JKAU: Eng. Sci., Vol. **28** No. 1, pp: 3 - 18 (1438 A.H./ 2017 A.D.)

Doi: 10.4197/Eng. 28-1.1

Satisfiability in Big Boolean Algebras *via* Boolean-Equation Solving

Ali M. Ali Rushdi and **Waleed Ahmad**

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

arushdi@kau.edu.sa

Abstract. This paper studies Satisfiability (SAT) in finite atomic Boolean algebras larger than the two-valued one B_2, which are named big Boolean algebras. Unlike the formula $g(\boldsymbol{X})$ in the SAT problem over B_2, which is either *satisfiable* or *unsatisfiable*, this formula for the SAT problem over a big Boolean algebra could be *unconditionally satisfiable, conditionally satisfiable*, or *unsatisfiable* depending on the nature of the *consistency condition* of the Boolean equation $\{g(\boldsymbol{X}) = 1\}$, since this condition could be an *identity*, a *genuine equation*, or a *contradiction.* The paper handles this latter SAT problem by using a conventional method and a novel one for deriving *parametric general solutions*, and subsequently utilizing expansion trees for generating *all particular solutions* of the aforementioned Boolean equation. Each of these two methods could be cast in pure algebraic form, but becomes much easier to visualize and comprehend when presented *via* the natural map of a big Boolean algebra, which (for historical reasons) is called the variable-entered Karnaugh map (VEKM). In the classical method, the number of parameters used is *minimized* and *compact* solutions are obtained. However, the parameters belong to the *underlying big Boolean algebra.* By contrast, the novel method does not attempt to minimize the number of parameters used, as it uses *independent* parameters belonging to the *two-valued Boolean algebra B_2* for each asserted atom in the Boole-Shannon expansion of the formula $g(\boldsymbol{X})$. Though the method produces non-compact expressions, it is much quicker in generating particular solutions. The two methods are demonstrated *via* two detailed examples.

Keywords: Satisfiability, Big Boolean algebras, Boolean-equation solving, Parametric solutions, Particular solutions, Novel method.

1. Introduction

Propositional or (two-valued) Boolean Satisfiability (SAT) is the problem of deciding whether a propositional logic formula $g(\boldsymbol{X})$ can be satisfied (equated to 1) given suitable propositional value assignments to the variables $\boldsymbol{X}$ of the formula. The formula is satisfiable if the solution set of the Boolean equation $\{g(\boldsymbol{X}) = 1\}$ is non-empty, and the formula is unsatisfiable if that solution set is empty[1-3]. For n variables, there are 2^n possible truth assignments to be checked. In fact, the SAT problem is the first problem that has been proven to be NP-complete. This means that it is a highly intractable problem, and unless P = NP, all SAT algorithms (and all algorithms for any NP-complete problem) require worst-case exponential time. Since all NP-complete problems are mutually reducible to one another in linear time, a SAT solver can be used to solve other NP-complete problems at a modest extra cost. The literature abounds with sophisticated algorithms[4-15] that are designed to deal with large or gigantic SAT problems. Most of these algorithms were initiated with and influenced by the celebrated Davis-Putnam search strategy[4, 5].

When studying the SAT problem, it is typically assumed that the pertinent formula is given as a Conjunctive Normal Form (CNF), *i.e.*, as a formula consisting of a conjunction (ANDing) of clauses (alterms), each of which consists of a disjunction (ORing) of literals, where a literal is a variable in un-complemented form X_i or in a complemented form $\bar{X}_i$. A CNF is also known in digital-design circles as a product-of-sums (pos) expression[1, 2, 16, 17]. The dual to a CNF is the Disjunctive Normal Form (DNF) which is a formula consisting of a disjunction (ORing) of products (terms), each of which consisting of a conjunction (ANDing) of literals. A DNF is also known as a sum-of-products (sop) expression[1, 2, 16, 17]. A special case of an sop expression is a disjoint (orthogonal) sop expression, which is one in which the ANDing of any two products is 0.

The formula $(X_1 \vee X_2 \vee X_3) \wedge (\bar{X}_1 \vee \bar{X}_2 \vee X_3)$ can be cited as a simple example of a satisfiable Formula. It has, in fact, several satisfying truth assignments, namely when $\{X_3 = 1$ with X_1 and X_2 being unspecified or don't cares$\}$ and when $\{X_3 = 0$ with X_1 being the complement of $X_2\}$. These are the six primitive assignments $\{X_1 = 0,\ X_2 = 0,\ X_3 = 1\}$, $\{X_1 = 0,\ X_2 = 1,\ X_3 = 1\}$, $\{X_1 = 1,\ X_2 = 0, X_3 = 1\}$, $\{\ X_1 = 1, X_2 = 1, X_3 = 1\}$, $\{X_1 = 0, X_2 = 1, X_3 = 0\}$ and $\{X_1 = 1, X_2 = 0, X_3 = 0\}$. It is easy to visualize that the 8-cell Karnaugh map for the aforementioned formula is 0-entered for the cell $\{X_1 = 0, X_2 = 0, X_3 = 0\}$ representing the clause or Maxterm $(X_1 \vee X_2 \vee X_3)$, and also for the cell $\{X_1 = 1, X_2 = 1, X_3 = 0\}$ representing the clause or Maxterm $(\bar{X}_1 \vee \bar{X}_2 \vee X_3)$, and hence this map is 1-entered for the remaining six cells. A Simple Example of an unsatisfiable formula is the formula $(X_1 \wedge \bar{X}_1)$. Likewise, the formula $(X_1 \vee X_1 \vee X_1) \wedge (\bar{X}_1 \vee \bar{X}_1 \vee \bar{X}_1)$ is unsatisfiable. Any formula containing all the Maxterms of a function is identically equal to 0, and hence is unsatisfiable. An example of such a formula is:

$$(\bar{X}_1 \vee \bar{X}_2) \wedge (\bar{X}_1 \vee X_2) \wedge (X_1 \vee \bar{X}_2) \wedge (X_1 \vee X_2)$$

If one or more Maxterms are dropped of this formula, it becomes satisfiable. For example, If we drop the clause $(X_1 \vee X_2)$, we obtain

$$(\bar{X}_1 \vee \bar{X}_2) \wedge (\bar{X}_1 \vee X_2) \wedge (X_1 \vee \bar{X}_2),$$

which is satisfiable by $\{X_1 = 0, X_2 = 0\}$.

