilizing this fact, an automaton with $k+l+2$ states can be constructed to define U_a as shown in the following diagram, (Fig. 1). S_1 is the initial state, and R is the rejection state. All of the states except R are accepting states. The label $<y_1$ stands for all letters of the alphabet which are smaller than y_1.

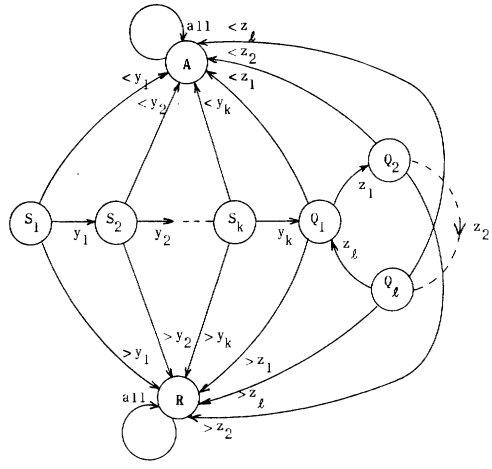

Fig. 1.

Now, assume that a is irrational. We shall prove that U_a is not regular by Nerode's Theorem.[1] The following statement shall be used. An event U is regular if and only if the equivalence relation $\sim$ (defined by the condition that for tapes X and Y, $X \sim Y$ if and only if for all tapes Z, whenever XZ is in U, then YZ is in U) is of finite index.

Consider the set S of tapes $X_i = x_1x_2 \cdots x_i$. Let X_i and X_j be two distinct tapes of S. Thus, $i \neq j$. It follows that

$$x_1x_2 \cdots x_ix_{i+1}x_{i+2} \cdots \neq x_1x_2 \cdots x_jx_{i+1}x_{i+2} \cdots$$

or the infinite fraction representing a is periodic. Let h be the least integer such that

$$x_1x_2 \cdots x_jx_{j+1} \cdots x_{j+h} = x_1x_2 \cdots x_jx_{i+1}x_{i+2} \cdots x_{i+h}$$

and

$$x_{j+h+1} \neq x_{i+h-1}.$$

and let $Z = x_{i+1}x_{i+2} \cdots x_{i+h+1}$. If $x_{j+h+1} < x_{i+h+1}$, then $X_jZ \notin U_a$ and $X_iZ \in U_a$ whence $X_i \not\sim X_j$. If $x_{j+h+1} > x_{i+h+1}$, then for every positive integer g, $X_jZ(\sigma-1)^g \in U_a$; but for some g, $X_iZ(\sigma-1)^g \notin U_a$, and again $X_i \not\sim X_j$. This implies that no two elements of S are equivalent and, therefore, that the index of $\sim$ is infinite. By Nerode's Theorem U_a is not regular.[2] Q.E.D.

Acknowledgment

The author wishes to thank Dr. M. Cohn of the Sperry Rand Research Center who criticized the manuscript.

[1] M. O. Rabin and D. Scott, "Finite Automata and their Decision Problems," *IBM J. Res. and Dev.*, vol. 3, pp. 114–125; April, 1959.

[2] One can also use Nerode's Theorem to prove the first part, but the construction of the automaton is easier to describe.

Regular Expressions from Sequential Circuits

J. A. BRZOZOWSKI, MEMBER, IEEE

Summary

In this paper the relation between a sequential circuit and its regular expression is investigated. The circuits are without special starting units. One method of analysis of a circuit leads to a set of equations whose solutions are regular expressions related to the state diagram of the circuit. In another approach, a set of regular equations, identical in form to the next state equations, is obtained directly from the circuit. By reversing the regular equations and using derivatives, the regular equations are transformed to a form related to the reverse state diagram. The discussion clarifies the relationship among circuits, regular expressions and state diagrams. Moreover, further insight is obtained into the solution of equations with regular expressions as unknowns.

