

CONSTRUCTION OF GENERAL SUBSUMPTIVE SOLUTIONS OF BOOLEAN EQUATIONS VIA COMPLETE-SUM DERIVATION

Ali Muhammad Ali Rushdi and Hussain Mobarak Albarakati

Department of Electrical and Computer Engineering, Faculty of Engineering,
King Abdulaziz University, P.O. Box 80204, Jeddah 21589, Saudi Arabia

Received 2013-04-12; Revised 2013-07-16; Accepted 2014-02-15

ABSTRACT

Boolean-equation solving permeates many diverse areas of modern science. To solve a system of Boolean equations, one usually combines them into an equivalent single Boolean equation $\{f(\bar{X})=0\}$ whose set of solutions is exactly the same as that of the original system of equations. One of the general classes of solutions for Boolean equations is the subsumptive general solution, in which each variable is expressed as an interval decided by a double inequality in terms of the succeeding variables. The solution validity depends on the satisfaction of a required consistency condition. In this study, we introduce a novel method (henceforth called the CS method) for producing subsumptive Boolean-equation solutions based on deriving the complete sum $(CS(f(\bar{X})))$ of the pertinent Boolean function $f(\bar{X})$. The complete sum $CS(f(\bar{X}))$ is a disjunction of all prime implicants of $f(\bar{X})$ and nothing else. It explicitly shows all information about $f(\bar{X})$ in the most compact form. We demonstrate the proposed CS solutions in terms of four examples, covering Boolean algebras of different sizes and using two prominent methods for deriving $CS(f(\bar{X}))$. Occasionally, the consistency condition results in a collapse of the underlying Boolean algebra into a smaller subalgebra. We also illustrate how an expansion tree (typically reduced to an acyclic graph) can be used to deduce a complete list of all particular solutions from the subsumptive solution. The present CS method yields correct solutions, since it fits into the frame of the most general subsumptive solution. Among competing subsumptive methods, the CS method strikes a reasonable tradeoff between the conflicting requirements of less computational cost and more compact form for the solution obtained. In fact, it is the second best known method from both criteria of efficiency and compactness of solution.

Keywords: Boolean Equations, Subsumptive General Solutions, Complete Sum, Blake Canonical Form, Consensus Generation, Absorption, Multiplication

1. INTRODUCTION

Boolean-equation solving permeates many diverse areas of modern science such as biology, grammars, chemistry, law, medicine, spectrography and graph theory. It is also an indispensable tool in operations research, the cryptanalysis and breaking of ciphers, Boolean function decomposition, Boolean Satisfiability

(SAT) problem solving, the synthesis, simulation, testing and diagnosis of digital networks and VLSI systems, output encoding and state assignments of finite state machines, automatic test-pattern generation and many other subareas of logical design.

To solve a system of Boolean equations, the equations are usually combined into an equivalent single Boolean equation $\{f(\bar{X})=0\}$ whose set of solutions is

Corresponding author: Ali Muhammad Ali Rushdi, Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, P.O. Box 80204, Jeddah 21589, Saudi Arabia

exactly the same as that of the original system of equations. This is conceptually simpler and computationally more efficient than obtaining the set of solutions for each equation and then forming the intersection of such sets to obtain the set of solutions of the overall system. Typically, either general subsumptive solutions or general parametric solutions are sought, from which an exhaustive enumeration of all particular solutions can be readily chieved (Rudeanu, 1974; 2001; 2003; 2010; Brown, 1990; Levchenkov, 2000a; 2000b; Tucker and Tapia, 1992; Rushdi, 2001b; 2004; 2012; Rushdi and Amashah, 2010; 2011; 2012; Rushdi and Albarakati, 2013a). In this study, we are interested in deriving a subsumptive general solution of the Boolean equation:

$$f(\bar{X}) = 0 \tag{1}$$

where, $f(\bar{X}) : B^n \rightarrow B, \bar{X} = [X_1, X_2, \dots, X_n]^T$ and B is a general finite Boolean algebra of 2^q elements ($q = 1, 2, 3, \dots$) In a subsumptive general solution, each of the variables X_j is expressed as an interval determined by the double inequality:

$$u_j(X_{j+1}, X_{j+2}, \dots, X_n) \leq X_j \leq v_j(X_{j+1}, X_{j+2}, \dots, X_n) \tag{2}$$

$(j = 1, 2, \dots, n)$

i.e., each variable X_j is a partially- defined function of the succeeding $(n-j)$ variables. In particular, the last variable X_n is determined as an interval $\{u_n \leq X_n \leq v_n\}$ where u_n and v_n are elements of B . The subsumptive solutions (2) are usually obtained subject to a certain consistency condition.

Typically, the lower bound u_j and the upper bound v_j for X_j in (2) are determined in terms of successive conjunctive or disjunctive eliminants of the original function f (Rudeanu, 1974; Brown, 1990; Tucker and Tapia, 1992; Rushdi, 2001b; 2004). In this study, we introduce a novel class of subsumptive Boolean-equation solutions based on deriving the Complete Sum (CS(f)) or Blake Canonical Form (BCF(f)) (Blake, 1937; Tison, 1967; Rudeanu, 1974; 2001; Reusch, 1975; Muroga, 1979; Cutler et al., 1979; Brown and Rudeanu, 1988; Brown, 1990; Kean and Tsiknis, 1990; Gregg, 1998; Rushdi, 2001a; Rushdi and Al-Yahya, 2000; 2001a; 2002) of the pertinent Boolean function $f(\bar{X})$. This class of solutions fits into the frame of the most general form of the subsumptive general solution since it satisfies the necessary and sufficient conditions set in (Rudeanu, 2010) for such a form. The CS solution obtained herein

has two minor advantages over other known methods of subsumptive solutions, namely, (a) it explicitly casts the consistency condition in CS form and hence provides immediate complete information about it and (b) it allows nesting the subsumptions in (2) according to any desirable permutation of the set of integers $\{1, 2, \dots, n\}$

2. MATERIALS AND METHODS

We briefly review essential concepts of the complete sum of a switching function, outline two prominent algorithms for its derivation, present the mathematics of Boolean-function solution via complete sum derivation and demonstrate the proposed method with four illustrative examples. In these examples, we show how the original equation (1) is converted into the equivalent equation:

$$CS(f(\bar{X})) = 0 \tag{3}$$

We also demonstrate how to convert (3) into a subsumptive solution (2) together with a consistency condition. Occasionally, the consistency condition results in a collapse of the underlying Boolean algebra B into a smaller subalgebra. In each example, we illustrate how an expansion tree (typically reduced to an acyclic graph) can be used to deduce a complete list of all particular solutions from the subsumptive solution.

3. COMPLETE SUM OF A BOOLEAN FUNCTION

The complete sum of a Boolean function f , to be denoted by $CS(f)$ is the all-prime-implicant disjunction that expresses f , i.e., it is a sum-of products (s-o-p) formula whose products are all the prime implicants of f . The complete sum is called the “Blake Canonical Form” by Brown (1990) in honor of A. Blake who was the first author to initiate and develop this concept in his Ph.D. dissertation (Blake, 1937). Since all the prime implicants of f are present in $CS(f)$ it is obviously unique and hence stands for a canonical representation of the Boolean function. The complete sum for an Incompletely-Specified Boolean Function (ISBF) $f = g \vee d(h)$ is that of the associated Completely-Specified Boolean Function (CSBF) $F = g \vee h$. This means that a study of the complete sum always involves a CSBF and does not really involve an ISBF. Henceforth, when we refer to a Boolean function f , we understand it is a CSBF.