There are many extensions of SAT which are problems that either use the same algorithmic techniques as used in SAT, or use SAT as a core engine. These include Satisfiability Modulo Theories (SMT), maximum satisfiability (MaxSAT), minimum satisfiability (MinSAT), model counting (SAT degree), and Quantified Boolean Formulas (QBF)[18]. Applications of SAT include many hard combinatorial problems such as problems that arise in formal verification, artificial intelligence, operations research, biology, cryptology, data mining, machine learning, mathematics, model-checking of finite-state systems, design debugging, inference in bioinformatics, knowledge-compilation, software model checking, software testing package, management in software distributions, checking of pedigree consistency, test-pattern generation in digital systems, design debugging and diagnosis, identification of functional dependencies in Boolean functions, technology-mapping in logic synthesis, and circuit delay computation[18].

While most algorithms, extensions and applications of the Boolean satisfiability problem are in the domain of two-valued crisp logic, there have been a few publications dealing with the satisfiability problem in multi-valued logic[19] and fuzzy logic[20, 21]. To the best of our knowledge, no work on the satisfiability problem over big Boolean algebras has, so far, appeared in the open literature.

The aim of this paper is to handle the SAT problem for the formula $g(\boldsymbol{X})$ over an arbitrary or big Boolean algebra[22, 23] by using a conventional method and a novel one for deriving parametric general solutions of the Boolean equation $\{g(\boldsymbol{X}) = 1\}$, and subsequently utilizing expansion trees for generating all particular solutions of the aforementioned Boolean equation. These methods could be cast in pure algebraic form[24-27], but become much easier to visualize and comprehend when presented *via* the natural map of a big Boolean algebra, which (for historical reasons) is called the variable-entered Karnaugh map (VEKM)[23, 28-37]. In the classical method, the number of parameters used is minimized and compact solutions are obtained[22, 23, 30, 38-41]. However, the parameters belong to the underlying big Boolean algebra. By contrast, the novel method does not attempt to minimize the number of parameters used, as it uses independent parameters belonging to B_2 for each asserted atom in the Boole-Shannon expansion of the formula $g(\boldsymbol{X})$. Though the method produces non-compact expressions, it is much quicker in generating particular solutions.

The organization of the rest of this paper is as follows: Section 2 lists certain useful features of big Boolean algebras, and clarifies important aspects of the SAT problem over them. Section 3 reviews classical parametric general solutions of big Boolean equations. This section makes the paper self-contained as it sets the stage for introducing the novel method for parametric general solutions in Section 4. Both sections explain how parametric solutions can be used to generate all particular solutions, which are all the satisfying instances of the SAT problem. Much work might be saved if all that is required is a single particular solution, *i.e.*, a single satisfying instance of the SAT problem. The two methods might be presented in a purely algebraic fashion by using the Boole-Shannon expansion to handle the discriminants of $g(\boldsymbol{X})$[22]. An equivalent (albeit more insightful) scheme is adopted herein by using the natural map of $g(\boldsymbol{X})$[23]. The two methods are demonstrated *via* two examples that illustrate the two prominent possible outcomes when the formula g(**X**) is unconditionally satisfiable or conditionally satisfiable. Section 5 concludes the paper.

2. Satisfiability over Big Boolean Algebras

A *Boolean algebra* is a quintuple $B = (\boldsymbol{B}, \vee, \wedge, 0, 1)$ in which $\boldsymbol{B}$ is a set, called the carrier; $\vee$ and $\wedge$ are binary operations on $\boldsymbol{B}$, and the zero (0) and unit (1) elements are *distinct* members of $\boldsymbol{B}$, with certain postulates on commutativity, distributivity, identities and complementation being satisfied. The following facts about a Boolean algebra can be deduced[22-26, 42-45]:

1. Every element X of $\boldsymbol{B}$ has a *unique complement* $\overline{X}$.

2. There is a *partial-order* or *inclusion* ($\leq$) relation on B that is *reflexive*, *anti-symmetric*, and *transitive*.

3. A Boolean algebra B enjoys many useful properties such as *associativity*, *idempotency*, *absorption*, *involution*, *consensus* and *duality*.

4. A Boolean algebra B is a *complemented distributive lattice* whose 0 and 1 values are *distinct.*

5. A nonzero element Z of $\boldsymbol{B}$ is said to be an *atom* of $\boldsymbol{B}$ if and only if for every $X \in \boldsymbol{B}$, the condition $X \leq Z$ implies that $X = Z$ or $X = 0$.

6. Every *finite* Boolean algebra B is *atomic*, *i.e.* for every nonzero element $X \in \boldsymbol{B}$, there is some atom Z such that $Z \leq X$. This viewpoint rejects the case $\{0 = 1\}$ as a contradiction, and ignores the possibility of an *atomless* algebra $\boldsymbol{B}_1$ in which $\{0 = 1\}$ is accepted!

7. Examples of Boolean algebras include the algebra of classes (subsets of a set), the algebra of propositional functions, the arithmetic Boolean algebra, the switching or two-element Boolean algebra, as well as big Boolean algebras.

8. Boolean algebras with the same number of elements are *isomorphic*.

9. Every finite Boolean algebra B has 2^M elements, where M is the cardinality of (number of elements in) the set of atoms of $\boldsymbol{B}$. Following Brown[22], we distinguish Boolean algebras larger than the two-valued one (the switching algebra $\boldsymbol{B}_2$, M =1) by naming them big Boolean algebras.

10. A Boolean function f: $\boldsymbol{B}^n \rightarrow \boldsymbol{B}$, where $\boldsymbol{B}$ is a carrier of 2^M elements, is uniquely determined by a truth table or a Karnaugh map partially representing f for the *restricted domain* $\{0, 1\}^n$ which is a *strict* subset of the complete domain $\boldsymbol{B}^n$.

11. The elements of $\boldsymbol{B}$ are named in terms of a *minimum* number of *abstract* variables or *generators* $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m)$, with the elements of $\boldsymbol{B}$ taken as the elements of the *free Boolean algebra* $FB(\mathbf{Y}) = FB(Y_1, Y_2, \ldots, Y_m)$ which is isomorphic to the Boolean algebra of switching functions of m variables, and possesses $M = 2^m$ atoms and 2^M elements. The smallest big Boolean algebra B_4 has a single generator a, two atoms $\bar{a}$ and a, and 4 partially-ordered elements ($0 \leq \{\bar{a}, a\} \leq 1$) that are the 4 switching functions of one variable. Figure 1 uses a 4-dimensional hypercube lattice to visualize the big Boolean algebra B_{16} which has two generators a and b, four atoms $\bar{a}\bar{b}, \bar{a}b, a\bar{b}$ and ab, and 16 partially-ordered elements that are the 16 switching functions of 2 variables. Figure 2 demonstrates a cubic lattice that represents the big Boolean algebra B_8 which still has two generators a and b, but only three atoms $\bar{a}\bar{b}$, $\bar{a}b$, and $a\bar{b}$, and 8 partially-ordered elements. Note that B_8 in Fig. 2 can be obtained from B_{16} in Fig. 1 by nullifying its atom ab. More information on the lattice constructions in Fig. 1 and 2 is available in Rushdi and Amashah[23, 30].

