1964

Short Notes

471

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 $y_1, y_2, \ldots, y_m$, where each $y_i$ 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

\[
y'_j = f_j(y_1, y_2, \ldots, y_m, x), \quad j = 1, 2, \ldots, m, \tag{1}
\]

\[
z = g(y_1, y_2, \ldots, y_m), \tag{2}
\]

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).

ACKNOWLEDGMENT

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

\[\]
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 \( 1, 2, \ldots, t \) an input sequence \( s = x(1), x(2), \ldots, x(t) \) of length \( L(s) = t \) is applied. This results in a state sequence \( q(2), q(3), \ldots, q(t+1) \), and an output sequence \( z(2), z(3), \ldots, z(t+1) \). \( s(1) \) 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 \( z(1) = 1 \). If \( s = 1 \) for the initial state \( q(1) \) at time 1, \( C \) is said to accept the set \( \lambda \) consisting of the sequence of zero length. (Further discussion of \( \lambda \) 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 \( R_0 \) 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 \( y = 0 \); hence the expression applicable when \( t = 2 \), 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 \( \lambda \). 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 \( A \), \( \lambda \) and \( B \) are regular expressions, has the solution \( X = A^*B \). Solving for \( R_0 \) we have

\[
R_0 = 0R_0 \vee 1iR_0 \vee \lambda \\
= (0 \vee 1)iR_0 \vee \lambda \\
= (0 \vee 1)i^*. \tag{7}\]

Hence

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

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 \( J_1 \). (By convention, assume that \( \lambda \) does not energize the input lead.)

2. Let the set of sequences energizing the \( j \)th input \( j = 1, 2, \ldots, n \), 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 \( f(S_1, S_2, \ldots, S_n) \). (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 \( R \vee \delta(y(1)), \) where \( \delta(0) = \phi \) (the empty set of sequences) and \( \delta(1) = \lambda \).

The above properties are illustrated in Fig. 2.

Consider a general sequential circuit \( C \) characterized by (1) and (2). Let the set of sequences energizing the input of the \( j \)th delay of \( C \) be called the \( j \)th 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:

\[
R_j = f_j[R_i \vee \delta(y_1(1)), R_i \vee \delta(y_1(1)), \ldots, R_m \vee \delta(y_n(1)), I_1], \quad f = j, 1, 2, \ldots, m, \tag{9}\]

\[
Q = g[R_i \vee \delta(y_1(1)), R_i \vee \delta(y_1(1)), \ldots, R_m \vee \delta(y_n(1))]. \tag{10}\]

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

\[
R_j = f_j[R_i, R_i, \ldots, R_i, I_1], \quad j = 1, 2, \ldots, m \tag{9a}\]

\[
Q = g[R_i, R_i, \ldots, R_m]. \tag{10a}\]
The theorem follows directly from Properties 1-3). Note the one-to-one correspondence of (9a) and (10a) to (1) and (2), respectively, when \( R_i \) is identified with \( y_j' \), \( R_j \) with \( y_i \), \( f_i \) with \( x_i \), 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

\[
P = g(R_1, R_2, \ldots, R_m). \tag{11}
\]

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 \( i = 1 \). Then \( R \), the set of sequences energizing the delay input, must satisfy the following equation:

\[
R = (11) \& \overline{R_i}, \tag{12}
\]

and \( P \), the set accepted by the circuit is given by

\[
P = R. \tag{13}
\]

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 \( X = AX + \sqrt{B} \) or \( X = XA + \sqrt{B} \) 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

\[
r_j = f_j(r_i, r_n, \ldots, r_m, I), \quad j = 1, 2, \ldots, m. \tag{14}
\]

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

\[
D_0 r_j = f_j(r_1, r_2, \ldots, r_m, 0) \tag{15},
\]

\[
D_1 r_j = f_j(r_1, r_2, \ldots, r_m, 1). \tag{16}
\]

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

\[
r = (11) \& \overline{I}, \tag{17}
\]

\[
D_0 \overline{r} = \overline{r} \& \overline{r} = \overline{r}, \tag{18}
\]

\[
D_1 \overline{r} = I \& \overline{r} = \overline{r}, \tag{19}
\]

\[
D_0 \overline{r} = D_0 \overline{r} = \overline{r}, \tag{20}
\]

\[
D_1 \overline{r} = D_1 \overline{r} = \overline{r} = r. \tag{21}
\]

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):

\[
r = D_0 \overline{r} \lor 1 D_1 \overline{r} \lor \delta(I) = D_0 \overline{r} \lor I D_1 \overline{r}, \tag{22}
\]

\[
D_0 \overline{r} = D_0 \overline{r} \lor 1 D_1 \overline{r} \lor \delta(D_0 \overline{r}) = 0 \lor 1 r \lor \overline{r} = \overline{r}. \tag{23}
\]

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

\[
p = \lambda \lor 0 \lor 1 = \lambda \lor 0 \lor 11_\lambda = (11)^* \lambda \lor 0. \tag{24}
\]

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

\[
P = (10 \lor \lambda)^* \tag{25}.\]

One can verify that \( P \) is equivalent to \( R_9 \) of (7) found by the forward approach.

