1968

Sequential Boolean equations

Shimon Even
Carnegie Mellon University

Albert R. Meyer

SEQUENTIAL BOOLEAN EQUATIONS

By

Shimon Even[*]

The Aiken Computation Laboratory
Harvard University

Albert R. Meyer

Department of Computer Science
Carnegie-Mellon University

[*] On leave of absence from Technion-Israel Institute of Technology.

March, 1968

This work was supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (SD-146) and is monitored by the Air Force Office of Scientific Research. Distribution of this document is unlimited.

ABSTRACT

The problem of solving sequential Boolean equations is shown to be equivalent to the problem of finding whether there exists a path on a labeled graph for every sequence of labels. Algorithms are given for testing whether a solution exists, and if a solution with a finite delay exists. In case of existence of solutions the algorithms provide them.

TABLE OF CONTENTS

I. Introduction

Assume $x_1, x_2, \ldots, x_m$ are independent binary variables and $y_1, y_2, \ldots, y_n$ are binary dependent variables (unknowns) satisfying the equation

$$F\ (x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n, dy_1, dy_2, \ldots, dy_n) = 0, \qquad (1)$$

where F is a Boolean expression of the described variables and the connectives "+" (Boolean addition), '.' (Boolean multiplication), '-' (complementation) and dz is the value of z one time unit later. Namely, it is assumed that time is discrete and

$$dz(t) = z(t + 1). \qquad (2)$$

Let us denote the vector $(x_1, x_2, \ldots, x_m)$ by X and the vector $(y_1, y_2, \ldots, y_n)$ by Y. Similarly, let

$$dY = (dy_1, dy_2, \ldots, dy_n).$$

X may have any one of the 2^m values and Y may have any one of the 2^n values. The sequence of values of X at time t = 1,2, ..., L is denoted by X(1), X(2), ..., X(L) and there are 2^{mL} such possible sequences. The sequence of values of Y at time t = 1, 2, ..., L, L + 1 is denoted by Y(1), Y(2), ..., Y(L + 1) and there are $2^{m(L + 1)}$ such possible sequences. The original equation (1) can be written in the form

$$F\ (X, Y, dY) = 0 \qquad (3)$$

Definition 1.1: The equation (3) is said to have a solution if for every finite sequence of values for X, namely, X(1), X(2), ..., X(L) there exists a sequence of values for Y, namely, Y(1), Y(2), ... Y(L), Y(L + 1) such that (3) will hold for every t = 1, 2, ..., L.

H. Wang investigated a seemingly larger class of equations, namely, equations of the form

$$G(X, dX, Y, dY) = 0 \qquad (4)$$

However, every equation of this form can be transformed into the form of (1) by the following steps:

(1) Define new dependent variables $z_1, z_2, \ldots, z_m$

(2) Form the following set of simultaneous equations, where Z denotes $(z_1, \ldots, z_m)$:

$$\left.\begin{array}{c} G(X, dZ, Y, dY) = 0 \\ z_1 = x_1 \\ z_2 = x_2 \\ \vdots \\ z_m = x_m \end{array}\right\} \quad (5)$$

(3) Transform the simultaneous equations into one equation using the rules $A=0$ and $B=0 \longleftrightarrow A+B = 0$, and $C=D \longleftrightarrow C\bar{D} + \bar{C}D = 0$. The resulting form in this case is:

$$H(X, Y, Z, dY, dZ) = 0 \qquad (6)$$

where $x_1, \ldots, x_m$ are independent variables, and $y_1, \ldots, y_n, z_1, \ldots, z_m$ are dependent variables.

Equation (6) is in the form of (1), and it is clear that equation (6) obtained in this way has a solution if and only if equation (1) has a solution. For example, consider Wang's Example 1: (Here the names of the variables are changed to agree with the present notation.)

Example 1: In the following equation, y is the dependent variable:

$$\bar{x}\ \bar{y}\ d\bar{y} + \bar{x}\ y\ dy + x\ \bar{y}\ dx + x\ y\ d\bar{x} = 0.$$

We replace dx by dz in the above equation, and form the simultaneous equations:

$$\left.\begin{array}{l} \bar{x}\bar{y}d\bar{y} + \bar{x}ydy + x\bar{y}dz + xyd\bar{z} = 0 \\ x = z \end{array}\right\}.$$

Now, $x = z$ is equivalent to $x\bar{z} + \bar{x}z = 0$, and the two simultaneous equations are equivalent to:

$$\bar{x}\bar{y}d\bar{y} + \bar{x}ydy + x\bar{y}dz + xyd\bar{z} + x\bar{z} + \bar{x}z = 0.$$

This last equation is of the form:

$$F(x, y, z, dy, dz) = 0$$

where x is an independent variable and both y and z are dependent variables. It has a solution if and only if the given equation has one. We remark that Definition 1.1 extends in an obvious way to equations involving terms such as $d^k x_i$ (where $d^k x_i(t) = x_i(t+k)$). Wang observes that such equations can be reduced to form (4), so we need not consider them further.

Definition 1.2: The equation (3) is said to have a solution with a finite delay d if the knowledge of X(1), X(2), ..., X(d) is sufficient to determine a value for Y(1), and for every t, the knowledge of Y(t), X(t), X(t + 1),..., X(t +d) is sufficient to determine a value for Y(t + 1) so that the determined part of the sequence of Y is part of a solution for the given sequence of X in the sense of Definition 1.1.

An alternative statement is that there exists functions f_1 and f_2 such that for any $L \geq d$ and any sequence $X(1), X(2), \ldots, X(L)$ the sequence $Y(1), \ldots, Y(L-d+1)$ given by:

$$\begin{aligned} &Y(1) = f_1\ (X(1), X(2), \ldots., X(d)) \\ &Y(t+1) = f_2(Y(t), X(t), X(t+1), \ldots., X(t+d)) \qquad (7) \\ &\text{for } t = 1, 2, \ldots, L-d \end{aligned}$$

can be augmented by some values for $Y(L-d+2),\ldots,Y(L+1)$ to satisfy equation (3) at every $t = 1, 2, \ldots, L$.

Wang described an algorithm for deciding whether a given equation has a solution, in which he makes use of solution tables. In the next section we shall prove that the problem of solving such equations is equivalent to the problem of deciding whether in a given labeled graph, for every sequence of labels there exists a path in the graph. In section 3 a method of solution is presented, which is believed to be more efficient than Wang's method.

Wang also described an algorithm for deciding whether a given equation has a solution with a finite delay. However, the authors are unable to understand the procedure in full. Furthermore, Theorem 6 in Wang's paper, which summarizes his results on this subject, is false.* A counter-example is given in Section 4. In Section 2 it is shown that the problem of deciding whether an equation has a solution with a finite delay is equivalent to the problem of deciding whether in a given labeled graph, for every sequence of labels there exists a path in the graph which can be determined with a finite delay. In Section 4 the solution to the latter problem is presented.

* Wang has informed us in a private communication that Theorem 6 is false as stated because of a misprint. In any case, we believe that our decision procedure is simpler than Wang's original procedure.