Unlike the SAT problem over B_2, where a formula $g(\boldsymbol{X})$ is either *satisfiable* if the solution set of the Boolean equation $\{g(\boldsymbol{X}) = 1\}$ is non-empty, or *unsatisfiable* if that solution set is empty, the SAT problem over a big Boolean algebra has three possibilities:

- The formula $g(\boldsymbol{X})$ is *unconditionally satisfiable* if the *consistency condition* of the Boolean equation $\{g(\boldsymbol{X}) = 1\}$ is an *identity*. In this case, $g(\boldsymbol{X})$ is satisfied by every particular solution of the Boolean equation, and the underlying Boolean algebra remains intact.

- The formula $g(\boldsymbol{X})$ is *conditionally satisfiable* if the *consistency condition* of the Boolean equation $\{g(\boldsymbol{X}) = 1\}$ is a genuine consistent *equation,* whose solutions nullify some atoms of the underlying Boolean algebra, thereby leading to its collapse to a smaller algebra. In this case $g(\boldsymbol{X})$ is satisfied (subject to the consistency condition) by every particular solution of the Boolean equation over the collapsed Boolean algebra.

- The formula $g(\boldsymbol{X})$ is *unsatisfiable* if the *consistency condition* of the Boolean equation $\{g(\boldsymbol{X}) = 1\}$ is a *contradiction* $\{1 = 0\}$. In this case, the solution set of the Boolean equation $\{g(\boldsymbol{X}) = 1\}$ is empty, and all atoms of the underlying Boolean algebra are nullified, thereby leading to its collapse to a single point. This happens when the function $g(\boldsymbol{X})$ is identically equal to 0, *e.g.*, when $g(\boldsymbol{X})$ is actually a conjunction of all Maxterms (even in disguise). A simple example of such a function in the big Boolean algebra B_{65536} with elements in $FB(\mathrm{a, b, c, d})$ is:

$$g(X_1, X_2) = (a \vee \bar{X}_1 \vee \bar{X}_2) \wedge (\bar{a} \vee \bar{X}_1 \vee \bar{X}_2) \wedge (b \vee X_1 \vee \bar{X}_2) \wedge (\bar{b} \vee X_1 \vee \bar{X}_2) \wedge (c \vee \bar{X}_1 \vee X_2) \wedge (\bar{c} \vee \bar{X}_1 \vee X_2) \wedge (d \vee X_1 \vee X_2) \wedge (\bar{d} \vee X_1 \vee X_2). \quad (1)$$

The SAT problem over a big Boolean algebra is handled herein by solving a Boolean equation. There are three main types of Boolean-equation solutions, which can be identified as subsumptive general solutions[22, 25-29, 31-34], parametric general solutions[22, 23, 25, 26, 30, 31, 38-41] and particular solutions. In a subsumptive general solution, each of the variables is expressed as an interval based on successive conjunctive or disjunctive eliminants of the underlying function $g(\boldsymbol{X})$. In a classical parametric general solution, each of the variables is expressed *via* arbitrary parameters, *i.e.*, *via* freely chosen elements of the underlying Boolean algebra. A particular solution is an assignment from the underlying Boolean algebra to every pertinent variable that makes the Boolean equation an identity. We are going to present a novel parametric general solution which uses independent parameters belonging to B_2 for each asserted atom in the Boole-Shannon expansion of the formula $g(\boldsymbol{X})$. This novel method will be seen to be a very convenient way of listing all particular solutions.

3. Classical Parametric General Solutions of Big Boolean Equations

In the classical method of parametric solution of big Boolean equations, the minimum number of parameters is sought. Though this method is well established[22, 23, 25, 26, 30, 31, 38-41], we review it herein to make the paper self-contained, and to set the stage for the next section. Brown[22] proved that n parameters are sufficient to construct a parametric general solution of an n-variable Boolean equation $g(\boldsymbol{X}) = 1$, where $g(\boldsymbol{X})$: $\boldsymbol{B}^n \to \boldsymbol{B}$. He proposed a procedure for constructing such a solution using the fewest possible parameters, $p_1, p_2, \ldots, p_k$, which are elements of $\boldsymbol{B}$, where $k \leq n$. Rushdi and Amashah[23, 30] adapted this procedure of Brown into a VEKM procedure as follows:

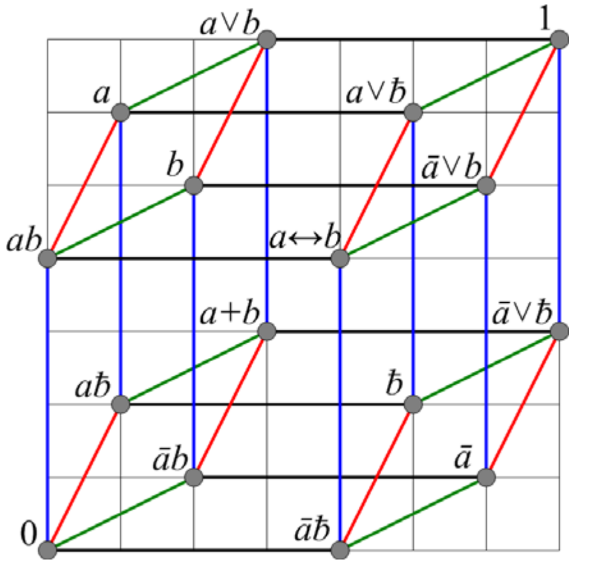


Fig. 1. A hypercube lattice indicating the partial ordering among the 16 elements of $\boldsymbol{B}_{16}$.

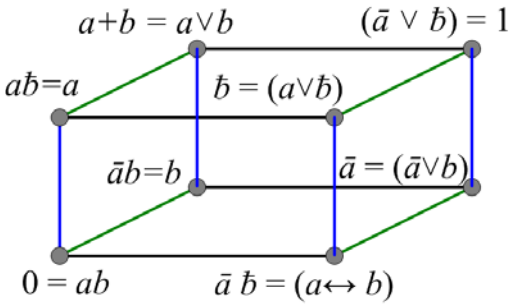

$a+b = a\vee b$
$(\bar{a} \vee \bar{b}) = 1$
$a\bar{b}=a$
$\bar{b} = (a\vee\bar{b})$
$\bar{a}b=b$
$\bar{a} = (\bar{a}\vee b)$
$0 = ab$
$\bar{a}\,\bar{b} = (a\leftrightarrow b)$

Fig. 2. The lattice in Fig. 1, collapsed under the condition ab = 0 so as to represent B8.