The concept of the complete sum of a Boolean function f is closely related to that of a syllogistic formula for f (Brown, 1990; Rushdi and Al-Yahya,

2001a). However, while $CS(f)$ is unique and canonical, there are infinitely many syllogistic formulas for f . A syllogistic formula of f can be defined as an s-o-p formula whose terms include, but are not necessarily excluded to, all the prime implicants of f , i.e., it is the complete sum of f disjuncted (possibly) with terms each of which subsumes some prime implicant of f . Other definitions and properties of a syllogistic formula is given by Brown (1990). Each of the following formulas are syllogistic formulas (Brown, 1990):

- A complete-sum formula
- An alterm (a disjunction of single literals)
- An s-o-p formula of monoform literals only
- An s-o-p formula such that no two terms in it have a consensus that does not appear in the formula.

If we compare the definition of a syllogistic formula for f to that of its complete sum $CS(f)$ we note that $CS(f)$ is minimal within the class of syllogistic formulas for f , i.e., the set of terms in any syllogistic formula for f is a superset of the set of terms in $CS(f)$ (Brown, 1990). Hence $CS(f)$ can be denoted by $ABS(F)$, where F is any syllogistic formula for f and $ABS(F)$ denotes an equivalent absorptive formula of F , i.e., a formula obtained from F by successive deletion of terms absorbed in other terms of F (Brown, 1990).

In view of our definition of $CS(f)$ as $ABS(F)$, it is obvious that $CS(f)$ may be generated by the following two-step procedure: (a) Find a syllogistic formula F for f and (b) Delete absorbed terms to obtain $ABS(F)$. Many techniques exist in the literature for carrying out step (a). These are categorized (Brown, 1990) into the three basic approaches of exhaustion of implicants, iterative consensus and multiplication. In the examples solved herein we employed two prominent algorithms for complete-sum derivation, namely, Tison algorithm and an algorithm utilizing the Variable-Entered Karnaugh Map (VEKM), which is succinctly labeled as VEKM folding.

3.1. Tison Algorithm

Tison method for obtaining all the prime implicants of a switching function F (i.e., obtainig) $CS(f)$ is a systematic streamlined version of the iterative-consensus technique. The original study of Tison appeared in (Tison, 1967), but a more readable exposition can be found in (Cutler *et al.*, 1979) or in (Muroga, 1979). The method is sometimes called “Tison method” for short, though its lengthier name serves to differentiate it from another Tison method, namely, that for the derivation of

all irredundant disjunctive forms (Muroga, 1979). The essence of the present Tison method is summarized in Theorem 1. This theorem is adapted from (Cutler *et al.*, 1979) to use the complete sum in the sense used by (Brown and Rudeanu, 1988), which is applicable to big Boolean algebras. In this theorem, the biform variables Y_1, Y_2, \dots, Y_m include the biform variables among the input variables \bar{X} and any biform generator among the algebra generators a, b, c, \dots

Theorem 1:

Start with a set of n_0 products $s_0 = \{T_1^{(0)}, T_2^{(0)}, \dots, T_{n_0}^{(0)}\}$ with m biform variables Y_1, Y_2, \dots, Y_m and a Boolean function f that is expressed by disjunction of the products in s_0 . For $1 \leq i \leq m$ repeat the following 2-part step that replaces a set of products s_{i-1} by an updated one s_i .

First, for $1 \leq j < k \leq n_{(i-1)}$ if Y_i appears complemented in one of the two products $T_j^{(i-1)}, T_k^{(i-1)}$ and appears uncomplemented in the other such that the two products have no other opposition, then they have a consensus with respect to Y_i . Form that consensus and add it to s_{i-1} . Finally, s_{i-1} is replaced by a superset $r_{(i-1)}$ of $p_{(i-1)}$ elements, where $p_{(i-1)}$ is greater than or equal to $n_{(i-1)}$.

Next, consider every pair $\{(T_j^{(i-1)}, T_k^{(i-1)}), \dots, j \neq k\}$ of (so far remaining) products in $r_{(i-1)}$. If $T_j^{(i-1)}$ subsumes $T_k^{(i-1)}$, then delete $T_j^{(i-1)}$. Otherwise, if $T_j^{(i-1)}$ is subsumed by $T_k^{(i-1)}$ then delete $T_k^{(i-1)}$. Whenever all subsumptions (and subsequent deletions) are exhausted, let the remaining set be $s_i = \{T_1^{(i)}, T_2^{(i)}, \dots, T_{n_i}^{(i)}\}$. The disjunction of products in any of the sets $s_i, 0 \leq i \leq m$ is an expression of f and the final set s_m consists of all prime implicants of f .

3.2. VEKM Folding

The variable-entered Karnaugh map (VEKM) is a useful tool that has a variable-handling capability better than that of the conventional Karnaugh map and that naturally handles general or big Boolean algebras (Rushdi, 1987; 1996; 2001a; 2001b; 2004; 2012; Rushdi and Amashah, 2011; Rushdi and Albarakati, 2013a; 2013b; Rushdi and Al-Yahya, 2000; 2001a; 2001b; 2002). In VEKM folding, a VEKM is used to represent the Boolean function and entries of the VEKM are converted into complete-sum entries via algebraic methods employing consensus generation and absorption (e.g., Tison method). In **Fig. 1a**, the function $f(\bar{X})$ is therefore assumed to have subfunctions or restrictions with respect to X_i which are already in CS form.

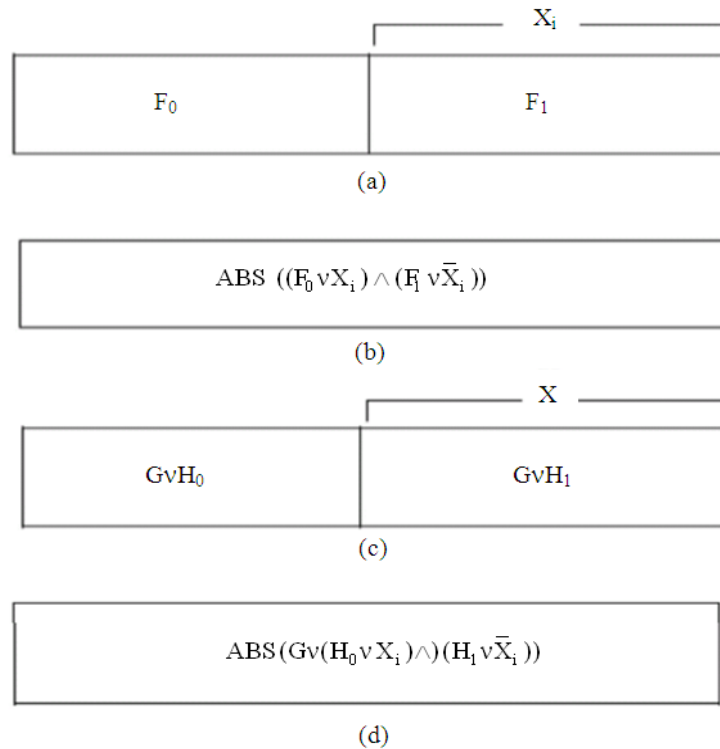


Fig. 1. The typical step of VEKM folding used in the derivation of CS(f) (a) $f(\bar{X}_i)$ with CS subfuncitons F_0 and F_1 (b) $f(\bar{X}_i)$ in CS form (c) $f(\bar{X}_i)$ with CS subfuncitons (GvH_0) and (GvH_1) (d) $f(\bar{X}_i)$ in CS form $f(\bar{X}_i)$

Figures 1a and 1b demonstrate the basic step in VEKM folding which converts a map variable X_i into an entered variable, while retaining CS entries in the new VEKM representation of the pertinent function. In **Fig. 1b**, we use $ABS(F)$ to denote an equivalent absorptive formula of F , i.e., a formula obtained from F by successive deletion of terms absorbed in other terms of F . The formula in **Fig. 1b** uses ANDing (multiplication) of CS formulas as an alternative for consensus generation. This multiplication is implemented via a multiplication matrix which allows an easy tracking of absorptions because of the fact that if a term is to be ever absorbed, then one of its absorbing terms will belong to either its row or to its column (Rushdi and Al-Yahya, 2001a). If the subfunctions F_0 and F_1 have some terms in common, i.e., if they can be written as $F_0 = G v H_0$ and $F_1 = G v H_1$ as shown in **Fig. 1c** where G is a disjunction of common terms, then “intelligent multiplication” ((Brown, 1990; Rushdi and Al-Yahya, 2001a) replaces $ABS((F_0 v X_i) ^ (F_1 v \bar{X}_i))$ in **Fig. 1b** by $ABS((G v (H_0 v X_i))(H_1 v \bar{X}_i))$ in **Fig. 1d**.