II. Equations and Graphs

Let G be a directed graph with N vertices: $V = \{v_1, v_2, \ldots, v_N\}$. Each edge is labeled with one letter of an alphabet Σ of σ letters: $\ell_1, \ell_2, \ldots, \ell_\sigma$. We may assume that there is at most one edge from v_i to v_j with label ℓ, since for our purposes having several such edges is the same as having one. There are no other restrictions on G. (There may be any number of edges with a given label emanating from a given vertex; clearly, such numbers must be integers between 0 and N. The graph may or may not be connected.)

Let the elements of Σ be called letters, and a finite sequence of letters be called a tape. A path is a finite sequence of edges: $e_1, e_2, \ldots, e_k$ such that e_i enters the same vertex from which e_{i+1} emanates, for every $i = 1, 2, \ldots, k-1$. An edge may be used any number of times on one path, and naturally, a path can go through one vertex any number of times, and thus, may include loops (circuits). With each path there is an associated (unique) tape of the labels appearing on the path in their natural order.

Let U be any set of vertices, and τ be any tape. $\Gamma(U,\tau)$ is the set of all vertices reachable from U by paths labeled τ. Σ^* is the set of all tapes with letters of Σ.

Definition 2.1: The graph G is said to be solvable with respect to Σ if for every tape in Σ^* there exists a path on G whose sequence of labels is the given tape.

In symbols:

$$G \text{ is solvable with respect to } \Sigma \Leftrightarrow \forall \tau \epsilon \Sigma^* [\Gamma(V,\tau) \neq \emptyset].$$

We shall now show that the problem of solving an equation of the type (3) in the sense of Definition 1.1. can be reduced to a problem of deciding whether a given graph is solvable with respect to some alphabet. The method is most easily shown by means of an example.

Consider, again, Example 1; after it has been transformed to the form of equation (1).

$$\bar{x}\bar{y}d\bar{y} + \bar{x}ydy + x\bar{y}dz + xyd\bar{z} + x\bar{z} + \bar{x}z = 0$$

For illustrative purposes we expand the left hand side on y, z, dy, dz. The resulting expression in this case is:

$$\begin{aligned} & yzdydz(\bar{x}) + yzdyd\bar{z}(1) + yzd\bar{y}dz(\bar{x}) + yzd\bar{y}d\bar{z}(1) + \\ & + y\bar{z}dydz(1) + y\bar{z}dyd\bar{z}(1) + y\bar{z}dyd\bar{z}(x) + y\bar{z}dyd\bar{z}(x) + \\ & + \bar{y}zdydz(1) + \bar{y}zdyd\bar{z}(\bar{x}) + \bar{y}zd\bar{y}dz(1) + \bar{y}zd\bar{y}dz(\bar{x}) + \\ & + \bar{y}\bar{z}dydz(x) + \bar{y}\bar{z}dyd\bar{z}(x) + \bar{y}\bar{z}d\bar{y}dz(1) + \bar{y}\bar{z}d\bar{y}d\bar{z}(1) = 0. \end{aligned}$$

Consider the first term: $yzdydz(\bar{x})$. The meaning of it is that $yzdydz = 1$ only if $\bar{x} = 0$, or in other words, the state $yz = 1$ may be followed by the state $dydz = 1$ only if $x = 1$. There are 4 possible values for (y,z), namely (0,0), (0,1), (1,0) and (1,1). We construct a graph with 4 vertices corresponding to these 4 states. Vertex (a,b) is connected to vertex (c,d) with an edge labeled e if and only if the left hand side of the equation becomes zero on the substitution of y=a, z=b, dy=c, dz=d and x=e. Therefore, in our case, vertex (1,1) is connected to (1,1) with an edge labeled 1. Similarly, each of the 16 terms may define up to 2 edges in the graph. The resulting graph is shown in Figure 1, and is also represented by Table 1.

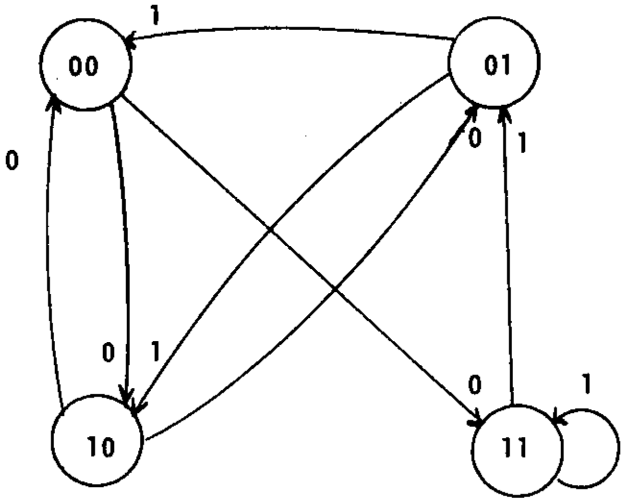

Figure 1.

The graph representing the equation of Example 1.

Label / Vertex	0	1
1 1	-	1 1, 0 1
1 0	0 1 , 0 0	-
0 1	-	1 0 , 0 0
0 0	1 1 , 1 0	-

Table 1.

A table representation of the graph of Equation 1.

It is interesting to note that Table 1 is similar to the characterizing table Wang has constrcted for this example.

In general, the graph representing an equation of the form (1) will have $N = 2^n$ and $\sigma = 2^m$. Assume we are given a sequence of values for X, namely, X(1), X(2), ..., X(L) and we want to find a sequence of values for Y, namely, Y(1), Y(2), ..., Y(L + 1) such that the equation will hold for t = 1, 2, ..., L. The meaning of it in the graph representation is that we are given a tape of length L and want to find a path of this length on the graph with the same tape of labels. If such a path can be found then the sequence of nodes the path goes through will specify a sequence of values for Y which satisfy the requirement, and vice versa. Thus, the problem of deciding whether an equation has a solution is equivalent to the problem of deciding whether the corresponding graph is solvable.

Given a graph G with N nodes and label-alphabet Σ with σ letters, one can construct a corresponding equation which has a solution if and only if the graph is solvable. Again, let us use an example, which will also be useful later, to demonstrate this point.

Example 2: Consider the graph shown in Figure 2.

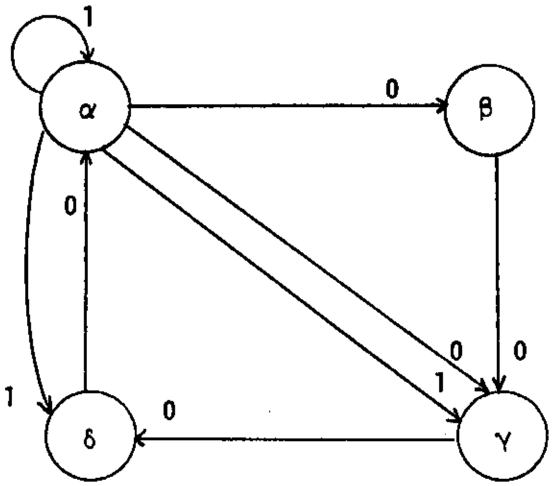

Figure 2.

The graph of Example 2.

Let us choose the following code for the nodes of the graph:

α : 0 0

β : 0 1

γ : 1 1

δ : 1 0.