(a) Construct a VEKM representing $g(\boldsymbol{X})$. Such a construction is achieved *via* a Boole-Shannon tree expansion[46, 47]. If the original Boolean equation is in the dual form $f(\boldsymbol{X})$= 0, then construct a VEKM for $f(\boldsymbol{X})$, and complement it cell-wise[28, 48] to obtain a VEKM for $\bar{f}(\boldsymbol{X}) = g(\boldsymbol{X})$.

(b) Expand the entries of the VEKM of $g(\boldsymbol{X})$ as ORing of appropriate atoms of the Boolean carrier $\boldsymbol{B}$, or equivalently as a minterm expansion of the free Boolean algebra representing **B**. If atom i ($1 \leq i \leq$ I) of $\boldsymbol{B}$ appears A_i times in the cells of this VEKM, then the number of particular solutions N is given by

$$N = \prod_{i=1}^{I} A_i. \qquad (2)$$

(c) If certain atoms of $\boldsymbol{B}$ do not appear at all in any cell of the VEKM for $g(\boldsymbol{X})$, then these atoms must be *forbidden* or *nullified*. Such a nullification constitutes *a consistency condition* for the given Boolean equation.

(d) Construct a VEKM for an associated auxiliary function $G(\boldsymbol{X},\boldsymbol{p})$. This VEKM is deduced from that of $g(\boldsymbol{X})$ through the following modifications:

(d1) Each appearance of an entered atom in the VEKM of $g(\boldsymbol{X})$ is ANDed with a certain element of a set of orthonormal tags of minimal size. An orthonormal set consists of a set of terms T_i, i = 1, 2, …, m, which are both exhaustive ($T_1 \vee T_2 \vee \ldots \vee T_m = 1$) and mutually exclusive ($T_i \wedge T_j = 0$ for $1 \leq i < j \leq m$). The terms T_i are products of parameters p_r, $1 \leq r \leq k$, in uncomplemented or complemented form. The set of tags for different atoms share the same parameters. The number of parameters k is minimized and is given by the ceiling of the base-2 logarithm of the maximum number of appearances A_i of an atom,

$$k = \lceil \max_i (\log_2 A_i) \rceil = \lceil \log_2 (\max_i A_i) \rceil \qquad (3)$$

(d2) Each nullified atom is entered as a don’t care in all the VEKM cells of the associated auxiliary function $G(\boldsymbol{X},\boldsymbol{p})$.

(e) The parametric solution is

X_i= The sum (ORing) of the entries in the 2^{n-1} cells constituting ($X_i = 1$), *i.e.* in half of the VEKM in which X_i is asserted), (i = 1, 2, …, n), namely

$$X_i = \vee_{\{\boldsymbol{Y}\in\{0,1\}^{\boldsymbol{n}} | Y_i=1\}} \mathrm{G}(\mathbf{Y},\mathbf{p}). \qquad (4)$$

(f) Apply an appropriate VEKM minimization procedure[46-49] to recast (4) in a minimal form.

Example 1

We apply the aforementioned technique to the function $g_1(\boldsymbol{X})\colon \boldsymbol{B}_4^3 \rightarrow \boldsymbol{B_4}$ given by the natural map in Fig. 3. This natural map is typically called a Variable-Entered Karnaugh Map (VEKM)[23, 28-34, 46-49] with map variable $\boldsymbol{X} = \{X_1, X_2, X_3\}$ and with an entered ‘variable’ that is not really a variable but is the generator a of the underlying Boolean algebra $\boldsymbol{B}_4 = \{0, 1, \bar{a}, a\}$. The map can be read [38, 39] to express $g_1(\boldsymbol{X})$ in pos (CNF) form as:

$$g_1(X) = (a \vee X_1 \vee X_2) \wedge (a \vee X_3) \wedge (\bar{X}_1 \vee \bar{X}_2 \vee X_3), \qquad (5)$$

or in sop (DNF) form as

$$g_1(\boldsymbol{X}) = a\bar{X}_1 \vee a\bar{X}_2 \vee X_1X_3 \vee X_2X_3 \quad (6)$$

The map in Fig. 3 indicates that $g_1(\boldsymbol{X})$ is unconditionally satisfied. In fact, the map displays three satisfying assignments $[X_1\, X_2\, X_3]$ = [0 1 1], [1 1 1], and [1 0 1], for which $g_1(\boldsymbol{X})$ = 1. Other possible satisfying assignments can be obtained if one constructs the complete Function Table of $g_1(\boldsymbol{X})\colon \boldsymbol{B}_4^3 \rightarrow \boldsymbol{B_4}$, which has a larger domain of $\boldsymbol{B}_4^3 = \{0, 1, \bar{a}, a\}^3$ consisting of 4^3= 64 cells. This larger domain includes the 8-cell domain $\{0, 1\}^3$ of the map in Fig. 3. Construction of this function table constitutes a brute force (exhaustive and time-consuming) method for obtaining not only the solution set of the equation $\{g_1(\boldsymbol{X}) = 1\}$ but also the solution set of the inequation $\{g_1(\boldsymbol{X}) \neq 1)\}$, which is the union of the solution sets of the equations $\{g_1(\boldsymbol{X}) = 0\}$, $\{g_1(\boldsymbol{X}) = a\}$, and $\{g_1(\boldsymbol{X}) = \bar{a}\}$. To give the reader a glimpse of the nature of a function table, we include the table for $g_1(\boldsymbol{X})$ in Fig. 4. The availability of such a table should not defeat the purpose of our present equation-solving strategy, simply because the construction of such a table is usually prohibitively tedious for any non-toy problem