Introduction

A sequential circuit [2]–[4] is constructed from unit delay elements and logical gates. It has state variables $y_1, y_2, \cdots, y_m$, where each y_j is an output of a unit delay element. For simplicity we consider circuits with a single input x and a single output z; the results are easily generalized later. The next state and output functions are

$$y_j' = f_j(y_1, y_2, \cdots, y_m, x), \quad j = 1, 2, \cdots, m, \tag{1}$$

$$z = g(y_1, y_2, \cdots, y_m), \tag{2}$$

where f_j and g are Boolean functions. We are using here the Moore [1] model where the output is a function of the state variables only; a Mealy [2] model can be treated in a similar way. For example, for the Moore circuit of Fig. 1(a) we have

$$y' = x \,\&\, \bar{y} \tag{3}$$

$$z = \bar{y}. \tag{4}$$

Logical connectives and gate symbols are defined in Fig. 1(b).

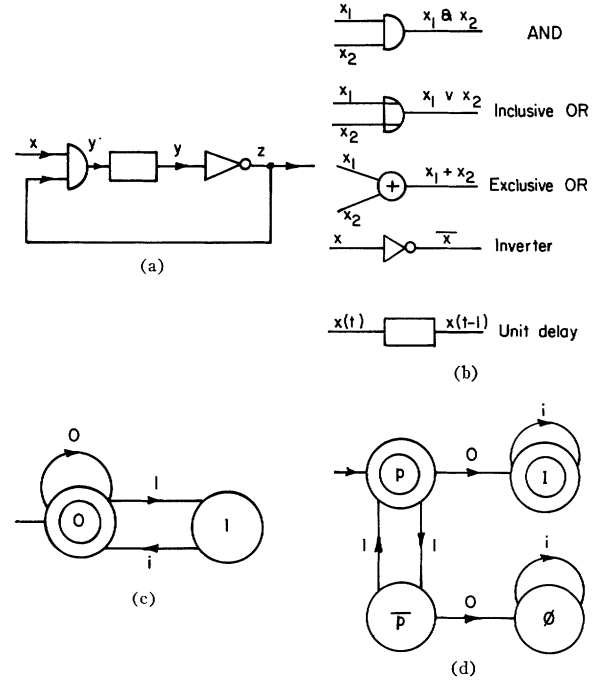

Fig. 1—(a) Circuit C_1. (b) Gate symbols. (c) State diagram. (d) Reverse state diagram.

Manuscript received March 30, 1964; revised July 13, 1964. The research reported was supported by the National Research Council of Canada under grant No. A-1617. Previously available from the Electrical Engineering Department, University of Ottawa, Canada, Tech. Rept. No. 64-3; January, 1964.

The author is with the Dept. of Electrical Engineering, University of Ottawa, Ontario, Canada.

The circuits considered operate synchronously, but there are no special starting signals [3], [4]. At $t=1$, the outputs of the delay elements are set to some arbitrary initial values; these values determine the starting state $q(1)$. At times 1, 2, $\cdots$, t an input sequence $s=x(1), x(2), \cdots, x(t)$ of length $L(s)=t$ is applied. This results in a state sequence $q(2), q(3), \cdots, q(t+1)$, and an output sequence $z(2), z(3), \cdots, z(t+1)$ [$z(1)$ depends on the initial state but not on s].

Definition 1: A sequential circuit C is said to *accept* [9] an input sequence s of length $t>0$ from initial state $q(1)$ if and only if, when C is started in $q(1)$ and s is applied, the output at time $t+1$ is $z=1$. If $z=1$ for the initial state $q(1)$ at time 1, C is said to accept the set λ consisting of the sequence of zero length. (Further discussion of λ may be found in [3], [4].)

The set of sequences accepted by a sequential circuit may be described by a regular expression, [3]–[12] and our problem here is to find such a regular expression for a given circuit. A basic knowledge of regular expressions is assumed; the minimum required background material is summarized in the Appendix, which also defines the symbols used in this paper.

Forward Approach

One can look at the problem of finding a regular expression for a circuit in two ways:

1) Given a circuit C in initial state q at time 1, what is the set of sequences which, when applied to C in state q, will leave C in a state with $z=1$? In this case we must examine the sequences in the forward direction, as time increases.
2) Given a circuit C with $z=1$ at time $t+1$, what sequences could have been applied to C started in the initial state at some earlier time in order to produce this condition? Here the sequences are examined in the reverse direction.