4. COMPLETE-SUM SOLUTIONS

We now introduce a novel class of subsumptive general solutions based on the derivation of the complete sum $F = CS(f)$ of the underlying function f in (1). This class produces a sequence of equations $F_j = 0$, ($j = 1, 2, \dots, n+1$) where F_1 is the complete sum of the original function f in (1) and $F_j = F_j(X_j, X_{j+1}, \dots, X_n)$ is expanded as:

$$F_j = P_j X_j v Q_j \bar{X}_j v R_j = 0 \tag{4}$$

Again with each of the coefficients P_j , Q_j and R_j being a function of $(X_{j+1}, X_{j+2}, \dots, X_n)$ and the final coefficients P_{n+1} , Q_{n+1} and R_{n+1} being elements of the underlying Boolean algebra B . The subsumptive solution for the variable X_j ($j = 1, 2, \dots, n$) is expressed by:

$$Q_j \leq X_j \leq \bar{P}_j \tag{5}$$

Provided the following consistency condition is satisfied:

$$F_{j+1} = R_j = 0 \tag{6}$$

The subsumptive solution is obtained by imposing the final consistency condition:

$$F_{n+1} = R_n = 0 \tag{7}$$

and working in reverse order ($j = n, (n-1), \dots, 1$) to solve (4) via (5) subject to (6). Note that in (6), we do not write F_{j+1} as $(P_j Q_j \vee R_j)$ in analogy with the Rudeanu algorithm in (Rudeanu, 2003), since $(P_j Q_j \leq R_j)$ as will be explained shortly. In fact, with $F = F_1$ being in complete-sum form, each $F_j (j = 1, 2, \dots, n+1)$ will be also in complete-sum form, i.e., each F_j will be a disjunction of all of its prime implicants (and nothing else). The Prime Implicants (PIs) of the complete sum formula F_j are of three types (Reusch, 1975; Thayse, 1978; Rushdi, 2001a).

The first type are PIs that have the un-complemented literal X_j . The disjunction of these PIs is $P_j X_j$ where:

$$P_j = CS(F_j(1_j)) \tag{8}$$

where $F_j(1_j)$ is the subfunction or restriction of $F_j(X_j, X_{j+1}, \dots, X_n)$ with X_j set to 1, i.e., $F_j(1_j) = (1, X_{j+1}, \dots, X_n)$.

The second type are PIs that have the complemented literal \bar{X}_j . Their disjunction is $Q_j \bar{X}_j$ where:

$$Q_j = CS(F_j(0_j)) \tag{9}$$

where $F_j(0_j)$ is the subfunction or restriction of $F_j(X_j, X_{j+1}, \dots, X_n)$ with X_j set to 0, i.e., $F_j(0_j) = (0, X_{j+1}, \dots, X_n)$.

The third type are PIs that are independent of the variable X_j (and hence have neither the literal X_j nor the literal \bar{X}_j). Their disjunction is R_j , where:

$$R_j = CS(F_j(0_j)F_j(1_j)) \tag{10}$$

where the product $F_j(0_j)F_j(1_j)$ is called the conjunctive eliminant of F_j with respect to $\{X_j\}$ (Brown, 1990), or the meet derivative of F_j with respect to X_j (Thayse, 1978). Equations 8-10 clearly relate the PIs of F_j to those of its subfunctions or restrictions $F_j(0_j)$ and $F_j(1_j)$ and were first noted in (Reusch, 1975). These equations confirm our earlier assertions that $P_j Q_j \leq R_j$ and that each $F_{j+1} = R_j$ is in a complete-sum form.

In actual implementation of (4), we directly arrange CS (f) in the form:

$$\begin{aligned} CS(f) &= F_1 = P_1 X_1 \vee Q_1 \bar{X}_1 \vee F_2 \\ &= P_1 X_1 \vee Q_1 \bar{X}_1 \vee P_2 X_2 \vee Q_2 \bar{X}_2 \vee F_3 \\ &= \bigvee_{j=1}^n (P_j X_j \vee Q_j \bar{X}_j) \vee F_{n+1} \end{aligned} \tag{4a}$$

and hence write the subsumptive solution (5) and the final consistency condition (7) simply by inspection. The order followed in the summation (ORing) in (4a) is not necessarily the natural order $\{1, 2, \dots, n\}$, but could be any permutation of it.

5. RESULTS

The present method of CS sub sumptive solution of Boolean equations utilizes a canonical representation that explicitly shows complete information about the pertinent Boolean function $f(\bar{X})$ in the most compact form. It is typically more efficient than the don't-care techniques in (Rushdi, 2001b; 2004), but while the CS solution obtains a rather compact solution that is not necessarily minimal, the don't-care techniques seek the most compact solution by using Boolean minimization methods.

The CS method is comparable in efficiency and compactness of solution to the Rudeanu method in (Rudeanu, 2003). Finally, the CS method might need slightly more effort than the conventional method based on constructing eliminants, but this extra effort pays off, since it results in a more compact solution and hence in easier generation of the tree (or acyclic graph) of particular solutions. The following examples demonstrate particular implementation details of the CS method and typical results obtained with it.

5.1. Example 1:

Let the function $f(\bar{X})$ in (1) be $f(X_1, X_2, X_3)$: $B_4^3 \rightarrow B_4$ where $B_4 = FB(a) = \{0, a, \bar{a}, 1\}$ and:

$$f(X_1, X_2, X_3) = \bar{a}\bar{X}_2\bar{X}_3 \vee \bar{a}\bar{X}_1\bar{X}_2X_3 \tag{11}$$

This function is already in CS form. It is an absorptive syllogistic formula in which the variable X_1 and the generator a are monoform, while the two bifform variables X_2 and X_3 generate no consensi since the only two terms involving them ($\bar{a}\bar{X}_2\bar{X}_3$ and $\bar{X}_1\bar{X}_2X_3$) have double opposition. We arrange the CS formula (11) as:

$$\begin{aligned} CS(f) &= (0)X_1\bar{v}(\bar{a} \vee X_2X_3)\bar{X}_1\bar{v}(0)X_2\bar{v} \\ &(\bar{a}\bar{X}_3)\bar{X}_2\bar{v}(0)X_3\bar{v}(0)\bar{X}_3\bar{v}(0) \end{aligned} \tag{12}$$

And hence obtain the subsumptive solution:

$$\begin{aligned}
 0 &= 0 \\
 0 &\leq X_3 \leq 1 \\
 \bar{a}\bar{X}_3 &\leq X_2 \leq 1 \\
 (\bar{a} \vee X_2 X_3) &\leq X_1 \leq 1
 \end{aligned}
 \tag{13}$$

The availability of CS formula (11) allows us to choose any appropriate nesting of variables. For example, instead of (12) we may write:

$$\begin{aligned}
 CS(f) &= (\bar{X}_1 X_2) X_3 \vee (\bar{a}\bar{X}_2) \bar{X}_3 \vee (0) X_2 \vee (0) X_1 \vee \\
 &(\bar{a}) \bar{X}_1 \vee (0)
 \end{aligned}
 \tag{14}$$

And hence obtain the alternative subsumptive solution:

$$\begin{aligned}
 0 &= 0 \\
 \bar{a} &\leq X_1 \leq 1 \\
 0 &\leq X_2 \leq 1 \\
 \bar{a}\bar{X}_2 &\leq X_3 \leq (X_1 \vee \bar{X}_2)
 \end{aligned}
 \tag{15}$$

A list of all particular solutions is neither compact nor insightful as a general solution. Such a list is produced via expansion trees from the general solutions. **Figure 2** shows the expansion tree used in producing all 21 particular solutions for $f = 0$ from the general subsumptive solution (13). To save space, we combined common nodes in the tree, thereby reducing it to an acyclic graph.