Steps of our proposed solution are illustrated by Fig. 5 and 6. In Fig. 5, the entries of the map for the function $g_1(\boldsymbol{X})$ in Fig. 3 are expanded in terms of the two atoms $\{a\,,\bar{a}\,\}$ of B_4, which happen to be the minterms of *FB(a)*. In Fig. 6, the 7-element set of the three-parameter orthonormal tags ($p_1\overline{p_2}p_3$, $p_1\overline{p_2}\,\overline{p_3}$, $p_1p_2p_3$, $p_1p_2\overline{p_3}$, $\overline{p_1}\,p_2p_3$, $\overline{p_1}p_2\overline{p_3}$, $\overline{p_1}\,\overline{p_2}$) is used for atom a and the 3-element set of the two-parameter orthonormal tags ($p_2\overline{p_3}$, $\overline{p_2}$, p_2p_3) is used for atom ā. Note that the two atoms share the two parameters p_2 and p_3, and the number of particular solutions is 7*3 = 21. The final parametric solution is given by[31]

$$X_1 = (\overline{p_1}\,\overline{p_2} \vee \overline{p_1}p_3 \vee a\,p_1p_2\overline{p_3}) \vee \bar{a}\,(\overline{p_2} \vee p_3) \tag{7a}$$

$$X_2 = (\overline{p_1}\,\overline{p_2} \vee \overline{p_1}\,\overline{p_3} \vee \overline{p_2}\,\overline{p_3}) \vee \bar{a}\,(\overline{p_2} \vee \overline{p_3}) \tag{7b}$$

$$X_3 = \overline{p_1} \vee p_2p_3 \vee \bar{a}. \tag{7c}$$

together with the consistency condition

$$0 = 0. \tag{7d}$$

Note that the parameters $\{p_1 \quad p_2 \quad p_3\}$ belong to the underlying big Boolean algebra $\boldsymbol{B}_4 = \{0, 1, \bar{a}, a\}$. Particular solutions can be deduced from the parametric solution (7) *via* a three-level quaternary expansion tree[31], which traverses a search space of 4^3 = 64 points (more than double the cardinality of the solution set), and is further complicated by the fact that expansion at a child node is dependent on earlier expansions at parent nodes. Though the parametric solution (7) has the advantage of being of minimum parameters, it is discredited for the tedious effort it needs to produce particular solutions.

Example 2

Now consider the function $g_2(\boldsymbol{X})$: $\boldsymbol{B}_{16}^3 \to \boldsymbol{B_{16}}$ given by its natural map in Fig. 6, where $\boldsymbol{B_{16}} = FB(a,b)$ is displayed as a complemented distributive lattice in Fig. 1. The map can be read[46, 47] to express $g_2(\boldsymbol{X})$ in pos (CNF) form as:

$$g_2(\boldsymbol{X}) = (\bar{X}_1) \wedge (\bar{b} \vee X_3) \wedge (\bar{a} \vee \overline{X_3}) \wedge (a \vee \overline{X_2} \vee X_3) \wedge (b \vee \bar{X}_2 \vee \overline{X_3}) \tag{8}$$

or in sop (DNF) form as:

$$g_2(\boldsymbol{X}) = a\bar{b}\bar{X}_1\overline{X_3} \vee \bar{a}b\bar{X}_1X_3 \vee \bar{b}\bar{X}_1\overline{X_2}\,\overline{X_3} \vee \bar{a}\bar{X}_1\overline{X_2}\,X_3 \tag{9}$$

The map in Fig. 7 indicates that $g_2(\boldsymbol{X})$ is not unsatisfiable (since it is not identically 0). Figure 7 can be used also to deduce that $g_2(\boldsymbol{X})$ is not unconditionally satisfiable, either. In fact, Fig. 7 tells us that only three of the four atoms of the underlying B_{16} algebra, namely $\bar{a}b, a\bar{b}$ *and* $\bar{a}\bar{b}$, appear in the entries of the map of Fig. 8. Their appearances are 2, 2, and 2, respectively, which means that the number of particular solutions is 2*2*2 = 8. The fourth atom ab does not appear anywhere in Fig. 7. Its nullification constitutes the consistency condition.

$$ab = 0, \tag{10a}$$

which causes the underlying Boolean algebra B_{16} to collapse to B_8 shown in Fig. 2. In this case, a single parameter p is needed so as to produce a 2-element orthonormal set $\{\bar{p}, p\}$. The 2 elements of this set are used to tag the 2 appearances of each of the asserted atoms ($\bar{a}b, a\bar{b}$ *and* $\bar{a}\bar{b}$) as shown in Fig. 8. In addition, the nullified atom ab is entered don't-care in every cell of the map in Fig. 8. Now, this map represents the auxiliary function $G_2(X_1, X_2, X_3; p)$ associated with $g_2(X_1, X_2, X_3)$. Finally the solution of $\{g_2(\boldsymbol{X}) = 1\}$ is given by:

$$X_1 = 0, \tag{10b}$$

$$X_2 = b\bar{p} \vee ap, \tag{10c}$$

$$X_3 = b \vee \bar{a}p. \tag{10d}$$

together with the consistency condition (10a) given earlier. The single parameter p in (10 b) – (10 d) belongs to the collapsed Boolean algebra B_8 in Fig. 2.

X_1

	a	a	0	a
X_3	a	1	1	1

X_2

$g_1(\boldsymbol{X})$

Fig. 3. A natural map representing the Boolean function $g_1(\boldsymbol{X})$ in Example 1.

X3/X1	0				1				a				$\bar{a}$			
0	a	a	a	a	a	0	0	a	a	0	0	a	a	a	a	a
1	a	1	a	1	1	1	1	1	a	1	a	1	1	1	1	1
a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
$\bar{a}$	a	1	a	1	1	$\bar{a}$	$\bar{a}$	1	a	$\bar{a}$	0	1	1	1	1	1
X2	0	1	a	$\bar{a}$	0	1	a	$\bar{a}$	0	1	a	$\bar{a}$	0	1	a	$\bar{a}$

Fig. 4. The function table of 64 cells for the function $g_1(\boldsymbol{X})$ of Example 1. The 8 cells constituting the Karnaugh map of Fig. 3 are shaded.

X_1

	a	a	0	a
X_3	a	$a \vee \bar{a}$	$a \vee \bar{a}$	$a \vee \bar{a}$

X_2

$g_1(\boldsymbol{X})$

Fig. 5. Entries of the map for the function $g_1(\boldsymbol{X})$ in Fig. 3 expanded in terms of atoms of $\boldsymbol{B}_4$ or minterms of FB(a).

X_1

	$a\,p_1\overline{p_2}p_3$	$a\,p_1\overline{p_2}\,\overline{p_3}$	0	$a\,p_1p_2\overline{p_3}$
X_3	$p_1p_2p_3a$	$a\,\overline{p_1}p_2\overline{p_3}$ $\vee\,\bar{a}p_2\overline{p_3}$	$a\,\overline{p_1}\,\overline{p_2}$ $\vee\,\bar{a}\overline{p_2}$	$a\,\overline{p_1}\,p_2p_3\,\vee$ $\bar{a}\,p_2p_3$

X_2

$G_1(X_1, X_2, X_3; p_1, p_2, p_3)$

Fig. 6. Each appearance of an entered atom in Fig. 5 ANDed with an element of a set of orthonormal tags. The parameters used are shared by the two atoms.