In this section we shall illustrate the forward approach by the example of Fig. 1(a). Let R_q be the set of all sequences accepted by the circuit from initial state q, *i.e.*, the sequences resulting in $z=1$. Suppose $y=0$ at $t=1$; then the required regular expression is R_0. Now, if an input $x=0$ is applied at $t=1$, the state at $t=2$ is still $y=0$; hence the expression applicable at $t=2$, after a 0 has been received, is still R_0. However, if $x=1$ is applied at $t=1$, the state at $t=2$ will be $y=1$; hence, after a 1 is received, the applicable expression is R_1. Since the output is 1 for the state $y=0$, R_0 must also contain λ. Thus

$$R_0 = 0R_0 \vee 1R_1 \vee \lambda. \tag{5}$$

Similarly we obtain for R_1:

$$R_1 = (0 \vee 1)R_0 = iR_0. \tag{6}$$

Equations of this type have been described in detail in the literature [3], [4] and can be solved using the fact that $X=AX \vee B$, where $A \not\ni \lambda$ and A and B are regular expressions, has the solution $X=A^*B$. Solving for R_0 we have

$$\begin{aligned} R_0 &= 0R_0 \vee 1iR_0 \vee \lambda \\ &= (0 \vee 1i)R_0 \vee \lambda \\ &= (0 \vee 1i)^*. \end{aligned} \tag{7}$$

Hence

$$R_1 = i(0 \vee 1i)^*. \tag{8}$$

Thus, if $y=0$ is the starting state the sequences accepted are denoted by (7), and if $y=1$ is the starting state the sequences accepted are denoted by (8).

It is clear that, in the analysis, the states of the circuit play the key role. Now several techniques are available for finding regular expressions from state diagrams and we shall not dwell on this method. We merely wanted to point out that the forward approach applied to a circuit involves the states of the circuit and the transitions among the states. Thus it is most efficient to construct the state diagram of the circuit first, and then proceed to find a regular expression using well-known methods.

The state diagram for the example is shown in Fig. 1(c). An incoming arrow indicates the starting state and a double circle denotes a state with an output $z=1$.

The objection to finding a regular expression from the state diagram is that the expression is not at all directly related to the circuit. Previously we have shown [10] that for linear sequential circuits there is another point of view in which the regular expressions are more directly related to the circuit. Such a point of view is next applied to the general case.

Regular Circuit Equations

As will be seen, when dealing directly with a sequential circuit, it is more convenient to consider whether a sequence s of length t produces an output at time t, rather than to ask whether the sequence is accepted (*i.e.*, produces an output at time $t+1$). For this reason we make the following definition.

Definition 2: An input sequence s of length $t>0$ *energizes* a point A in a sequential circuit C started in state q, if and only if, the signal at A is 1 at time t. The sequence of zero length energizes all points with 1 signals at time 1.

The reader can easily verify the following properties of sequential circuits:

1) The set of sequences energizing the input lead is the set of all sequences ending in a 1 which is denoted by the regular expression $I1$. (By convention, assume that λ does not energize the input lead.)
2) Let the set of sequences energizing the jth input $j=1, 2, \cdots, n$, of a logical gate performing the Boolean function f, be denoted by S_j. Then the set of sequences energizing the output of that gate is given by $f(S_1, S_2, \cdots, S_n)$. (The gate is assumed to have no delay.)
3) If R is the set of sequences energizing the input of a unit delay, then its output is energized by $Ri \vee \delta(y(1))$, where $\delta(0)=\phi$ (the empty set of sequences) and $\delta(1)=\lambda$.

The above properties are illustrated in Fig. 2.

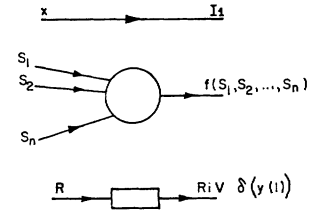

Fig. 2—Effects of circuit elements.