5.2. Example 2:

The function $(X_1, X_2, X_3): B_{16}^3 \rightarrow B_{16}$ which satisfies $\{f = 0\}$ is given by:

$$\begin{aligned}
 f(X_1, X_2, X_3) &= bX_3 \vee bX_1 \bar{X}_2 \vee \bar{b}X_2 \bar{X}_3 \vee \bar{a}\bar{X}_1 \bar{X}_2 \vee \\
 &\bar{a}\bar{X}_2 X_3 \vee \bar{a}X_2 \bar{X}_3 \vee \bar{a}bX_1 X_2
 \end{aligned}
 \tag{16}$$

Here the underlying Boolean algebra $B_{16} = FB(a, b)$ is the free Boolean algebra generated by the two generators a and b . This algebra has 16 elements that are exactly the binary switching functions of a and b . These elements constitute a complemented distributive lattice in the form of a four dimensional hypercube. **Figure 3** shows the hypercube lattice of B_{16} and indicates partial ordering among its 16 elements. Notable among these 16 elements are the 0 and 1 elements and also the four elements $\bar{a}\bar{b}$, $\bar{a}b$, $a\bar{b}$ and ab which are the minterms of a

switching function of two variables, or the atoms of the atomic algebra B_{16} , or the dimensions of its hypercube representation.

Figure 4-7 demonstrate the derivation of CS(f) for the function f in (16) via VEKM folding. Initially, we represent f in **Fig. 4** by its natural map, which is a VEKM of map variables X_1, X_2 and X_3 and entries that are functions of the generators a and b . These entries are written in CS forms. The VEKM in **Fig. 4** is now folded (according to the rules of **Fig. 1**), first with respect to X_1 (**Fig. 5**), then with respect to X_2 (**Fig. 6**) and finally with respect to X_3 (**Fig. 7**), while retaining CS entries during each folding. **Figure 7** is simply an ANDing table (multiplication table) for the two entries in **Fig. 6**. Every absorbed term is encircled with an arrow pointing to the absorbing term (which happens to be on the same row or the same column). The remaining terms, which are *PIs* of f , are not circled and are stressed in bold. Their disjunction is CS(f) given by:

$$\begin{aligned}
 CS(f) &= bX_3 \vee \bar{a}\bar{X}_2 X_3 \vee aX_1 X_2 X_3 \vee \bar{a}X_2 \bar{X}_3 \vee \bar{b}X_2 \bar{X}_3 \\
 &\vee \bar{a}bX_1 X_2 \vee bX_1 \bar{X}_2 \vee \bar{a}\bar{X}_1 \bar{X}_3 \vee \bar{a}\bar{X}_1 \bar{X}_2 \vee \bar{a}b
 \end{aligned}
 \tag{17}$$

Now, we rearrange the CS formula (17) so as to express X_1 in terms of X_2 and X_3 and to express X_2 in terms of X_1 :

$$\begin{aligned}
 CS(f) &= (aX_2 X_3 \vee \bar{a}bX_2 \vee b\bar{X}_2) X_1 \vee (\bar{a}\bar{X}_3 \vee \bar{a}\bar{X}_2) \bar{X}_1 \\
 &\vee (\bar{a}\bar{X}_3 \vee \bar{b}\bar{X}_3) X_2 \vee (\bar{a}X_3) \bar{X}_2 \vee (b) X_3 \vee (0) \bar{X}_3 \vee \bar{a}b
 \end{aligned}
 \tag{18}$$

Now, some complementation is needed as follows:

$$\begin{aligned}
 P_1 &= aX_2 X_3 \vee \bar{a}bX_2 \vee b\bar{X}_2 \\
 \bar{P}_1 &= (\bar{a} \vee \bar{X}_2 \vee \bar{X}_3) (\bar{a} \vee b \vee \bar{X}_2) (\bar{b} \vee X_2) \\
 &= (\bar{a} \vee \bar{X}_2 \vee b\bar{X}_3) (\bar{b} \vee X_2) \\
 &= \bar{a}\bar{b} \vee \bar{a}X_2 \vee \bar{b}\bar{X}_2 \vee bX_2 \bar{X}_3 \\
 P_2 &= (\bar{a} \vee \bar{b}) \bar{X}_3 \\
 \bar{P}_2 &= ab \vee X_3
 \end{aligned}$$

Hence, the subsumptive solution is:

$$\begin{aligned}
 \bar{a}b &= 0 \\
 0 &\leq X_3 \leq \bar{b} \\
 \bar{a}X_3 &\leq X_2 \leq (ab \vee X_3) \\
 (\bar{a}\bar{X}_3 \vee \bar{a}\bar{X}_2) &\leq X_1 \leq (\bar{a}\bar{b} \vee \bar{a}X_2 \vee \bar{b}\bar{X}_2 \vee bX_2 \bar{X}_3)
 \end{aligned}
 \tag{19}$$

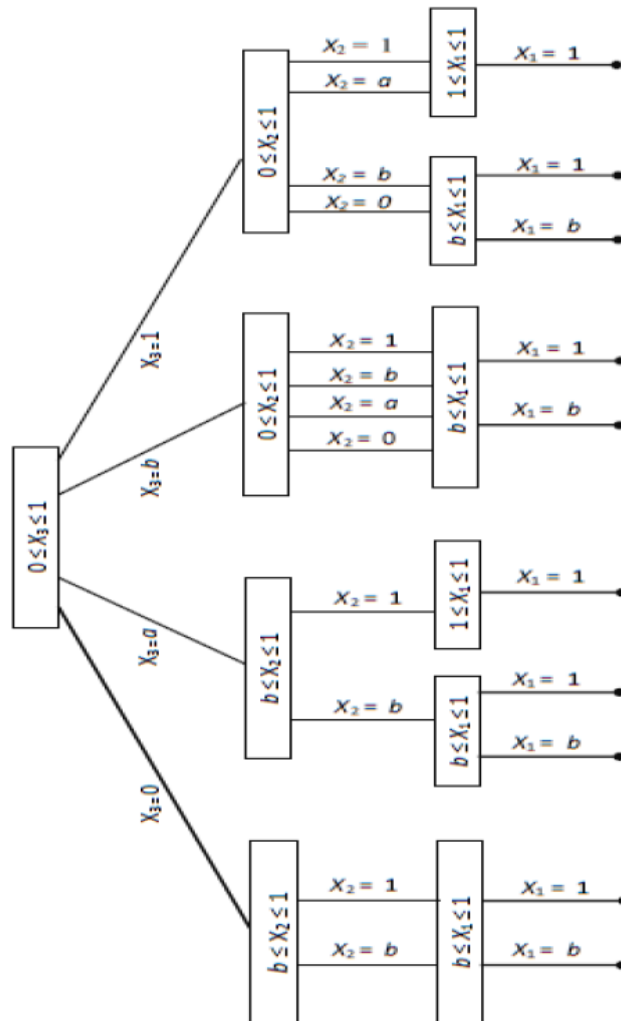


Fig. 2. Expansion tree (reduced to an acyclic graph) for obtaining all particular solutions of Example 1 from the general subsumptive solution (13). For clarity \bar{a} is written as b

The consistency condition ($\bar{a}b=0$) will force the lattice in **Fig. 3** to lose one atom or one dimension and hence to collapse to the three dimensional cube in **Fig. 8**. The subsumptive solution (19) can be used, if necessary, to develop all particular solutions of $f = 0$. **Figure 9** shows part of the expansion tree that can be used for this purpose. A complete listing of all 45 particular solutions (albeit with X_1 interchanged with X_3) is available in (Rushdi, 2004; Rushdi and Amashah, 2011).