This code may be chosen arbitrarily, as long as each state has a unique codeword. Clearly, the only requirement is that the length of the codewords, p, satisfy the condition

$$2^p \geq N.$$

In case $2^p > N$ not all codewords are assigned to nodes, and they may be thought of as assigned to nodes which are neither entered nor left by. edges. Such isolated nodes will never be a part of a chosen path, and the corresponding values for Y will never appear in a chosen sequence of values for Y.

In our example the choice of $\Sigma = \{0,1\}$ is obvious. Again, in general the letters of the alphabet will have to be coded into binary codewords of length q such that $2^q \geq \sigma$. In this case, if $2^q > \sigma$ then one may assign several codewords to a given label.

The resulting table, for our example, is given in Table 2.

X Y	0	1
0 0	0 1 , 1 1	0 0 , 1 1 , 1 0
0 1	1 1	-
1 1	1 0	-
1 0	0 0	-

Table 2.

The coded table for Example 2.

Now, the table is translated into an equation of the form (1), where the left hand side is expanded with respect to y_1, y_2, dy_1, dy_2.

$$\begin{aligned}
&y_1y_2dy_1dy_2(1) + y_1y_2dy_1d\bar{y}_2(x) + y_1y_2d\bar{y}_1dy_2(1) + y_1y_2d\bar{y}_1d\bar{y}_2(1) + \\
&+ y_1\bar{y}_2dy_1dy_2(1) + y_1\bar{y}_2dy_1d\bar{y}_2(1) + y_1\bar{y}_2d\bar{y}_1dy_2(1) + y_1\bar{y}_2d\bar{y}_1d\bar{y}_2(x) + \\
&+ \bar{y}_1y_2dy_1dy_2(x) + \bar{y}_1y_2dy_1d\bar{y}_2(1) + \bar{y}_1y_2d\bar{y}_1dy_2(1) + \bar{y}_1y_2d\bar{y}_1d\bar{y}_2(1) + \\
&+ \bar{y}_1\bar{y}_2dy_1dy_2(0) + \bar{y}_1\bar{y}_2dy_1d\bar{y}_2(\bar{x}) + \bar{y}_1\bar{y}_2d\bar{y}_1dy_2(x) + \bar{y}_1\bar{y}_2d\bar{y}_1d\bar{y}_2(\bar{x}) = 0
\end{aligned}$$

which is equivalent to:

$$xd\bar{y}_1dy_2 + \bar{x}\bar{y}_1d\bar{y}_2 + y_2d\bar{y}_1 + y_1\bar{y}_2dy_1 + y_1dy_2 + xy_2 = 0.$$

Definition 2.2. The graph G is said to be solvable with respect to Σ with a finite delay d if the knowledge of the first d labels of the tape $\ell(1)$, $\ell(2)$, ..., $\ell(d)$ is sufficient to determine the initial vertex V(1), and for every t, the knowledge of V(t), $\ell(t)$, $\ell(t + 1)$, ..., $\ell(t+d)$ is sufficient to determine a vertex V(t + 1), so that the determined part of the sequence of vertices is part of a path with the given tape of labels.

The same construction, demonstrated by Examples 1 and 2 above, shows that the problem of deciding whether an equation is solvable with a finite delay d is equivalent to the problem of deciding whether a graph (with $\sigma=2^q$) is solvable with a finite delay d.

Before we continue with the description of our solution for the two decision problems, two remarks concerning related problems should be made:

(1) The variables of equation (1) were assumed to be binary, namely, over $\{0, 1\}$. However, any equation of the form (1) with variables over any finite alphabet may be constructed. All such equations

are representable by a graph, and both decision problems for them are equivalent to the decision problems for graphs.

(2) There is some interest in solving equations of the form (1) under a restricted set of sequences of X. Clearly, in these cases $x_1, x_2, \ldots, x_m$ are not independent variables. The problem can still be translated to an equivalent problem on a graph, where the restrictions are translated to restriction on tapes. Sometimes this restriction can be expressed by an augmented graph.

III. An Algorithm for Deciding if a Graph is Solvable and the Specification of a Solution.

The problem of deciding whether a given graph is solvable is equivalent to the problem of deciding whether the corresponding multipath automaton accepts all tapes. The concept of multipath automata (nondeterministic automata) and the technique used here has been originated by Rabin and Scott [6] and discussed by others. (For example, see reference 7.)

We define a multipath automaton A, corresponding to a graph G with labels from Σ as follows:

(1) The set of states is V.

(2) The input alphabet is Σ.

(3) The set of next states for state s and input letter ℓ is given by $\Gamma(\{s\},\ell)$.

(4) The set of initial states is V.

(5) The set of final (accepting) states is V.

Clearly, if a tape is accepted by A then there exists a path labeled with this tape in G. Therefore, the problem of deciding whether G is solvable reduces to the problem of deciding whether A accepts all tapes over Σ. This means that in the construction of D(A), (the single path, or deterministic equivalent of A,) the empty set should never be generated. In case it is generated, the corresponding graph is unsolvable. Otherwise, it is solvable.

Let us demonstrate the procedure on Example 2. The transition table of D(A) is shown in Table 3. Since

	0	1
$\alpha\beta\gamma\delta$	$\alpha\beta\gamma\delta$	$\alpha\gamma\delta$
$\alpha\gamma\delta$	$\alpha\beta\gamma\delta$	$\alpha\gamma\delta$

Table 3.
Generation of D(A) for Example 2.

the empty set of states has not been generated, all tapes are accepted by the automaton, namely, G is solvable. The reader may test Example 1 and will find that its generation table has only one line.

The construction of D(A) also yields a simple solution to the following problem: Given a specific tape $\ell(1)$, $\ell(2), \ldots, \ell(L)$, find a path in G with this tape of labels. Let S(1) be the set of all vertices of G. For every t = 1, 2, ..., L, let $S(t + 1) = \Gamma\ (S(t), \ell(t))$. The result of this construction is a sequence S(1), S(2), ..., S(L), S(L + 1) such that all states in S(t + 1) are reachable by a path labeled $\ell(1)$, $\ell(2), \ldots, \ell(t)$. Now, to find a path which satisfies the requirement, simply trace it backwards on the sequence as follows: Choose any element of S(L + 1). (If this set is empty, there is no path satisfying the requirement.) Call it V(L + 1). Assuming that V(t + 1) is known, choose V(t) to be any vertex of S(t) which leads to V(t + 1) with an edge labeled $\ell(t)$. Clearly, there must be at least one such element in S(t) if V(t + 1) is an element of S(t + 1).

Example 3. Consider the graph whose transition table is given in Table 4. Assume that

	0	1
α	γ	δ
β	δ	γ
γ	α,β	-
δ	-	γ,δ

Table 4.
Transition table for Example 3

it is required to find a path for the tape 1 1 0 1. The sequence of sets of vertices is as follows:

$$\left\{\alpha,\beta,\gamma,\delta\right\}\xrightarrow{1}\left\{\gamma,\delta\right\}\xrightarrow{1}\left\{\gamma,\delta\right\}\xrightarrow{0}\left\{\alpha,\beta\right\}\xrightarrow{1}\left\{\gamma,\delta\right\}.$$

Let us choose v(5) = γ. The vertex γ is entered by edges labeled 1 and emanating from β and δ. However, δ is not a member of S(4). Thus, v(4) = β. v(3) = γ is unique, and so is v(2) = δ. For v(1) one can choose between α and δ. Therefore, the sequence $\alpha\delta\gamma\beta\gamma$ is one solution.