X_3

	$\bar{b}$	$a\bar{b}$	$\bar{a}b$	$\bar{a}$
X_1	0	0	0	0

X_2

$g_2(\boldsymbol{X})$

Fig. 7. A natural map representation of the Boolean function $g_2(\boldsymbol{X})$ in (8) or (9).

X_3

	$\bar{a}\bar{b}\bar{p} \vee a\bar{b}\bar{p} \vee$ $d(ab)$	$a\bar{b}p \vee$ $d(ab)$	$\bar{a}b\bar{p} \vee$ $d(ab)$	$\bar{a}\bar{b}p \vee \bar{a}b\,p \vee$ $d(ab)$
X_1	$d(ab)$	$d(ab)$	$d(ab)$	$d(ab)$

X_2

$G_2(X_1, X_2, X_3; p)$

Fig. 8. Each appearance of an entered atom in Fig. 6 is ANDed with a certain element of the set of orthonormal tags { $p, \bar{p}$ } while the atom ab that appears nowhere in Fig. 6 is entered don't care.

4. Novel Parametric General Solutions of Big Boolean Equations

The procedure in Sec. 3 is modified herein by disallowing any sharing of parameters among atoms, *i.e.* using an independent parameter for each atom. This modification is implied in the shortcut[22, 23, 30] used to obtain specific instances of particular solutions. It results in a dramatic increase in the number of parameters needed from k given by (3) to K given by:

$$K = \sum_i^I \lceil \log_2 A_i \rceil. \qquad (11)$$

However, instead of demanding parameters that exhaust all elements of the underlying Boolean algebra, each element of the new set of independent parameters is now required to span only the two values 0 and 1, *i.e.*, the elements of $\boldsymbol{B_2}$.

Example 1 (Revisited)

When we apply our novel method to $g_1(\boldsymbol{X})$ given by (5) or (6) and represented by either Fig. 3 or Fig. 5, we obtain the associated function $G_1{}'(X_1, X_2, X_3; p_1, p_2, p_3, p_4, p_5)$ in Fig. 9, which is similar to the corresponding function $G_1(X_1, X_2, X_3; p_1, p_2, p_3)$ in Fig. 6, with the sole exception that while the tags for the atom a are kept intact as functions of $p_1, p_2\ and\ p_3$, the tags for the other atom $\bar{a}$ retain their form but become functions of new parameters $p_4\ and\ p_5$ replacing $p_2\ and\ p_3$. Correspondingly, the solution of $\{\ g_1(\boldsymbol{X}) = 1\}$ becomes:

$$X_1 = a\ (\overline{p_1}\ \overline{p_2} \vee \overline{p_1} p_3 \vee p_1 p_2 \overline{p_3}) \vee \bar{a}\ (\overline{p_4} \vee p_5) \qquad (12a)$$

$$X_2 = a\ (\overline{p_1}\ \overline{p_2} \vee \overline{p_1}\ \overline{p_3} \vee \overline{p_2}\ \overline{p_3}) \vee \bar{a}\ (\overline{p_4} \vee \overline{p_5}) (12b)$$

$$X_3 = \text{a}\ (\overline{p_1} \vee p_2 p_3) \vee \bar{a}. \qquad (12c)$$

together with the consistency condition:

$$0 = 0. \qquad (12d)$$

Here each of the five parameters $p_1, p_2, p_3, p_4\ and\ p_5$ now belongs to $\boldsymbol{B}_2$ rather than $\boldsymbol{B}_4$. Equations (12a) - (12c) can now be written in matrix form as:

$$[X_1 \quad X_2 \quad X_3] = (a \wedge \mathbf{Co}(a)) \vee (\bar{a} \wedge \mathbf{Co}(\bar{a})) \qquad (13a)$$

where the vector $\boldsymbol{X}^T$ is a disjunction of two *total* contributions $(a \wedge\ \mathbf{Co}(a))$ and $(\bar{a}\ \wedge\ \mathbf{Co}(\bar{a}))$. Here, the contributions $\mathbf{Co}(a)$ and $\mathbf{Co}(\bar{a})$ are given by:

$$\mathbf{Co}(a) = [\overline{p_1}\ \overline{p_2} \vee \overline{p_1} p_3 \vee p_1 p_2 \overline{p_3} \quad \overline{p_1}\ \overline{p_2} \vee \overline{p_1}\ \overline{p_3} \vee \overline{p_2}\ \overline{p_3} \quad \overline{p_1} \vee p_2 p_3] \qquad (13b)$$

$$\mathbf{Co}(\bar{a}) = [\overline{p_4} \vee p_5 \quad \overline{p_4} \vee \overline{p_5} \quad 1\ ] \qquad (13c)$$

This new solution is definitely more cumbersome than the earlier one in (7), but it has the distinctive advantage of allowing a much quicker way to list all particular solutions. Instead of a three-level quaternary expansion tree, we now need a five-level binary tree (which means a reduction by half of the search space from 4^3 = 64 points to 2^5 = 32 points). However, we can *even do better than that* by avoiding the use of an expansion tree altogether and exactly targeting the 21 individual particular solutions. We note that $\mathbf{Co}(a)$ is a function of $p_1, p_2\ and\ p_3$ only, while $\mathbf{Co}(\bar{a})$ is a function of $p_4\ and\ p_5$ only. We can draw a Karnaugh map of 3 variables (and hence 8 cells) to represent $\mathbf{Co}(a)$, and another map of two variables (and hence 4 cells) to represent $\mathbf{Co}(\bar{a})$. However our original tag sets consist of only 7 and 3 elements, respectively. So we need map-like structures to represent these orthonormal tags. Simply, we need a 7-variable map of variables $p_1, p_2\ and\ p_3$ with cells $\overline{p_1}\ \overline{p_2}\ \overline{p_3}$ and $\overline{p_1}\ \overline{p_2}\ p_3$ combined as $\overline{p_1}\ \overline{p_2}$ and also a 2- variable map of variables $p_4\ and\ p_5$ with cells $\overline{p_4}\ \overline{p_5}$ and $\overline{p_4} p_5$ combined as $\overline{p_4}$. These map-like structures are now entered by specific values of $(a \wedge \mathbf{Co}(a))$ and $(\bar{a} \wedge \mathbf{Co}(\bar{a}))$ *via* (13b) and (13c). According to (13a), a particular solution is the disjunction of an arbitrarily-chosen entry in Fig. 10(a) with an arbitrarily-selected entry in Fig. 10(b). For example, we can use the entry $[a \quad a \quad a]$ in the leftmost cell of Fig. 10(a) ORed elementwise with the entry $[\bar{a} \quad \bar{a} \quad \bar{a}]$ in the leftmost cell in Fig. 10(b) to produce a specific particular solution

$$[a \quad a \quad a] \vee [\bar{a} \quad \bar{a} \quad \bar{a}] =$$
$$[a \vee \bar{a} \quad a \vee \bar{a} \quad a \vee \bar{a}] = [1 \quad 1 \quad 1] \qquad (14)$$

Since any of the 7 cells of Fig. 10(a) can go with any of the 3 cells of Fig. 10(b), it is clear that we can produce 7*3 = 21 particular solutions, as expected. These 21 particular solutions are listed as groups of 3 (in a map-like structure similar to that of Fig. 10(a)) in Fig. 11. Substituting of any of these solutions into (5) or (6) produces a value of 1 for $g_1(\boldsymbol{X})$. All of these particular solutions involve either a or $\bar{a}$, except the three solutions $[1 \quad 1 \quad 1]$, $[0 \quad 1 \quad 1]$ and $[1 \quad 0 \quad 1]$ which were earlier detected on the Karnaugh map of Fig. 3.