Consider a general sequential circuit C characterized by (1) and (2). Let the set of sequences energizing the input of the jth delay of C be called the jth *delay* (regular) *set*, and be denoted by R_j. Let the set of sequences energizing the output lead be called the *output* (regular) *set* and be denoted by Q. Then we can state:

Theorem 1: The delay and output sets of a sequential circuit C characterized by (1) and (2), must obey the equations:

$$R_j = f_j[R_1i \vee \delta(y_1(1)), R_2i \vee \delta(y_2(1)), \cdots, R_mi \vee \delta(y_m(1)), I1], \quad j = 1, 2, \cdots, m, \tag{9}$$

$$Q = g[R_1i \vee \delta(y_1(1)), R_2i \vee \delta(y_2(1)), \cdots, R_mi \vee \delta(y_m(1))]. \tag{10}$$

If in the initial state all delay outputs are 0, then a simpler form of (9) and (10) results

$$R_j = f_j[R_1i, R_2i, \cdots, R_mi, I1], \quad j = 1, 2, \cdots, m \tag{9a}$$

$$Q = g[R_1i, R_2i, \cdots, R_mi]. \tag{10a}$$

The theorem follows directly from Properties 1)–3). Note the one-to-one correspondence of (9a) and (10a) to (1) and (2), respectively, when R_j is identified with y_j', $R_j i$ with y_j, $I1$ with x, and Q with z. For simplicity we shall deal with the case of zero initial conditions. Eqs. (9a) and (10a) will be called the *regular circuit equations* and our problem is to solve these equations. However, in order to conform with the previous literature, we must find P, the set of sequences *accepted* by the circuit. This can be done as a result of the following corollary.

Corollary: If the delay and output sets of a sequential circuit are given by (9a) and (10a), then the set P of sequences accepted by the circuit is

$$P = g(R_1, R_2, \cdots, R_m). \tag{11}$$

This is evident from the definitions of accepted and energizing sequences.

As an example consider the circuit of Fig. 1(a) started with $y=0$ at $t=1$. Then R, the set of sequences energizing the delay input, must satisfy the following equation:

$$R = (I1)\ \&\ \overline{Ri}, \tag{12}$$

and P, the set accepted by the circuit is given by

$$P = \overline{R}. \tag{13}$$

Reverse Approach

We have shown that the analysis of a circuit gives rise to a set of equations with regular expressions as unknowns. At present it is known [3], [4] how to solve equations of the type $X=AX\vee B$ or $X=XA\vee B$ and equations arising from the analysis of linear circuits [10]. It will be shown that the method of derivatives [12]–[15], also called quotients [15] (see Appendix), allows us to obtain a solution for any set of equations from a general circuit.

In order to apply derivatives it is necessary to reverse [11] (see Appendix) the regular equations for the circuit. Let $r_j=R_j^-$ for notational convenience. Then (9a) becomes

$$r_j = f_j(ir_1, ir_2, \cdots, ir_m, 1I), \qquad j = 1, 2, \cdots, m. \tag{14}$$

Note what happens when we take derivatives with respect to 0 and 1

$$D_0 r_j = f_j(r_1, r_2, \cdots, r_m, \phi) \tag{15}$$

$$D_1 r_j = f_j(r_1, r_2, \cdots, r_m, I). \tag{16}$$

Thus the derivatives of the reverse expressions r_j with respect to sequences of length one are simply Boolean functions of the r_j. Consequently all derivatives with respect to longer sequences are also Boolean functions of the r_j. Now it is known [12]–[15] that r_j has a finite number of distinct derivatives; since there is a finite number of Boolean functions of the r_j, the process of constructing the derivatives will terminate without difficulty.

Consider the simple circuit of Fig. 1(a) again. Here

$$r = (1I)\ \&\ \overline{ir}, \tag{17}$$

$$D_0 r = \phi\ \&\ \bar{r} = \phi \tag{18}$$

$$D_1 r = I\ \&\ \bar{r} = \bar{r}, \tag{19}$$

$$D_{10} r = D_0 r = \bar{\phi} = I, \tag{20}$$

$$D_{11} r = \overline{D_1 r} = \bar{\bar{r}} = r. \tag{21}$$

The process terminates here because no new derivatives are found for sequences of length 3. At this point we can write r in terms of its derivatives [12] (see Appendix):

$$r = 0D_0 r \vee 1D_1 r \vee \delta(r) = 0\phi \vee 1D_1 r = 1D_1 r, \tag{22}$$

$$D_1 r = 0D_{10} r \vee 1D_{11} r \vee \delta(D_1 r) = 0I \vee 1r \vee \lambda = \bar{r}. \tag{23}$$

Now, since we are really interested in $\bar{r}=p=P^-$, we can solve for p:

$$\begin{aligned} p &= \lambda \vee 0I \vee 1r = \lambda \vee 0I \vee 11p \\ &= (11)^*(\lambda \vee 0I). \end{aligned} \tag{24}$$

Finally, reversing p we obtain another valid expression for the circuit of Fig. 1(a):

$$P = (I0 \vee \lambda)(11)^*. \tag{25}$$

One can verify that P is equivalent to R_0 of (7) found by the forward approach.