The solution to this problem, presented here, is close in method of approach to the methods discussed by Schutzenberger.[8]

IV. An Algorithm for Deciding if a Graph is Solvable with a Finite Delay and Specification of a Solution.

A graph G has a solution with delay zero if and only if there exists a non-empty set of vertices $V' \subset V$ such that for all $\ell \in \Sigma$ and $v \in V'$,

$\Gamma(\{v\},\ell) \cap V' \neq \emptyset$. This condition yields a simple test for the existence of a solution with delay $d = 0$. Let $V_1 = \{v|\exists\ell\epsilon\Sigma \ni \Gamma(\{v\},\ell) = \emptyset$. Now, let Γ_1 be Γ , restricted to $V - V_1$. Again, let

$$V_2 = \{v|\exists\ell\epsilon\Sigma \ni \Gamma_1(\{v\},\ell) = \emptyset$$

and let Γ_2 be Γ_1 restricted to $(V - V_1) - V_2$, etc. There exists a solution for $d = 0$ if and only if for some $i \geq 2$ $V_i = \emptyset$ and $(\ldots((V - V_1) - V_2) - \ldots) - V_{i-1} \neq \emptyset$.

The test for $d > 0$ is similar in approach, but one uses the information about the next $d + 1$ letters before making the transition to the next states. Formally, a set of graphs G^i for $i \geq 0$ are defined. By convention $G^0 = G$ and the rules for the construction of G^i are as follows:

(1) The vertices of G^i are the $(i + 1)$ - tuples $(v,\ell(1),\ell(2),\ldots,\ell(i))$ where $v \in V$ and $\ell(1),\ldots,\ell(i)$ are elements of Σ.

(2) Vertex $(v,\ell(1),\ell(2),\ldots.,\ell(i))$ is joined with vertex $(v', \ell(2), \ldots, \ell(i),\ell)$ with label ℓ if and only if $v' \in \Gamma(\{v\}, \ell(1))$. There are no other edges in G^i.

The test whether G has a solution with delay d consists simply in testing whether G^d has a solution with delay zero. The procedure is illustrated by the following example:

<u>Example 4</u>: Let G be given in Table 5.

	0	1
a	b,c	a,c
b	c	-
c	a	-

Table 5.
Description of G of Example 4.

States b and c transfer nowhere with label 1 , and therefore are eliminated from the table. Next, a has no transfer for label 0. The conclusion is that G has no solution with zero delay.

The graph G^1, constructed according to the above rules, is given in Table 6.

	0	1
a 0	b 0 , c 0	b 1 , c 1
a 1	a 0 , c 0	a 1 , c 1
b 0	c 0	c 1
b 1	-	-
c 0	a 0	a 1
c 1	-	-

Table 6
Description of G^1 of Example 4.

We now proceed to test G^1 for a solution with delay zero. In Table 7 the successive steps in the test of G^1 are shown. First b 1 and c 1 are

	0	1
a 0	b 0 , c 0	-
a 1	a 0 , c 0	a 1
b 0	c 0	-
c 0	a 0	a 1

(a)

	0	1
a 1	c 0	a 1
c 0	-	a 1

(b)

	0	1
a 1	-	a 1

(c)

Table 7
The test of G^1, Example 4.

eliminated and the resulting table is shown in Table 7 (a). Next a 0 and b 0 are eliminated, as in Table 7 (b), etc. Since all vertices of G^1 are eliminated, G has no solution of order 1. The tests for G^2 and G^3 are shown in Tables 8 and 9.

	0	1
a 0 0	b 0 0 , c 0 0	b 0 1 , c 0 1
a 0 1	b 1 0 , c 1 0	b 1 1 , c 1 1
a 1 0	a 0 0 , c 0 0	a 0 1 , c 0 1
a 1 1	a 1 0 , c 1 0	a 1 1 , c 1 1
b 0 0	c 0 0	c 0 1
b 0 1	c 1 0	c 1 1
b 1 0	-	-
b 1 1	-	-
c 0 0	a 0 0	a 0 1
c 0 1	a 1 0	a 1 1
c 1 0	-	-
c 1 1	-	-

(a)

	0	1
a 0 0	b 0 0 , c 0 0	b 0 1 , c 0 1
a 0 1	-	-
a 1 0	a 0 0 , c 0 0	a 0 1 , c 0 1
a 1 1	a 1 0	a 1 1
b 0 0	c 0 0	c 0 1
b 0 1	-	-
c 0 0	a 0 0	a 0 1
c 0 1	a 1 0	a 1 1

(b)

	0	1
a 0 0	b 0 0 , c 0 0	c 0 1
a 1 0	a 0 0 , c 0 0	c 0 1
a 1 1	a 1 0	a 1 1
b 0 0	c 0 0	c 0 1
c 0 0	a 0 0	-
c 0 1	a 1 0	a 1 1

(c)

	0	1
a 0 0	b 0 0	c 0 1
a 1 0	a 0 0	c 0 1
a 1 1	a 1 0	a 1 1
b 0 0	-	c 0 1
c 0 1	a 1 0	a 1 1

(d)

Table 8
The test of G^2, Example 4.

	0	1
a 0 0	-	c 0 1
a 1 0	a 0 0	c 0 1
a 1 1	a 1 0	a 1 1
c 0 1	a 1 0	a 1 1

(e)

	0	1
a 1 0	-	c 0 1
a 1 1	a 1 0	a 1 1
c 0 1	a 1 0	a 1 1

(f)

	0	1
a 1 1	-	a 1 1
c 0 1	-	a 1 1

(g)

Table 8 (continued)
The test of G^2,Example 4.

	0	1
a 0 0 0	(4) ~~b 0 0 0~~ , c 0 0 0	b 0 0 1 , (3) ~~c 0 0 1~~
a 0 0 1	(2) ~~b 0 1 0~~ , c 0 1 0	(2) ~~b 0 1 1~~ , c 0 1 1
(2) ~~a 0 1 0~~	(1) ~~b 1 0 0~~ , (1) ~~c 1 0 0~~	(1) ~~b 1 0 1~~ , (1) ~~c 1 0 1~~
(2) ~~a 0 1 1~~	(1) ~~b 1 1 0~~ , (1) ~~c 1 1 0~~	(1) ~~b 1 1 1~~ , (1) ~~c 1 1 1~~
a 1 0 0	a 0 0 0 , c 0 0 0	a 0 0 1 , (3) ~~c 0 0 1~~
a 1 0 1	(2) ~~a 0 1 0~~ , c 0 1 0	(2) ~~a 0 1 1~~ , c 0 1 1
a 1 1 0	a 1 0 0 , (1) ~~c 1 0 0~~	a 1 0 1 , (1) ~~c 1 0 1~~
a 1 1 1	a 1 1 0 , (1) ~~c 1 1 0~~	a 1 1 1 , (1) ~~c 1 1 1~~
(4) ~~b 0 0 0~~	c 0 0 0	(3) ~~c 0 0 1~~
b 0 0 1	c 0 1 0	c 0 1 1
(2) ~~b 0 1 0~~	(1) ~~c 1 0 0~~	(1) ~~c 1 0 1~~
(2) ~~b 0 1 1~~	(1) ~~c 1 1 0~~	(1) ~~c 1 1 1~~
(1) ~~b 1 0 0~~	-	-
(1) ~~b 1 0 1~~	-	-
(1) ~~b 1 1 0~~	-	-
(1) ~~b 1 1 1~~	-	-
c 0 0 0	a 0 0 0	a 0 0 1
(3) ~~c 0 0 1~~	(2) ~~a 0 1 0~~	(2) ~~a 0 1 1~~
c 0 1 0	a 1 0 0	a 1 0 1
c 0 1 1	a 1 1 0	a 1 1 1
(1) ~~c 1 0 0~~	-	-
(1) ~~c 1 0 1~~	-	-
(1) ~~c 1 1 0~~	-	-
(1) ~~c 1 1 1~~	-	-

Table 9
The test of G^3, Example 4.