In passing, we note that $g_1(\boldsymbol{X})$ is partially symmetric in X_1 and X_2, *i.e.*, $g_1(X_1, X_2, X_3) = g_1(X_2, X_1, X_3)$[50]. Therefore, if $[\alpha, \beta, \gamma]$ is a particular solution of $\{g_1(\boldsymbol{X}) = 1\}$ then $[\beta, \alpha, \gamma]$ is also a particular solution of it. This observation can be verified for the set of particular solutions in Fig. 11.

Example 2 (Revisited)

We now apply our novel method to Example 2. Figure 12 replicates Fig. 8, with the single parameter p (belonging to B_{16} collapsed to B_8) replaced by three independent parameter, p_1, p_2 *and* p_3 (belonging to B_2) used in the tags of atoms $a\bar{b}$, $\bar{a}b$ *and* $\bar{a}\bar{b}$, respectively, while the map auxiliary function $G_2(X_1, X_2, X_3; p)$ is replaced by $G_2{}'(X_1, X_2, X_3; p_1, p_2, p_3)$. Correspondingly, the solution of $\{g_2(\boldsymbol{X}) = 1\}$ becomes

$$X_1 = 0, \qquad (15a)$$

$$X_2 = b\overline{p_2} \vee ap_1, \qquad (15b)$$

$$X_3 = b \vee \bar{a}p_3 = \bar{a}(b \vee p_3), \qquad (15c)$$

$$ab = 0. \qquad (15d)$$

Figure 13 lists all 8 particular solutions for $\{g_2(\boldsymbol{X}) = 1\}$ obtained by assigning independent binary values to the 3 parameters p_1, p_2 *and* p_3. Each of these 8 solutions involves some form (complemented or non-complemented) of the generator a or the generator b. That is why none of them was detected by the Karnaugh map of Fig. 6. Note that if any of the 8 particular solutions is substituted into (8) or (9) it does not produce $\{g_2(\boldsymbol{X}) = 1\}$ directly, but it produces $\{g_2(\boldsymbol{X}) = a\bar{b} \vee \bar{a}b \vee \bar{a}\bar{b}\}$ which reduces to $g_2(\boldsymbol{X}) = 1$ with the aid of the consistency condition (15d).

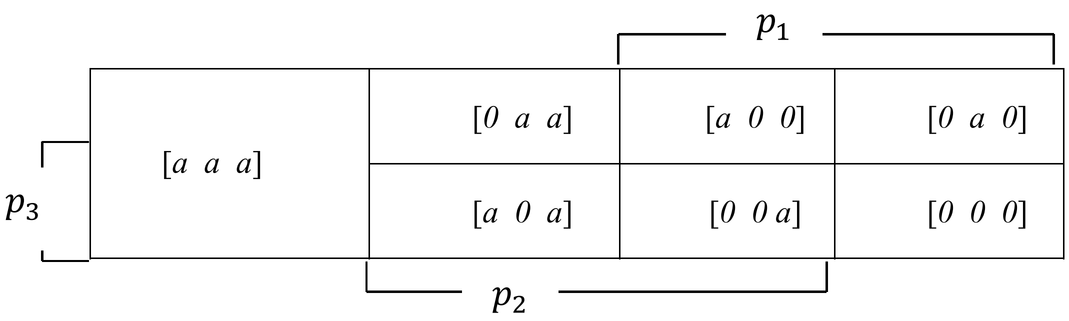
X1

	$p_1\overline{p_2}p_3a$	$p_1\overline{p_2}\,\overline{p_3}a$	0	$p_1p_2\overline{p_3}a$
X3	$p_1p_2p_3a$	$\overline{p_1}p_2\overline{p_3}a$ $\vee\ \bar{a}p_4\overline{p_5}$	$\overline{p_1}\,\overline{p_2}a$ $\vee\ \overline{\bar{a}p_4}$	$\overline{p_1}\,p_2p_3 \vee a$ $\bar{a}\,p_4p_5$

X2

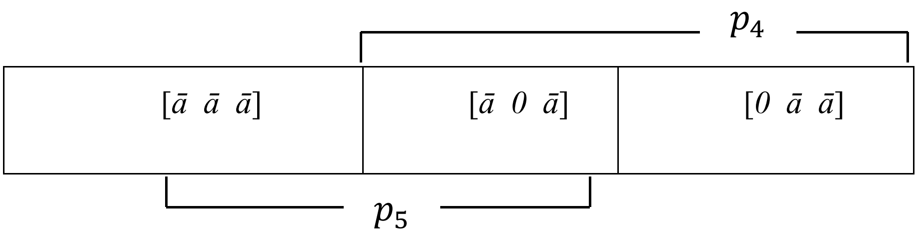
$$G'(X_1, X_2, X_3; p_1, p_2, p_3, p_4, p_5)$$

Fig. 9. Each appearance of an entered atom in Fig. 5 ANDed with a certain element of a set of orthonormal tags. The parameters used for different atoms are independent.



$[a\ \ a\ \ a]$	$[0\ \ a\ \ a]$	$[a\ \ 0\ \ 0]$	$[0\ \ a\ \ 0]$
	$[a\ \ 0\ \ a]$	$[0\ \ 0\ a]$	$[0\ \ 0\ \ 0]$

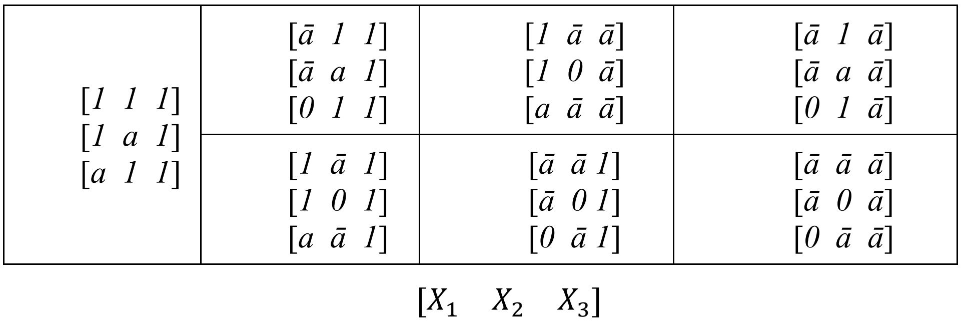
a. Total contribution $a\mathbf{Co}(a)$ of atom $\boldsymbol{a}$ to a particular solution $[X_1\quad X_2\quad X_3]$.