Returning to $p=\bar{r}$, we see that the derivatives of p

$$D_\lambda p = p,\ D_0 p = I,\ D_1 p = \bar{p},\ D_{10} p = \phi,$$

define a state diagram [Fig. 1(d)] which is the reverse of that of Fig. 1(c).

In summary the reader can verify that

1) For every one-input, one-output sequential circuit, one can write a set of regular equations directly related to the circuit structure (Theorem 1).
2) The equations can be reversed as in (14).
3) The derivatives of r_j with respect to sequences of length 1 are Boolean functions of the r_j as in (15) and (16).
4) The reverse of the set of sequences accepted by the circuit is given by
$$p = g(r_1, r_2, \cdots, r_m).$$
5) Since for $a \in A_k$, $D_a p = g(D_a r_1, D_a r_2, \cdots, D_a r_m)$, the derivatives of p with respect to sequences of length 1 are Boolean functions of the r_j.
6) Conequently, all derivatives of p are Boolean functions of the r_j.
7) To find the derivatives of p it is necessary to know the derivatives of the r_j only with respect to sequences of length 1.
8) The distinct derivatives of p can be found in a finite number of steps. Hence the characteristic equations [11], [12] for p and its derivatives can be found.
9) The characteristic equations can be solved for p using the solution of the general form $X=AX\vee B$.
10) The regular expression p obtained in this way is related to the reverse state diagram of the circuit.
11) The accepted regular expression is found by reversing p.

Generalizations

The example that we used was chosen for the sake of simplicity but was rather restricted. If there is more than one input the extension is straightforward. Suppose a next state equation has the form

$$y_1' = (x_1\ \&\ y_1) \vee (\bar{x}_1\ \&\ x_2\ \&\ \bar{y}_1\ \&\ y_2). \tag{26}$$

Now each function of the inputs represents, in general, several input combinations; for example $x_1=(x_1\&\overline{x_2})\vee(x_1\&x_2)$. Represent each input combination by its decimal equivalent: $(x_1, x_2)=(0, 0)$ by 0, $(x_1, x_2)=(1, 0)$ by 2, etc. Then x_1 is represented by $2\vee 3$. Thus R_1 corresponding to (26) is

$$R_1 = (I(2 \vee 3))\ \&\ (R_1 i) \vee (I1)\ \&\ \overline{R_1 i}\ \&\ (R_2 i).$$

If there are r outputs z_k, then we must find r regular expressions P_k, or we can treat P_k as components of a regular vector [12]P. At any rate the extension is again straightforward.

It is felt that an example with more than one delay should be given to illustrate a more general case. For the circuit of Fig. 3 we have

$$\begin{aligned} R_1 &= \overline{I1}\ \&\ (R_1 i + R_2 i) \\ R_2 &= \overline{R_1 i}\ \&\ (I1 \vee R_2 i) \\ P &= \overline{R_1}\ \&\ R_2. \end{aligned}$$

Reversing we obtain

$$\begin{aligned} r_1 &= \overline{1I}\ \&\ (ir_1 + ir_2) \\ r_2 &= \overline{ir_1}\ \&\ (1I \vee ir_2) \\ p &= \overline{r_1}\ \&\ r_2. \end{aligned}$$

Now in order to construct derivatives of p we require the derivatives of r_1 and r_2 with respect to sequences of length one. Thus:

$$D_0 r_1 = r_1 + r_2 \qquad D_0 r_2 = \overline{r_1}\ \&\ r_2$$

$$D_1 r_1 = \phi \qquad D_1 r_2 = \overline{r_1}.$$

This is sufficient to construct derivatives of p

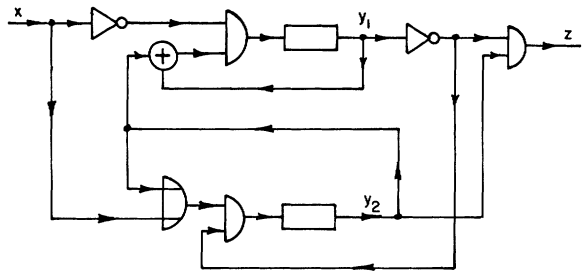

Fig. 3—Circuit C_2.