In Table 9 the successive reductions were performed by crossing out entries, and the stage at which an entry was crossed out is shown in parentheses. The resulting table is given in Table 10. Since it is not empty, G has a solution with delay 3, and it is specified by Table 10.

	0	1
a 0 0 0	c 0 0 0	b 0 0 1
a 0 0 1	c 0 1 0	c 0 1 1
a 1 0 0	a 0 0 0 , c 0 0 0	a 0 0 1
a 1 0 1	c 0 1 0	c 0 1 1
a 1 1 0	a 1 0 0	a 1 0 1
a 1 1 1	a 1 1 0	a 1 1 1
b 0 0 1	c 0 1 0	c 0 1 1
c 0 0 0	a 0 0 0	a 0 0 1
c 0 1 0	a 1 0 0	a 1 0 1
c 0 1 1	a 1 1 0	a 1 1 1

Table 10

The solution with d=3 for Example 4.

The preceding test combined with an upper bound for the delay in any graph with N vertices will yield a procedure for testing whether a solution with finite delay exists. We now proceed to derive such an upper bound.

Definition 4.1: Let G be a graph labeled by an alphabet Σ. The k-merge of G, G_k is the following graph:

(a) The vertices of G_k are V, the same as those of G.

(b) The label-alphabet Σ^k is the set of all label sequences of Σ, of length k.

(c) v_i is connected in G_k to v_j with an arc labeled $\ell(1)\ \ell(2)\ldots\ell(k)$ if and only if there exists a path from v_i to v_j in G labeled with the same sequence.

Informally, the difference between G^k and G_k is that in the former the transitions are made in a way similar to the transitions in G with the same label-alphabet, except that one "looks ahead" k + 1 places before making a decision as to the first step. In the latter one jumps k places all at once.

Lemma 1: If G is solvable with delay d then G_d is solvable with delay 1.

Proof: For G_d to be solvable with delay 1 a safe transition has to be made when faced with two letters of Σ^d, namely, 2d letters of Σ. Since G is d solvable, the knowledge of 2d letters enables one to decide upon d transitions, namely, upon a transition in G_d.

Q.E.D.

Lemma 2: If G_k is solvable with delay 1 then G is solvable with delay (at most) 2k - 1.

Proof: If a safe transition can be made in G_k upon the knowledge of two letters of Σ^k, then a safe transition can be made in G upon the knowledge of 2k letters. (In this case, not only one transition can be made, but k of them; however, it is not known whether another transition can be made before k more letters are known.)

Q.E.D.

Definition 4.2: Two graphs G' and G" are said to be similar if they have the same set of vertices (namely, V' = V") and for every letter $\ell' \in \Sigma'$ (the label alphabet of G') there exists a letter $\ell'' \in \Sigma''$ (the label alphabet of G") such that for every $v \in V$, $\Gamma'(\{v\},\ell') = \Gamma''(\{v\},\ell'')$, and for every $\ell'' \in \Sigma''$ there exists a $\ell' \in \Sigma'$ for which the same condition holds.

Note that the number of letters in Σ' is not necessarily the same as that in Σ''. Also, the number of possible functions $f(v) = \Gamma(\{v\},\ell)$ is 2^{N^2}, where $N = \#(V)$. This follows from the fact that for each $v, \Gamma(\{v\},\ell)$ may have 2^N possible values, and since there are N vertices, the total number of functions is $(2^N)^N = 2^{N^2}$.

Lemma 3: If G' and G" are similar and G' is solvable with delay 1, then G" is solvable with delay 1.

Proof: Define a relation * between the letters of Σ' and those of Σ'' as follows: for $\ell' \in \Sigma'$ and $\ell'' \in \Sigma''$,

$$\ell' * \ell'' \Leftrightarrow \forall\, v \in V\ [\Gamma'(\{v\}, \ell') = (\{v\}, \ell'')].$$

From the similarity it follows that for every ℓ' there exists an ℓ'', and for every ℓ'' there exists an ℓ' such that $\ell' * \ell''$. This relation can be extended to tapes in the natural way.

Now, assume the tape $\ell''(1), \ell''(2), \ldots, \ell''(L)$ is given. To demonstrate the fact that G" is solvable with delay 1 simply follow these steps:

(1) Let the initial vertex be one of the initial vertices chosen in G' when the first label on the tape is $\ell'(1)$, where $\ell'(1) * \ell''(1)$. Call it $v(1)$.

(2) Let $\ell'(i)$ be any letter of Σ' which satisfies $\ell'(i) * \ell''(i)$. Choose $v(i)$ as the next vertex in G' when the present vertex is $v(i - 1)$ and the first two letters are $\ell'(i - 1)$ and $\ell'(i)$. (This step is followed for all $2 \leq i \leq L$.)

Because of the similarity, the fact that the choice is safe for G' implies that it is safe for G". Q.E.D.

Consider now the sequence of graphs: $G = G_1, G_2, \ldots, G_k, \ldots$

Lemma 4: If G_i is similar to G_j then for every $h > 0$, G_{i+h} is similar to G_{j+h}.

Proof: $\Gamma(\{v\}, \ell(1)\ \ell(2) \ldots \ell(j+h))$ is the set of vertices connected to v with an arc labeled $\ell(1)\ \ell(2) \ldots \ell(j+h)$ in G_{j+h}. By definition,

$$\Gamma(\{v\}, \ell(1) \ldots \ell(j+h)) = \Gamma(\Gamma(\{v\}, \ell(1) \ldots \ell(j)), \ell(j+1) \ldots \ell(j+h)).$$

Since G_i is similar to G_j there exists a tape $\ell'(1) \ldots \ell'(i)$ such that for all $v \in V$

$$\Gamma(\{v\}, \ell(1) \ldots \ell(j)) = \Gamma(\{v\}, \ell'(1) \ldots \ell'(i)).$$

Thus,

$$\Gamma(\{v\}, \ell(1) \ldots \ell(j+h)) = \Gamma(\Gamma(\{v\}, \ell'(1) \ldots \ell'(i)), \ell(j+1) \ldots \ell(j+h))$$
$$= \Gamma(\{v\}, \ell'(1) \ldots \ell'(i)\ \ell(j+1) \ldots \ell(j+h)).$$

The same argument may be repeated with the roles of i and j reversed.

Q.E.D.