5.3. Example 3

The function $f = (X_1, X_2, X_3): B_{16}^3 \rightarrow B_{16}$ which satisfies ($f = 0$) is given by:

$$f(X_1, X_2, X_3) = ab \vee a\bar{b}\bar{X}_1 \vee \bar{a}X_1X_3 \vee \bar{b}X_1X_2 \vee \bar{a}\bar{b}\bar{X}_1 \vee \bar{a}\bar{X}_1X_2 \vee \bar{b}\bar{X}_1X_3 \tag{20}$$

The complete sum of this function is:

$$CS(f) = ab \vee a\bar{X}_1 \vee bX_1 \vee \bar{a}\bar{b}X_2 \vee \bar{b}\bar{X}_1X_2 \vee \bar{a}\bar{X}_1X_2 \vee X_3 \tag{21}$$

This CS formula is now arranged to give:

$$CS(f) = (a \vee \bar{b}\bar{X}_2)X_1 \vee (b \vee \bar{a}X_2) \bar{X}_1 \vee (\bar{a}\bar{b})X_2 \vee (0)\bar{X}_2 \vee (1)X_3 \vee (0)\bar{X}_3 \vee ab \tag{22}$$

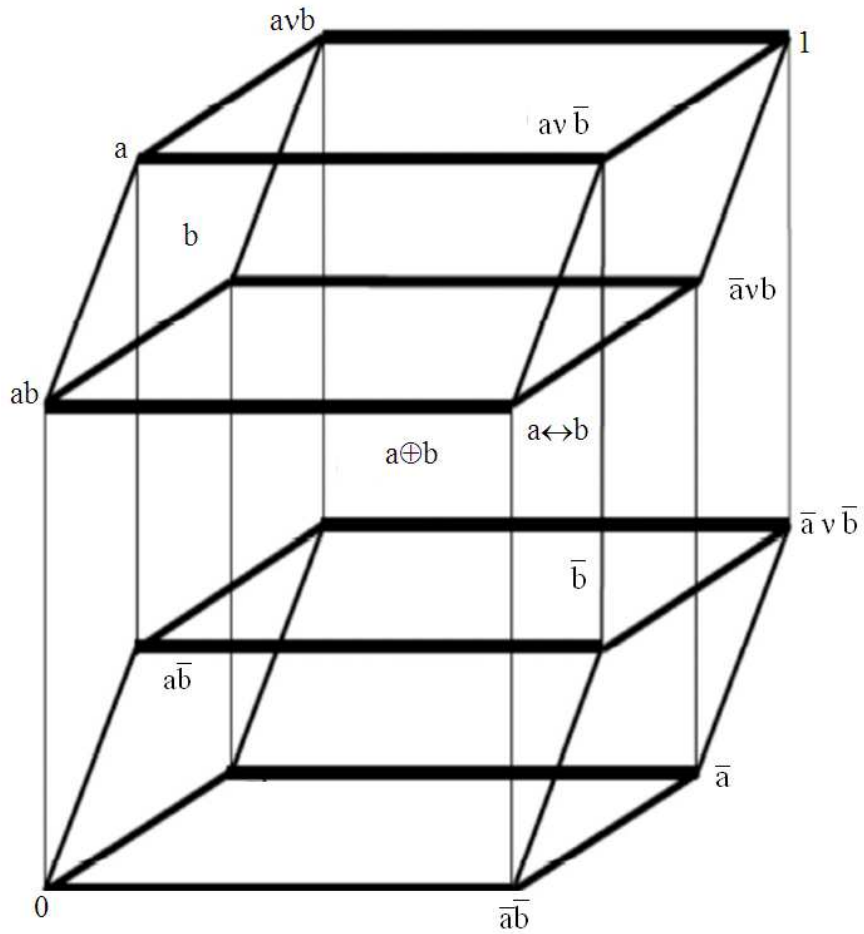


Fig. 3. A hypercube lattice indicating the partial ordering among the 16 elements of the atomic algebra B_{16} . Notable among these elements are the four atoms $\bar{a}\bar{b}$, $\bar{a}b$, $a\bar{b}$ and ab

		$\overbrace{\hspace{2cm}}^{x_1}$	
	\bar{a}	$\bar{a} \vee \bar{b}$	$\bar{a} \vee b$
	$\bar{a} \vee \bar{b}$	b	$a \vee b$
$\underbrace{\hspace{1cm}}_{x_3}$	$\bar{a} \vee b$	b	$a \vee b$
		$\underbrace{\hspace{2cm}}_{x_2}$	

$$f(\vec{x})$$

Fig. 4. A VEKM representation for $f(\vec{X})$ with CS entries

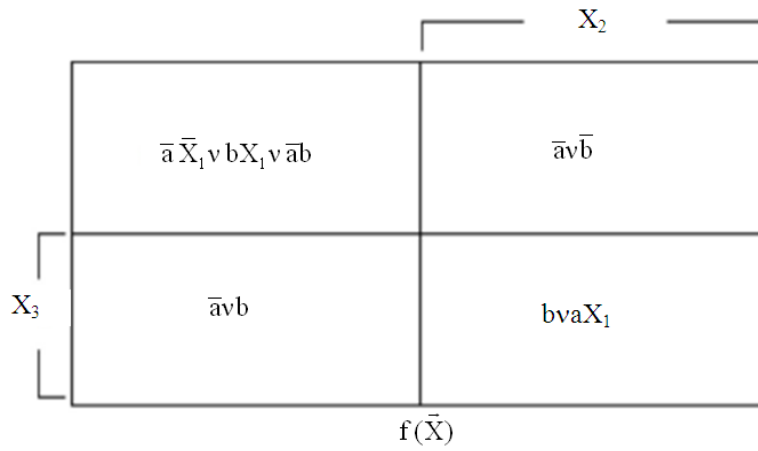


Fig. 5. The VEKM in Fig. 4 folded w.r.t. X_1 and still having CS entries

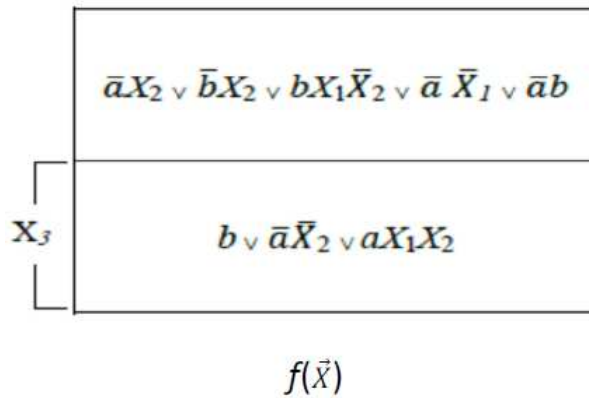


Fig. 6. The VEKM in Fig. 5 folded w.r.t. X_2 and still having CS entries

	\bar{X}_3	b	$\bar{a}\bar{X}_2$	aX_1X_2
X_3	—	bX_3	$\bar{a}\bar{X}_2X_3$	$aX_1X_2X_3$
aX_2	$aX_2\bar{X}_3$	$\bar{a}bX_2$	—	—
$\bar{b}X_2$	$\bar{b}X_2\bar{X}_3$	—	—	$a\bar{b}X_1X_2$
$bX_1\bar{X}_2$	$bX_1\bar{X}_2\bar{X}_3$	$bX_1\bar{X}_2$	$\bar{a}bX_1\bar{X}_2$	—
$\bar{a}\bar{X}_1$	$\bar{a}\bar{X}_1\bar{X}_3$	$\bar{a}b\bar{X}_1$	$\bar{a}\bar{X}_1\bar{X}_2$	—
$\bar{a}b$	$\bar{a}b\bar{X}_3$	$\bar{a}b$	$\bar{a}b\bar{X}_2$	—