$$p = \overline{r_1} \,\&\, r_2$$
$$D_0 p = \overline{r_1 + r_2} \,\&\, (\overline{r_1} \,\&\, r_2) = \phi$$
$$D_1 p = I \,\&\, \bar{r}_1 = \bar{r}_1$$
$$D_{10} p = \overline{r_1 + r_2} = I + r_1 + r_2$$
$$D_{11} p = I$$
$$D_{100} p = I + r_1 + r_2 + (\bar{r}_1 \,\&\, r_2) = \bar{r}_1 \vee r_2$$
$$D_{101} p = I + \phi + \bar{r}_1 = r_1, \text{ etc.}$$

As a final point, consider the effect of a starting state different from 0, on the reverse approach. This can be illustrated by the example of Fig. 1(a). If $y=1$ at $t=1$, the proper equation for R is

$$R = (I1) \,\&\, \overline{Ri \vee \lambda}.$$

The remainder of the procedure is the same.

Conclusions

We have investigated the possibility of obtaining a regular expression directly from a sequential circuit. We have shown that both the direct and the reverse approach make use of a state diagram. It appears that the only more direct method would be to solve the regular circuit equations without using derivatives (which in effect constitute a state diagram). Before this can be done we must learn more about the regular algebra. However, it appears unlikely that a much simpler method can be found.

Appendix

Regular expression [3]–[12] is defined recursively:

1) λ, ϕ, and a (in A_k) are regular expressions.
2) If P and Q are regular expresssions then so are PQ, P^*, $f(P, Q)$, where $f(P, Q)$ denotes any Boolean expression in P and Q. The symbols for common Boolean functions are shown in Fig. 1(b).
3) Only expressions obtained from Rules 1) and 2) by a finite number of applications are regular.

We use the same symbol for a sequence as for the set of sequences consisting of only that sequence. Regular expressions denote sets of sequences. Above A_k is the input alphabet, $A_k = \{0, 1, \cdots, (k-1)\}$, λ is the set consisting of the sequence of zero length and ϕ is the empty set of sequences. Also PQ denotes concatenation, and

$$P^* = \bigcup_{n=0}^{\infty} P^n,$$

where $P^2 = PP$, etc., and $P^0 = \lambda$. Furthermore we let the set of all sequences be I, and the set of all sequences of length 1 be $i = 0 \vee 1 \vee \cdots \vee (k-1)$.

The *derivative* [12]–[15] $D_s R$ of a regular set R with respect to a sequence s is a regular set defined by

$$D_s R = \{t \mid st \in R\},$$

and is found recursively. If a, $a_j \in A_k$ and P, Q are regular expressions then the regular expressions for derivatives are found as follows:

1a) $D_a a = \lambda$

$D_a b = \phi$ for $b = \lambda$ or $b = \phi$ or $b \in A_k$ and $b \neq a$

1b) $D_a(P^*) = (D_a P)P^*$

$D_a(PQ) = (D_a P)Q \vee \delta(P) D_a Q$

$D_a(f(P, Q)) = f(D_a P, D_a Q)$

where $\delta(P) = \lambda$ if $P \supset \lambda$, $\delta(P) = \phi$ otherwise.

2) $D_{a_1 a_2} P = D_{a_2}(D_{a_1} P)$

$D_{a_1 a_2 \cdots a_r} P = D_{a_r}(D_{a_1 a_2 \cdots a_{r-1}} P)$

$D_\lambda P = P$.