The total number of non-similar graphs over V is bounded by the following argument: There are 2^{N^2} functions one can assign to each letter. The set of functions associated with each graph can be chosen in $2^{2^{N^2}}$ ways. Note that the number of letters to which the same function is assigned is of no interest here as long as there is one letter to which the function is assigned. This is due to the fact that the graphs are compared up to similarity only.

Thus in the sequence of graphs, G_1, G_2,... only a finite number, at most $2^{2^{N^2}}$, non-similar graphs may appear. Also, by Lemma 4, once a graph appears in the sequence which is similar to a previous one, the sequence has entered a period, as far as similarity is concerned. By Lemma 3 it follows that if any G_i in the sequence is solvable with delay 1 then some G_i with $i \leq 2^{2^{N^2}}$ is solvable with delay 1. By Lemma 1, if G is solvable with a finite delay then for some $i \leq 2^{2^{N^2}}$, G_i is solvable with delay 1. It follows, by Lemma 2 that the delay d satisfies the condition

$$d \leq 2 \cdot 2^{2^{N^2}} - 1 ,$$

or

$$d \leq 2^{2^{N^2}+1} - 1 .$$

The following theorem summarizes the results of this section:

Theorem: There exists an algorithm for deciding whether a given graph with N vertices is solvable with a finite delay. In case the delay is finite, it is at most $2^{2^{N^2}+1} - 1$.

A straightforward method for deciding upon the existence of a solution with finite delay consists simply in testing G^B for a solution with delay zero, where B is any known upper bound for the size of finite delay. However, since we have shown only that $B \leq 2^{2^{N^2}+1} - 1$, this approach is impractical.

Alternatively, one may successively test the k-merges for a solution with delay one until one discovers a G_k similar to G_i for some $i < k$. This computation may be substantially shortened by observing that if ℓ_1 and ℓ_2 are labels in any graph such that $\Gamma(\{v\}, \ell_1) \supset \Gamma(\{v\}, \ell_2)$ for all vertices v, then label ℓ_1 may be ignored. We illustrate this with an example.

Example 5. Let G be the graph given by Table 11.

	0	1	2	3
a	bc	ac	bc	abc
b	c	-	-	a
c	a	-	ab	b

Table 11.
Description of G of Example 5.

It is easy to see that G is solvable. We want to test G for a solution with finite delay.

The transitions for label 3 include those of label 1, and so we eliminate label 3. The new graph, which we call G', is given in Table 12. The test given at the beginning of this section may be used to show that G' has no solution with delay one.

	0	1	2
a	bc	ac	bc
b	c	-	-
c	a	-	ab

Table 12.
Description of G' of Example 5.

We now wish to test the 2-merge of G for a solution with delay one. However, since the transitions of any label of G_2 of the form $\ell 3$ where $\ell=0,1,2,3$ will include the transitions of $\ell 1$ (and similarly for 3ℓ and 1ℓ), we actually need only test the 2-merge of G'. The 2-merge of G', which we call G_2', is given in Table 13(a). Observing that the transitions for labels 00, 10, 12, and 20 include those of 11 (and similarly 02 includes 22, and 21 includes 01), we eliminate labels to obtain G_2'' given in Table 13(b).

Again, the test of G''_2 for a solution with delay one gives a negative result.

	00	01	02	10	11	12	20	21	22
a	ac	-	ab	abc	ac	abc	ac	-	ab
b	a	-	ab	-	-	-	-	-	-
c	bc	ac	bc	-	-	-	bc	ac	bc

(a)

	01	11	22
a	-	ac	ab
b	-	-	-
c	ac	-	bc

(b)

Table 13

Description of G'_2 and G''_2 of Example 5.

Continuing, we wish to test G_3, but it is sufficient to test a graph G'_3 whose labels consist only of length three tapes of the form $x\cdot\ell$ where x is a label of G''_2 and ℓ is a label of G'. G'_3 is given in Table 14(a), and after eliminating labels in G'_3 we obtain G''_3 given in Table 14(b). G''_3 has no solution with delay one.

Finally, we construct G'_4 (Table 15(a)) from the labels of G''_3 and G', and then G''_4 (Table 15(b)) by eliminating labels. Since G''_4 is similar to G''_2, we conclude that G''_4 has no solution with delay one, and therefore neither does G''_k for $k > 4$. Hence G has no solution with finite delay.

	010	011	012	110	111	112	220	221	222
a	-	-	-	abc	ac	abc	bc	ac	bc
b	-	-	-	-	—	-	-	-	-
c	abc	ac	abc	-	-	-	ac	-	ab

(a)

Table 14.

	011	111	222
a	-	ac	bc
b	-	-	-
c	ac	-	ab

(b)

Table 14 continued

Description of G_3' and G_3'' of Example 5.

	0110	0111	0112	1110	1111	1112	2220	2221	2222
a	-	-	-	abc	ac	abc	ac	-	ab
b	-	-	-	-	-	-	-	-	-
c	abc	ac	abc	-	-	-	bc	ac	bc

(a)

	0111	1111	2222
a	-	ac	ab
b	-	-	-
c	ac	-	bc

(b)

Table 15.

Description of G_4' and G_4'' of Example 5.

The procedure applied to Example 5 can be further improved in several ways. Nevertheless there are small graphs for which an enormous amount of computation is required to test the k-merges. We conjecture that the best bound on the size of finite delay grows more slowly than $2^{2^{N^2}}$.

Our next aim is to show that the upper bound for finite delay must be at least N(N-2). We describe a family of graphs, H_N, where H_N has N vertices and is solvable with a finite delay d = N (N - 2) but with no smaller delay. The structure of H_N for N = 4,5,... is shown in Table 16 and Figure 3.

	0	1
V_1	V_2, V_3	$V_1, V_2, \ldots, V_N$
V_2	V_3	-
$\vdots$	$\vdots$	$\vdots$
V_i	V_{i+1}	-
$\vdots$	$\vdots$	$\vdots$
V_{N-1}	V_N	-
V_N	V_1	-

Table 16
Description of H_N.

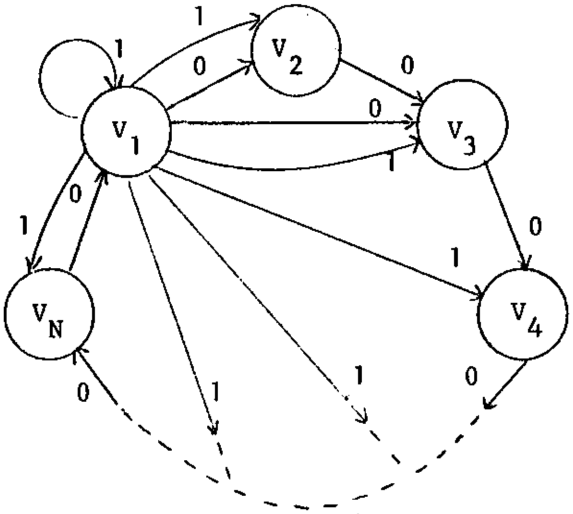

Figure 3
The graph H_N.