Fig. 7. ANDing table for the two entries in Fig. 6, producing a VEKM of 0 map variable or an algebraic expression of f in CS form

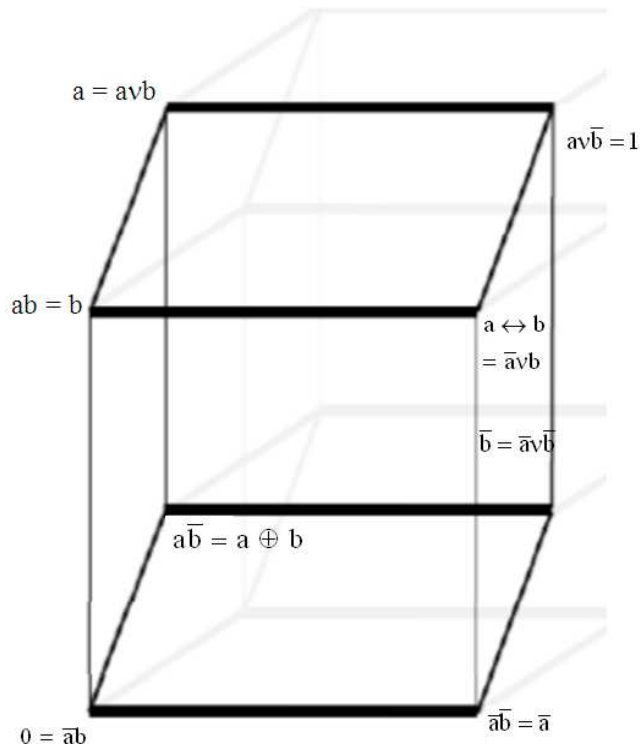


Fig. 8. The lattice in Fig. 3 when collapsed under the condition $\bar{a}b = 0$

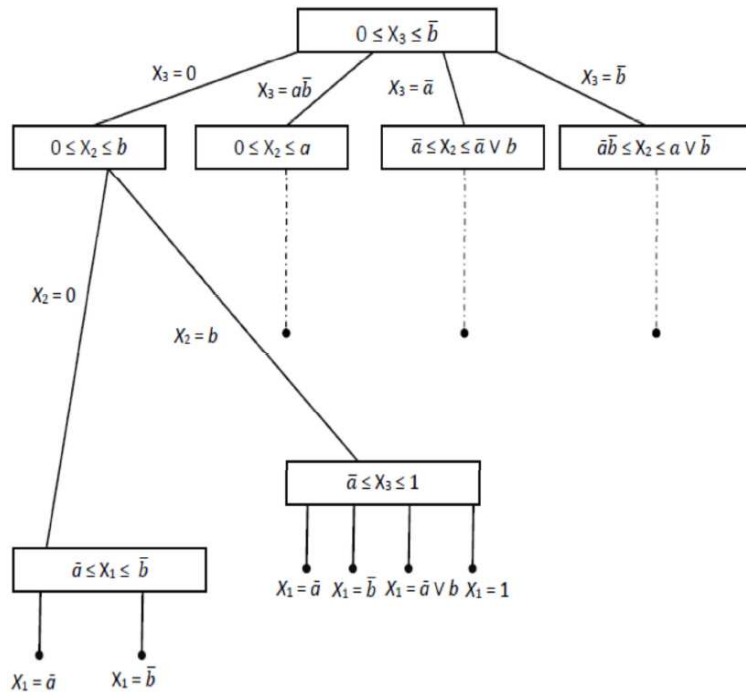


Fig. 9. Expansion tree for obtaining all the particular solutions of Example 2 from the general subsumptive solution (19)

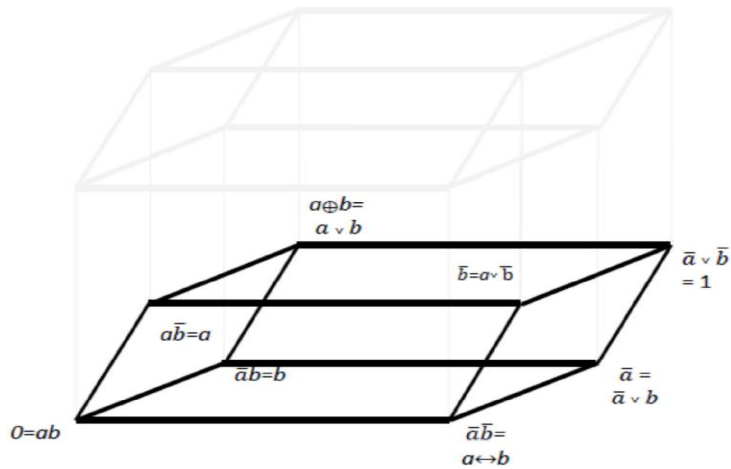


Fig. 10. The lattice in Fig. 3 when collapsed under the condition $ab=0$

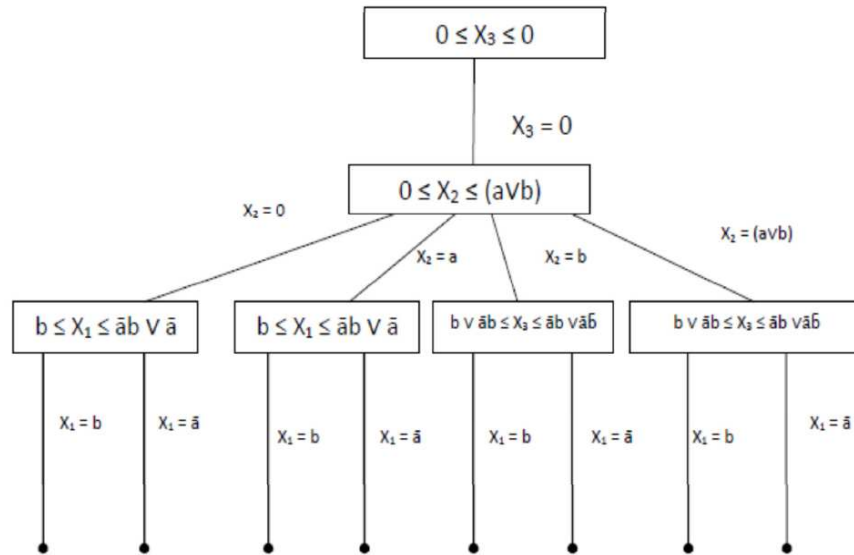


Fig. 11. Expansion tree for obtaining all the particular solutions of Example 2 from the general subsumptive solution (23)
The final subsumptive solution is:

$$\begin{aligned}
 &ab = 0 \\
 &0 \leq X_3 \leq 0 \\
 &0 \leq X_2 \leq (a \vee b) \\
 &(b \vee \bar{a}X_2) \leq X_1 \leq (\bar{a}b \vee \bar{a}\bar{X}_2)
 \end{aligned}
 \tag{23}$$

Figure 10 illustrates the acyclic-graph production of all 8 particular solutions from the general solution (23). Here, the consistency condition ($ab = 0$) made the underlying Boolean algebra collapse from the hypercube lattice of B_{16} in Fig. 3 to the cubic lattice of B_8 in Fig. 11.