The *reverse* [11] R of a regular expression R is defined recursively:

$$x^- = x \text{ for } x \in A_k \text{ or } x = \lambda \text{ or } x = \phi$$
$$(PQ)^- = Q^- P^-$$
$$(P^*)^- = (P^-)^*$$
$$(f(P, Q))^- = f(P^-, Q^-).$$

References

[1] E. F. Moore, "Gedanken-experiments on sequential machines," in "Automata Studies, Annals of Math. Studies," C. E. Shannon and J. McCarthy, Eds., Princeton University Press, Princeton, N. J., No. 34, pp. 129–153; 1956.
[2] G. H. Mealy, "A method for synthesizing sequential circuits," *Bell. Sys. Tech. J.*, vol. 34, pp. 1045–1079; September, 1955.
[3] D. N. Arden, "Delayed Logic and Finite State Machines," *Proc. 2nd Ann. Symp. on Switching Circuit Theory and Logical Design*, pp. 133–151; September, 1961. Also in "Theory of Computing Machine Design," University of Michigan Engineering Summer Conferences, Ann Arbor, Mich., pp. 1–35; 1960.
[4] J. A. Brzozowski, "A survey of regular expressions and their applications," IRE Trans. on Electronic Computers, vol. EC-11, pp. 324–335; June, 1962, and Dept. of Elec. Engrg., Digital Systems Lab., Princeton University, Princeton, N. J., Tech. Rept. No. 4; April, 1961.
[5] S. C. Kleene, "Representation of events in nerve nets and finite automata," in "Automata Studies, Annals of Math. Studies," C. E. Shannon and J. McCarthy Eds., Princeton University Press, Princeton, N. J., No. 34, pp. 3–41, 1956.
[6] R. McNaughton and H. Yamada, "Regular expressions and state graphs for automata," IRE Trans. on Electronic Computers, vol. EC-9, pp. 39–47; March, 1960.
[7] J. A. Brzozowski and E. J. McCluskey, Jr., "Signal flow graph techniques for sequential circuit state diagrams," IEEE Trans. on Electronic Computers, vol. EC-12, pp. 67–76; April, 1963.
[8] I. M. Copi, C. C. Elgot, and J. B. Wright, "Realization of events by logical nets," *J. ACM*, vol. 5, pp. 181–196; April, 1958.
[9] M. O. Rabin and D. Scott, "Finite automata and their decision problems," *IBM J. Res. and Dev.*, vol. 3, pp. 114–125; April, 1959.
[10] J. A. Brzozowski, "Regular expressions for linear sequential circuits," *Proc. First Annual Allerton Conf. on Circuit and System Theory*, pp. 406–426; November, 1963.
[11] J. A. Brzozowski, "Canonical regular expressions and minimal state graphs for definite events," *Proc. Symp. on the Mathematical Theory of Automata*, Polytechnic Press of the Microwave Research Institute Symposia Ser., Polytechnic Inst. of Brooklyn, N. Y., vol. 12 pp. 529–561; 1963.
[12] J. A. Brzozowski, "Derivatives of regular expressions," *J. ACM*, vol. 11, pp. 481–494; October, 1964. See also "Properties of Regular Expressions and State Diagrams," Dept. of Elec. Engrg., Digital Systems Lab., Princeton University, Princeton, N. J.; Tech. Rept. No. 15; March, 1962.
[13] C. C. Elgot and J. D. Rutledge, "Operations on finite automata," *Proc. 2nd Ann. Symp. on Switching Theory and Logical Design*, pp. 129–132; October, 1961.
[14] R. E. Stearns and J. Hartmanis, "Regularity preserving modifications of regular expressions," *Information and Control*, vol. 6, pp. 55–69; September, 1963.
[15] S. Ginsburg and G. F. Rose, "Operations which preserve definability in languages," *J. ACM*, vol. 10, pp. 175–195; 1963.

A Sampled-Data Analog of Neuron Properties

E. E. NELSON, MEMBER, IEEE, AND H. F. WOLF

A mathematical neuron model is discussed that can be used in investigating neural net activity. This model incorporates many of the data processing capabilities of biological neurons in a simple, iterative set of equations. The neuron properties simulated are linear, weighted summation of inputs, standard signal shape, threshold firing, refractory recovery, signal delay, and temporal summation. The necessary equations are derived by dissecting these characteristics into small and separate entities and then generating a sampled-data model that conveniently combines their effects. A flow chart that diagrams the necessary program to implement this neuron analog on a digital computer is developed.

In recent years, the tempo of activity in the area of neural modeling has shown a marked increase. This heightened interest has been

Manuscript received February 14, 1964.
The authors are with Douglas Missile and Space Systems Div., Newport Beach, Calif.