Returning to \( p = \overline{r} \), we see that the derivatives of \( p \)

\[
D_0 \overline{p} = r, \quad D_1 \overline{p} = I, \quad D_2 \overline{p} = \overline{r}, \quad D_3 \overline{p} = \phi,
\]

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 4).
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

\[
p = g(r_1, r_2, \ldots, r_m).
\]

5) Since for \( a \in A_i \), \( D_0 p = g(D_0 r_1, D_0 r_2, \ldots, D_0 r_m) \), the derivatives of \( p \) with respect to sequences of length 1 are Boolean functions of the \( r_j \).
6) Consequently, 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 \( X = AX + \sqrt{B} \).
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

\[
y_j' = (x_1 \& y_1) \lor (x_2 \& y_2) \lor (x_3 \& y_3). \tag{26}
\]

Now each function of the inputs represents, in general, several input combinations; for example \( x_1 = (x_2 \& x_3) \lor (x_1 \& x_3) \). Represent each input combination by its decimal equivalent: \( x_1 = x_2 + x_3 = (0, 0) \) by \( 0 \), \((x_1, x_3) = (1, 0) \) by \( 2 \), etc. Then \( x_1 \) is represented by \( 2 \lor 3 \). Thus \( R_1 \) corresponding to (26) is

\[
R_1 = (I(2 \lor 3)) \& (R_3) \lor (I1) \& \overline{R_3} \& (R_4).
\]

If there are \( r \) outputs \( z_a \), then we must find \( r \) regular expressions \( P_a \) or we can treat \( P_a \) as components of a regular vector \([12]\). 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

\[
R_1 = I1 \& (R_3 + I3)
\]

\[
R_3 = I3 \& (I1 \lor R_4)
\]

\[
P = \overline{R_3} \& R_4.
\]

Reversing we obtain

\[
r_1 = \overline{I} \& (r_1 + I3)
\]

\[
r_3 = \overline{I} \& (I1 \lor I3)
\]

\[
p = r_1 \& r_3.
\]

Now in order to construct derivatives of \( r_1 \) and \( r_2 \) with respect to sequences of length one. Thus:

\[
D_0 r_1 = r_1 + r_2
\]

\[
D_1 r_1 = \phi
\]

\[
D_2 r_1 = \overline{r_1}.
\]

This is sufficient to construct derivatives of \( p \).
The set of sequences is a regular set if

\[ \begin{align*}
D_0P &= (D_0P)P^* \\
D_0Q &= (D_0P)Q/V(\delta(P)Q)
\end{align*} \]

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

The reverse \[11\] \( R \) of a regular expression \( R \) is defined recursively:

\[\begin{align*}
x^* &= x \quad \text{for } x \in A_k \text{ or } x = \lambda \text{ or } x = \phi \\
(PQ)^* &= (P^*)Q^* \\
(P^*)^* &= (P^*)^*
\end{align*}\]

The symbols for common Boolean functions are shown in Fig. 1(b).

**Appendix**

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

1. \( \lambda, \phi, \) and \( a \) (in \( A_k \)) are regular expressions.
2. If \( P \) and \( Q \) are regular expressions 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, \( A_k = [0, 1, \ldots, k-1] \), \( \lambda \) is the set consisting of the sequence of zero length and \( \phi \) is the empty set of sequences. Also \( PQ \) denotes concatenation, and

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

where \( P^* = PP \), 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 \( I = 0/1 \ldots \varepsilon (k-1) \).

The derivative \[12\]-[15] \( D_kR \) of a regular set \( R \) with respect to a sequence \( s \) is a regular set defined by

\[ D_kR = \{ i | s \in R \}, \]

and is found recursively. If \( a_0 \notin A_k \) and \( P, Q \) are regular expressions then the regular expressions for derivatives are found as follows:

1)\[ D_0a = \lambda \]

2)\[ D_0b = \phi \text{ for } b = \lambda \text{ or } b = \phi \text{ or } b \in A_k \text{ and } b \neq a \]

**REFERENCES**