5.4. Example 4

Consider the Boolean equation:

$$f(X) = \bar{b}X \vee \bar{a}\bar{X} \vee ac \vee b\bar{c} = 0 \tag{24}$$

where, $f(X) = B \rightarrow B$ and $B = FB(a, b, c)$ is a Boolean algebra of $2^{2 \cdot 3} = 256$ elements constituting all the switching (bivalent Boolean) functions of three arguments a, b and c . In the following, we update $f(X)$ gradually into CS form using Tison algorithm, by adding

consensi with respect to the biform variable X and the three biform generators a, b, c respectively. Of course, there will be some need herein for absorbing subsuming terms whenever such terms emerge:

$$\begin{aligned}
 f(X) &= (\bar{b}X \vee a\bar{X} \vee ac \vee b\bar{c}) \vee \bar{a}\bar{b} \\
 &= (\bar{b}X \vee a\bar{X} \vee ac \vee b\bar{c} \vee \bar{a}\bar{b}) \vee c\bar{X} \vee \bar{b}c \\
 &= (\bar{b}X \vee a\bar{X} \vee ac \vee b\bar{c} \vee \bar{a}\bar{b} \vee c\bar{X} \vee \bar{b}c) \vee c\bar{X} \vee \bar{a}\bar{c} \\
 &= (\bar{b}X \vee a\bar{X} \vee ac \vee b\bar{c} \vee \bar{a}\bar{b} \vee c\bar{X} \vee \bar{b}c \vee c\bar{X} \vee \bar{a}\bar{c}) \vee \\
 &ab \vee b\bar{X} \vee aX \vee (\bar{a}\bar{X} \vee \bar{b}X \vee \bar{a}\bar{b})
 \end{aligned} \tag{25}$$

The last line of equation (25) has three terms enclosed in parentheses which are absorbed in other terms. This equation represents $CS(f)$ when these terms are omitted and can be rearranged as:

$$CS(f) = (PX \vee Q\bar{X} \vee R) \tag{26a}$$

Where:

$$R = a \vee b \vee c \tag{26b}$$

$$Q = \bar{a} \vee b \vee c \tag{26c}$$

$$Q = ac \vee b\bar{c} \vee \bar{a}\bar{b} \vee \bar{b}c \vee \bar{a}\bar{c} \vee ab \tag{26d}$$

from which one obtains:

$$(\bar{a} \vee b \vee c) \leq X \leq \bar{a}\bar{b} \tag{27a}$$

Subject to the consistency condition:

$$(ac \vee b\bar{c} \vee \bar{a}\bar{b}) \vee \bar{b}c \vee \bar{a}\bar{c} \vee ab = 0 \tag{27b}$$

The terms in parentheses in (27b) are the only terms that would appear in consistency conditions by the eliminants method (Rudeanu, 1974; Brown, 1990), the don't-care method (Rushdi, 2001b; 2004) or the Rudeanu method (Rudeanu, 2003). Other terms in (27b) are generalized consensi of the earlier terms. The condition (27b) indicates that six out of the eight atoms of the underlying Boolean algebra are nullified. These are the atoms $\bar{a}\bar{b}\bar{c}, \bar{a}bc, a\bar{b}\bar{c}, abc, \bar{a}\bar{b}c$ and $a\bar{b}c$. Hence B retains the two (now complementary) atoms $\bar{a}\bar{b}c$ and $a\bar{b}c$ and hence it reduces into a 4-element Boolean algebra:

$$B = \{0, \alpha, \bar{\alpha}, 1\} \tag{28}$$

where, $\alpha = \bar{a}\bar{b}c$ and $\bar{\alpha} = a\bar{b}c$. The two bounds in (27a) are both equal to $\bar{\alpha}$ and hence X has a single particular solution $X = \bar{\alpha}$

6. DISCUSSION

The present method of *CS* subsumptive solution of Boolean equations utilizes a canonical representation that explicitly shows complete information about the pertinent Boolean function $f(\bar{X})$ in the most compact form. It is typically more efficient than the don't-care techniques in (Rushdi, 2001b; 2004), but while the *CS* solution obtains a rather compact solution that is not necessarily minimal, the don't-care techniques seek the most compact solution by using Boolean minimization methods.

The *CS* method is comparable in efficiency and compactness of solution to the Rudeanu method in (Rudeanu, 2003). The *CS* method might need slightly more effort than the conventional method based on constructing eliminants, but this extra effort pays off, since it results in a more compact solution and hence in easier generation of the tree (or acyclic graph) of particular solutions. The *CS* solution obtained herein has two minor advantages over other known methods of subsumptive solutions, namely, (a) it explicitly casts the consistency condition in *CS* form and hence provides immediate complete information about it and (b) it allows nesting the subsumptions in (2) according to any desirable permutation of the set of integers $\{1, 2, \dots, n\}$.

The *CS* method easily detects if a Boolean equation $f(\bar{X}) = 0$ is inconsistent, for then it produces $CS(f(\bar{X})) = 1$, which leads to the inconsistency $\{1 = 0\}$, or in other words, leads to a consistency condition $\{1 = 0\}$ which is not satisfiable. In this case, all atoms of B are nullified, B collapses to a single element and the solution set is empty.

7. CONCLUSION

A novel method for obtaining the general subsumptive solution of a general Boolean equation is introduced. The method is based on the derivation of the complete sum $CS(f)$ of the pertinent Boolean function. Prominent methods for such a derivation are briefly outlined and utilized in four demonstrative examples, covering Boolean algebras of various sizes.

Occasionally, the consistency conditions result in collapse of the underlying Boolean algebra into a smaller subalgebra. An expansion tree (typically reduced to an acyclic graph) is used to deduce a complete list of all particular solutions from the subsumptive solution. The present CS method is guaranteed to produce correct solutions, thanks to the fact that it fits into the frame of the most general subsumptive solution. Among competing subsumptive methods, the CS method is relatively good (albeit not necessarily the best) from both criteria of computational efficiency and compactness of solutions obtained. The CS method also enjoys the advantages of casting its consistency condition in a complete (but compact) form and of allowing the nesting of subsumptions in any possible permutation.

The CS method for solving Boolean equations can be classified as an application of the Modern Syllogistic Method (MSM) (Blake, 1937; Brown, 1990; Gregg, 1998; Rushdi and Al-Shehri, 2002; Rushdi and Baz, 2007; Rushdi and Ba-Rukab, 2007; 2008a; 2008b; 2009; 2014). In this perspective, the equation $f(\bar{X})=0$ can be viewed as a set of premises in a logic-deduction process, while the equation $CS(f(\bar{X}))$ is thought of as a set of consequents in this deduction.

An interesting topic for further research is whether a linear (Reed-Muller) representation of the pertinent Boolean function (see, e.g., Rushdi and Ghaleb, 2013; Rushdi and Alsogati, 2013) could provide a new alternative for solving the corresponding Boolean equation.

8. ACKNOWLEDGEMENT

This article was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah. The authors, therefore, acknowledge with thanks DSR technical and financial support.