It is easy to see that H_N is solvable. It remains to show that it is solvable with a delay d = N (N - 2). For that purpose consider tapes of the form $0^m\,1$. For m = 0 one may not start at $V_2, V_3, \ldots, V_N$. For m = 1 may not start at $V_1, V_2, \ldots, V_{N-1}$, etc. Table 17 indicates for each m the vertices which cannot be used as the start vertex for a path labeled $0^m 1$. It shows, for example, that V_3 cannot be the start vertex for the tape $0^{N(N-2)-1}\,1$ and V_2 cannot be the start vertex for $0^{N(N-2)}\,1$.

m	Non-initial Vertices
0	$v_2, v_3, \ldots, v_N$.
1	$v_1, v_2, \ldots, v_{N-1}$.
2	$v_N, v_1, \ldots, v_{N-2}$.
⋮	⋮
N-3	$v_5, v_6, \ldots, v_N, v_1, v_2, v_3$.
N-2	$v_4, v_5, \ldots, v_N, v_1, v_2$.
N-1	$v_3, v_4, \ldots, v_N$.
⋮	⋮
2N-4	$v_6, v_7, \ldots, v_N, v_1, v_2, v_3$.
2N-3	$v_5, v_6, \ldots, v_N, v_1, v_2$.
2N-2	$v_4, v_5, \ldots, v_N$.
⋮	⋮
(N-3) (N-1)	v_{N-1}, v_N.
⋮	⋮
(N-3) (N-1) + (N-2)	v_1, v_2.
(N-2) (N-1)	v_N.
⋮	⋮
(N-2) (N-1) + (N-2)	v_2.
(N-2) N + 1	—

Table 17

Non-initial vertices of H_N for 0^m1.

Thus, once forced into vertex v_1, and faced with $0^{N(N-2)}$ as the prefix of the forthcoming tape, no decision can be made as to the choice of the next state. Vertex v_3 is forbidden, since a 1 on the next place in the tape may occur; and vertex v_2 is forbidden, since a 01 on the next places in the tape may occur. Furthermore, a long sequence of 0's leads to v_1 within N-1 letters. Thus, $d \geq N(N-2)$.

In order to show that H_N has a solution with delay $N \cdot (N-2)$, we indicate how to choose a path with this delay.

First, consider tapes which begin with 0^m1 for any m, $0 \leq m \leq N \cdot (N-2)-1$. The initial vertex in a path for such a tape may be any vertex which is not forbidden (by Table 17) for 0^m1. Similarly, if a tape begins with $0^{N \cdot (N-2)}$, the initial vertex may be any vertex except v_2.

Now suppose that a path through H_N has been specified as far as some vertex v and that the remainder of the tape which led to v begins with $\ell 0^m 1$, where $\ell=0$ or 1 and $0 \leq m \leq N \cdot (N-2)-1$. The next vertex in the path may be any vertex in $\Gamma(\{v\},\ell)$ which is not forbidden for 0^m1. If the remainder of the tape begins with $\ell 0^{N \cdot (N-2)}$, the next vertex may be any vertex in $\Gamma(\{v\},\ell)$ except v_2. That there will always exist a possible next vertex in $\Gamma(\{v\},\ell)$ follows from the construction of Table 17 and our choice of initial vertices. We omit a formal proof.

Example 2 is H_4. An equation corresponding to it is given by (8). Since d=8 in this case, it constitutes a counter example to the statement of Theorem 6 in Wang's paper.*

* According to Theorem 6 $d \geq 2^{2m}(2^n - n - 1) + 1$. In Example 2, m = 1 and n = 2. Thus, d should be ≤ 5, which it is not. The mistake is not correctable by a fixed coefficient, since for m = 1 it would imply $d \leq 4(N - \lfloor \log_2 N \rfloor - 1) + 1$ which is asymptotically smaller than N(N-2).

Acknowledgments

The authors would like to thank Professor H. Wang (now with Rockefeller University, N. Y.) for the introduction to the problem; Mr. I. Golan (Technion, Haifa, Israel) for the participation in the early stages of this research; Professor W. Semon, for providing S. Even with the facilities, and participating in the research while S. Even was in Syracuse University during the summer of 1967; Dr. G. Ott (Sperry Rand Research Center, Sudbury, Massachusetts) for participation in the later stages of this research.

References

(1) Wang, Hao, "Circuit Synthesis by Solving Sequential Boolean Equations", Zeitshr. f. mat. Logik und Grundlagen d. Math. Bd. 5, S. 291-322 (1959). Also in A Survey of Mathematical Logic, by Hao Wang, Studies in logic and the foundations of mathematics, Science Press, Peking, and North-Holland Publishing Company, Amsterdam, 1963, pp. 269-305.

(2) Lewis, C. I., A Survey of Symbolic Logic, Berkeley, 1918 (Dover, 1960, pp. 144-162).

(3) Phister, M., Jr., Logical Design of Digital Computers, John Wiley and Sons, 1959.

(4) Ashenhurst, R. I., "Simultaneous Equations in Switching Theory", Theory of Switching, Report No. BL-5, Section 1, Harvard Computation Laboratory, 1954.

(5) Birkhoff, Garrett, and Saunders MacLane, A Survey of Modern Algebra, The Macmillan Company, 1947, Section 5, Chapter XI.

(6) Rabin, M. O., and D. Scott, "Finite automata and their decision problems", IBM Journal of Research and Development, Vol. 3, No. 2, 1959, pp. 114-125. Also in Sequential Machines - Selected Papers, edited by E. F. Moore, Addision-Wesley Publishing Co., 1964, pp. 63-91.

(7) Ginsburg, S., An Introduction to Mathematical Machine Theory, Addison-Wesley Publishing Co., 1962, Section 4.2.

(8) Schutzenberger, M. P., "A Remark on Finite Transducers". Information and Control 4, pp. 135-196, 1961.

APPENDIX

SOLUTION OF BOOLEAN EQUATIONS

Let a, b, ..., c be independent Boolean variables (fixed elements of a Boolean algebra) and x, y, ..., z be dependent Boolean variables (unknown elements of the same Boolean algebra). Let G(a, b, ..., c, x, y..., z) and H(a, b, ..., b, x, y, ..., z) be any two expressions of these variables.

Definition: A set of values for x, y, ..., z is said to constitute a solution of the equation G = H if when the values are substituted into the equation, the equation becomes an identity.

The problem of solving Boolean equations was settled in the beginning of this century and is discussed by Lewis [2]. More recently, several other authors have discussed the problem, for example, Phister [3] and Ashenhurst [4]. Nevertheless, the subject remains generally unknown and unused. We present a brief exposition in an effort to clarify the subject, assuming the reader is already familiar with the elementary properties of Boolean algebra (as used, for example in the logical design of circuits). Clearly, there is no claim of originality in the appendix.

Theorem A1: The equation $\sum_{i=1}^{n} G_i = 0$ is equivalent to the simultaneous set of equations $G_i = 0$ for i = 1, 2, ... n.

Proof: Clearly $G_i = 0$ for $i = 1, 2, \ldots, n$ implies $\sum_{i=1}^{n} G_i = 0$. Now assume $\sum_{i=1}^{n} G_i = 0$. Multiply the equation by G_k.

$$0 = \sum_{i=1}^{n} G_k G_i = G_k + \sum_{i \neq k} G_k G_i \geq G_k .$$ Hence, $G_k = 0$.

Here idempotency and absorption were used. Q.E.D.

Theorem A2: The equation $G = H$ is equivalent to the equation $G\bar{H} + \bar{G}H = 0$.