9. REFERENCES

- Blake, A., 1937. Canonical expressions in boolean algebra. Ph.D. Dissertation, Department of Mathematics, University of Chicago.
- Brown, F.M. and S. Rudeanu, 1988. Prime implicants of dependency functions. *Anal. Univ. Bucu.*, 37: 16-11.
- Brown, F.M., 1990. *Boolean Reasoning: The Logic of Boolean Equations*. 1st Edn., Kluwer Academic Publishers, Boston, USA.
- Cutler, R.B., K. Kinoshita and S. Muroga, 1979. Exposition of Tison's method to derive all prime implicants and all irredundant disjunctive forms for a given switching function. Department of Computer Science, University of Illinois, Urbana, IL, USA.
- Gregg, J.R., 1998. *Ones and Zeros: Understanding Boolean Algebra, Digital Circuits and the Logic of Sets*. 1st Edn., IEEE Press, New York, USA.
- Kean, A. and G. Tsiknis, 1990. An incremental method for generating prime implicants/implicates. *J. Symbolic Computat.*, 9: 185-206. DOI: 10.1016/S0747-7171(08)80029-6
- Levchenkov, V.S., 2000a. Boolean equations with many unknowns. *Comput. Math. Model.*, 11: 143-153. DOI: 10.1007/BF02359181
- Levchenkov, V.S., 2000b. Solution of equations in Boolean algebra, *Comput. Math. Model.*, 11: 154-163. DOI: 10.1007/BF02359182
- Muroga, S., 1979. *Logic Design and Switching Theory*. 1st Edn., Wiley, New York, NY, USA.
- Reusch, B., 1975. Generation of prime implicants from subfunctions and a unifying approach to the covering problem. *IEEE Trans. Comput.*, C-24: 924-930. DOI: 10.1109/T-C.1975.224338
- Rudeanu, S., 1974. *Boolean Functions and Equations*. 1st Edn., North-Holland Publishing Company and American Elsevier, Amsterdam, the Netherlands.
- Rudeanu, S., 2001. *Lattice Functions and Equations*. 1st Edn., Springer Verlag, London, UK.
- Rudeanu, S., 2003. Algebraic methods versus map methods of solving Boolean equations. *Int. J. Comput. Mathem.*, 80: 815-817. DOI: 10.1080/0020716031000087159
- Rudeanu, S., 2010. Boolean sets and most general solutions of Boolean equations. *Inform. Sci.*, 180: 2440-2447. DOI: 10.1016/j.ins.2010.01.029
- Rushdi, A.M. and A.O. Baz, 2007. Computer-assisted resolution of engineering ethical dilemmas. *Proceedings of the 7th Saudi Engineering Conference, (SEC '07), Riyadh, Saudi Arabia*, pp: 409-418.
- Rushdi, A.M. and A.S. Al-Shehri, 2002. Logical reasoning and its supporting role in the service of security and justice. *J. Security Stud.*, 11: 115-153.
- Rushdi, A.M. and H.A. Al-Yahya, 2000. A Boolean minimization procedure using the variable-entered Karnaugh map and the generalized consensus concept. *Int. J. Elec.*, 87: 769-794. DOI: 10.1080/00207210050028724

- Rushdi, A.M. and H.A. Al-Yahya, 2001a. Derivation of the complete sum of a switching function with the aid of the variable-entered Karnaugh map. *J. King Saud Univ. Eng. Sci.*, 13: 239-269.
- Rushdi, A.M. and H.A. Al-Yahya, 2001b. Further improved variable-entered Karnaugh map procedures for obtaining the irredundant forms of an incompletely-specified switching function. *J. King Abdulaziz Univ. Eng. Sci.*, 13: 111-152.
- Rushdi, A.M. and Al-Yahya, 2002. Variable-entered Karnaugh map procedures for obtaining the irredundant disjunctive forms of a switching function from its complete sum. *J. King Saud Univ. Eng. Sci.*, 14: 13-27.
- Rushdi, A.M.A. and H.M. Albarakati, 2013a. The inverse problem for Boolean equations. *J. Comput. Sci.*, 8: 2098-2105. DOI: 10.3844/jcssp.2012.2098.2105
- Rushdi, A.M.A. and H.M. Albarakati, 2013b. Using variable-entered Karnaugh maps in determining dependent and independent sets of Boolean functions. *J. King Abdulaziz Univ. FCIT*, 2: 1-27.
- Rushdi, A.M.A. and A.A. Alsogati, 2013. On reduced scalar equations for synchronous Boolean networks. *J. Math. Stat.*, 9: 262-276. DOI: 10.3844/jmssp.2013.262.276
- Rushdi, A.M. and M.H. Amashah, 2010. Parametric general solutions of Boolean equations via variable-entered Karnaugh maps. *J. Qassim Univ. Eng. Compu. Sci.*, 3: 59-71.
- Rushdi, A.M. and M.H. Amashah, 2011. Using variable-entered karnaugh maps to produce compact parametric general solutions of Boolean equations. *Int. J. Comput. Math.*, 88: 3136-3149. DOI: 10.1080/00207160.2011.594505
- Rushdi, A.M.A. and M.H. Amashah, 2012. Purely-algebraic versus VEKM methods for solving big Boolean equations. *J. King Abdulaziz Univ. Eng. Sci.*, 23: 75-85.
- Rushdi, A.M. and O.M. Ba-Rukab, 2007. Some Engineering Applications of the modern syllogistic method. Proceedings of the 7th Saudi Engineering Conference, (SEC '07), Riyadh, Saudi Arabia, pp: 389-401.
- Rushdi, A.M. and O.M. Ba-Rukab, 2008a. Powerful features of the modern syllogistic method of propositional logic. *J. Math. Stat.*, 4: 186-193. DOI: 10.3844/jmssp.2008.186.193
- Rushdi, A.M. and O.M. Ba-Rukab, 2008b. The modern syllogistic method as a tool for engineering problem solving, *J. Qassim Univ. Eng. Comput. Sci.*, 1: 57-70.
- Rushdi, A.M. and O.M. Ba-Rukab, 2009. An exposition of the modern syllogistic method of propositional logic. *J. Umm Al-Qura Univ. Eng. Archit.*, 1: 17-49.
- Rushdi, A.M.A. and O.M. Ba-Rukab, 2014. Switching algebraic analysis of relational databases. *J. Mathem. Stat.*, 10: xxx-xxx. DOI: 10.3844/jmssp.2014.xxx.xxx
- Rushdi, A.M.A. and F.A.M. Ghaleb, 2013. On self-inverse binary matrices over the binary Galois field. *J. Math Stat.*, 9: 238-248. DOI: 10.3844/jmssp.2013.238.248
- Rushdi, A.M., 1987. Improved variable-entered Karnaugh map procedures. *Comput. Elec. Eng.*, 13: 41-52. DOI: 10.1016/0045-7906(87)90021-8
- Rushdi, A.M., 1996. Karnaugh Map, In: Hazewinkel, M. (Ed.), *Encyclopedia of Mathematics*. Kluwer Academic publishers, Boston, USA, 1: 327-328.
- Rushdi, A.M., 2001a. Prime-implicant extraction with the aid of the variable-entered Karnaugh map. *J. Sci., Med. Eng.*, 13: 53-74.
- Rushdi, A.M., 2001b. Using variable-entered Karnaugh maps to solve Boolean equations. *Int. J. Comput. Math.*, 78: 23-38.
- Rushdi, A.M., 2004. Efficient solution of Boolean equations using variable-entered Karnaugh maps. *J. King Abdulaziz Univ. Eng. Sci.*, 15: 105-121.
- Rushdi, A.M., 2012. A comparison of algebraic and map methods for solving general Boolean equations. *J. Qassim Univ. Eng. Comput. Sci.*, 4: 1-32.
- Thayse, A., 1978. Meet and join derivatives and their use in switching theory. *IEEE Trans. Comput.*, C-27: 713-720. DOI: 10.1109/TC.1978.1675178
- Tison, P., 1967. Generalization of consensus theory and application to the minimization of Boolean functions. *IEEE Trans. Elec. Comput.*, EC-16: 446-456. DOI: 10.1109/PGEC.1967.264648
- Tucker, J.H. and M.A., Tapia, 1992. Using Karnaugh maps to solve Boolean equations by successive elimination. Proceedings of the IEEE Southeastcon 92, Birmingham, AL, USA, pp: 589-592.