Proof: $G\bar{H} + \bar{G}H = 0 \iff G\bar{H} = 0$ and $\bar{G}H = 0$ (Theorem A1).

$\iff G\bar{H} = 0$ and $G + \bar{H} = 1$.

$\iff \bar{H}$ is the complement of G.

$\iff G = H$. Q.E.D.

In view of Theorem A2, it is sufficient to consider equations of the form $F(a, b, \ldots, c, x, y, \ldots, z) = 0$.

Theorem A3: (The expansion theorem)

$$F(a, b, \ldots, c, x, y, \ldots, z) = xF(a, b, \ldots, c, 1, y, \ldots, z) + \bar{x}F(a, b, \ldots, c, 0, y, \ldots, z).$$

(The proof is through the disjunctive canonical form. [5])

Thus, every equation may be transformed into the form $xA + \bar{x}B = 0$ where A and B are expressions in $a, b, \ldots, c$ and $y, \ldots, z$, but without x.

Theorem A4: The equation $xA + \bar{x}B = 0$ has a solution if and only if $AB = 0$ has a solution.

Proof: Assume $x, y, \ldots, z$ constitutes a solution of $xA + \bar{x}B = 0$.

$$0 = (AB)0 = AB(xA + \bar{x}B) = xAB + \bar{x}AB = AB(x + \bar{x}) = AB.$$

Now assume $y, \ldots, z$ constitute a solution of the equation $AB = 0$. We observe that the value $x = B$ with the given $y, \ldots, z$ constitute a solution of $xA + \bar{x}B = 0$. Q.E.D.

Theorem A5: Let $xA + \bar{x}B = 0$ be a solvable equation. For every element m of the Boolean algebra, $x = B + \bar{A}m$ is a part of a solution to the original equation.

By "a part of a solution" is meant that there exist values for $y, \ldots, z$ which together with $x = B + \bar{A}m$ constitute a solution.

Proof: Use the values of $y, \ldots, z$ which constitute a solution for the equation $AB = 0$. (That this equation has a solution is the consequence of (Theorem A4.) Also use $x = B + \bar{A}m$. On substitution in the original equation one gets: $(B + \bar{A}m)A + \bar{B}(A + \bar{m})B = 0$. Q.E.D.

Theorem A6: If x is a part of a set of values which constitute a solution of the equation $xA + \bar{x}B = 0$, then there exists an element m of the Boolean algebra such that $x = B + \bar{A}m$.

Proof: $xA + \bar{x}B = 0 \Rightarrow xA = 0$ and $\bar{x}B = 0$.

$B + \bar{A}x = B + \bar{A}x + Ax = B + x = \bar{x}B + xB + x = x$.

Namely, the theorem is satisfied with $m = x$ itself. Q.E.D.

The conclusion of these six theorems is that

(1) Any set of simultaneous equations may be transformed to a single equation of the form $F(a, b, \ldots c, x, y, \ldots, z) = 0$.

(2) The equation may be transformed to the form $xA + \bar{x}B = 0$.

(3) We now consider the equation $AB = 0$. Any set of values of $y, \ldots, z$ which constitute a solution of $AB = 0$ may be augmented by $x = B + \bar{A}m$, where m is arbitrary, to form a solution of the original equation.

(4) By repeating this process as many times as there are unknowns, we reduce it to an equation with no unknowns.

If this equation is reduced to an identity, then the original equation is solvable and all solutions are as characterized.

If the last equation is not an identity, then the original equation has no solution.

Example: Assume we want to design a full serial adder with two binary inputs, a and b. The carry c must satisfy the equation: dc = ab + bc + ca. Here dc (t) = c(t + 1).

(We shall not concern ourselves with the design of the sum.) Assume we wish to use an R-S Flip-Flop with inputs R and S and output c which is characterized by the equations

$$\left.\begin{aligned} dc &= S + \bar{R}c \\ RS &= 0 \end{aligned}\right\}$$

The design problem is to express S and R in terms of a, b, and c. S and R must satisfy the simultaneous equations

$$S + \bar{R}c = ab + bc + ca$$

$$RS = 0$$

By Theorem A2 and elementary operations, one can show that the first equation is equivalent to

$$S(\bar{a}\bar{b} + \bar{b}\bar{c} + \bar{c}\bar{a} + \bar{R}\bar{a}\bar{b}c) + \bar{S}(Rab + Rbc + Rca + ab\bar{c} + \bar{R}\bar{a}\bar{b}c) = 0.$$

Use Theorem A1 to join the two equation into one:

$$S(\bar{a}\bar{b} + \bar{b}\bar{c} + \bar{c}\bar{a} + \bar{R}\bar{a}\bar{b}c + R) + \bar{S}(Rab + Rbc + Rca + ab\bar{c} + \bar{R}\bar{a}\bar{b}c) = 0.$$

By Theorem A4, R must satisfy the equation:

$$(\bar{a}\bar{b} + \bar{b}\bar{c} + \bar{c}\bar{a} + \bar{R}\bar{a}\bar{b}c + R)(Rab + Rbc + Rca + ab\bar{c} + \bar{R}\bar{a}\bar{b}c) = 0.$$

which is equivalent to

$$R(ab + bc + ca) + \bar{R}(\bar{a}\bar{b}\bar{c}) = 0.$$

The last equation has a solution if and only if $(ab + bc + ca)(\bar{a}\bar{b}c) = 0$, which is indeed an identity.

Thus,

$$R = \bar{a}\bar{b}c + (\bar{a}\bar{b} + \bar{b}\bar{c} + \bar{c}\bar{a})m_r$$

where m_r is an arbitrary switching function. Next, substitute this value of R into the equation for S. The resulting equation is

$$S(\bar{a}\bar{b} + \bar{b}\bar{c} + \bar{c}\bar{a}) + \bar{S}(ab\bar{c}) = 0.$$

and

$$S = ab\bar{c} + (ab + bc + ca)m_s.$$

In this example S is independent of m_r, but this is not always the case. If we choose

$$m_r = \bar{a}\bar{b}\bar{c} \qquad \text{and} \qquad m_s = abc, \qquad \text{then}$$

$$R = \bar{a}\bar{b} \qquad \text{and} \qquad S = ab,$$

which is a well known solution of this problem.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Carnegie-Mellon University Department of Computer Science Pittsburgh, Pennsylvania 15213	2a. REPORT SECURITY CLASSIFICATION UNCL 2b. GROUP

3. REPORT TITLE

SEQUENTIAL BOOLEAN EQUATIONS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

Even, Shimon and Albert R. Meyer

6. REPORT DATE March 1968	7a. TOTAL NO. OF PAGES 38	7b. NO. OF REFS 8

8a. CONTRACT OR GRANT NO. SD-146 ARPA b. PROJECT NO. 9718 c. 6154501R 681304 d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

--Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Air Force Office of Scientific Research 1400 Wilson Boulevard (SRI) Arlington, Virginia 22209

13. ABSTRACT

The problem of solving sequential Boolean equations is shown to be equivalent to the problem of finding whether there exists a path on a labeled graph for every sequence of labels. Algorithms are given for testing whether a solution exists, and if a solution with a finite delay exists. In case of existence of solutions the algorithms provide them.

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

Security Classification