$[\bar{a}\ \ \bar{a}\ \ \bar{a}]$	$[\bar{a}\ \ 0\ \ \bar{a}]$	$[0\ \ \bar{a}\ \ \bar{a}]$

b. Total contribution $\bar{a}\,\mathbf{Co}(\bar{a})$ of atom $\bar{a}$ to a particular solution $[X_1\quad X_2\quad X_3]$.

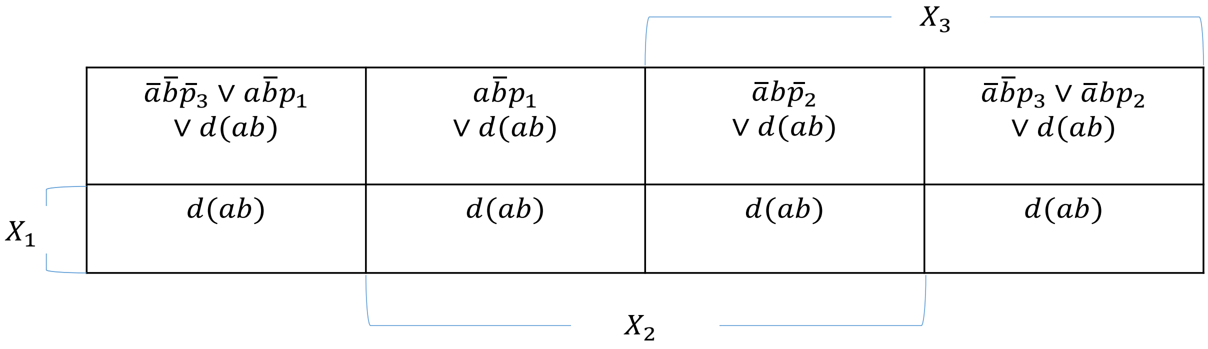
Fig. 10. Total contributions of atoms $\boldsymbol{a}$ and $\bar{\boldsymbol{a}}$ to a particular solution $[X_1\quad X_2\quad X_3]$ shown as entries in map-like structures that represent the original orthonormal tags.



$[1\ \ 1\ \ 1]$ $[1\ \ a\ \ 1]$ $[a\ \ 1\ \ 1]$	$[\bar{a}\ \ 1\ \ 1]$ $[\bar{a}\ \ a\ \ 1]$ $[0\ \ 1\ \ 1]$	$[1\ \ \bar{a}\ \ \bar{a}]$ $[1\ \ 0\ \ \bar{a}]$ $[a\ \ \bar{a}\ \ \bar{a}]$	$[\bar{a}\ \ 1\ \ \bar{a}]$ $[\bar{a}\ \ a\ \ \bar{a}]$ $[0\ \ 1\ \ \bar{a}]$
	$[1\ \ \bar{a}\ \ 1]$ $[1\ \ 0\ \ 1]$ $[a\ \ \bar{a}\ \ 1]$	$[\bar{a}\ \ \bar{a}\ 1]$ $[\bar{a}\ \ 0\ 1]$ $[0\ \ \bar{a}\ 1]$	$[\bar{a}\ \ \bar{a}\ \ \bar{a}]$ $[\bar{a}\ \ 0\ \ \bar{a}]$ $[0\ \ \bar{a}\ \ \bar{a}]$

$[X_1\quad X_2\quad X_3]$

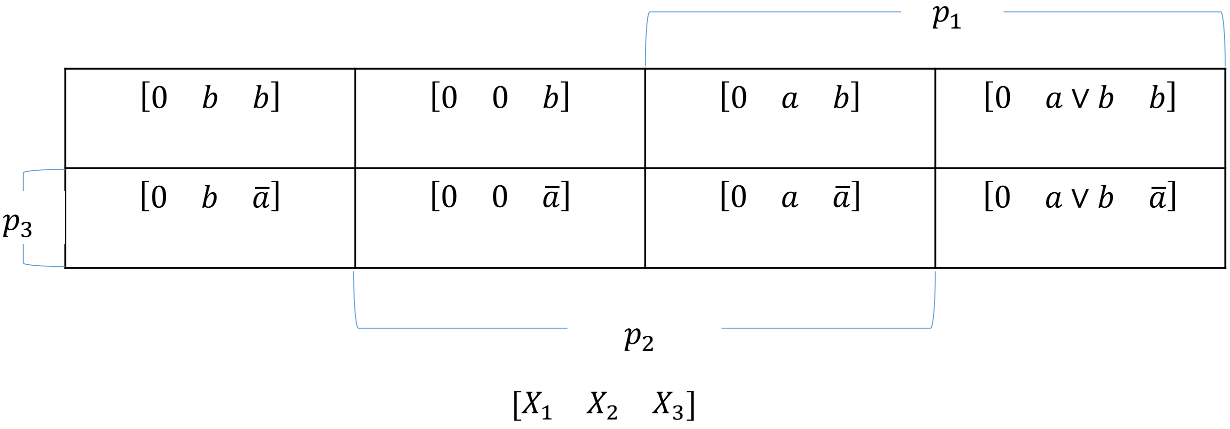
Fig. 11. A listing of all 21 particular solutions of $\{g_1(\boldsymbol{X}) = 1\}$ in Example 1.



$\bar{a}\bar{b}\bar{p}_3 \vee a\bar{b}p_1 \vee d(ab)$	$a\bar{b}p_1 \vee d(ab)$	$\bar{a}b\bar{p}_2 \vee d(ab)$	$\bar{a}\bar{b}p_3 \vee \bar{a}bp_2 \vee d(ab)$
$d(ab)$	$d(ab)$	$d(ab)$	$d(ab)$

$G(X_1, X_2, X_3; p_1, p_2, p_3)$

Fig. 12. The auxiliary function for Example 2 with tags of independent parameters belonging to $\mathbf{B}_2$.



$[0 \quad b \quad b]$	$[0 \quad 0 \quad b]$	$[0 \quad a \quad b]$	$[0 \quad a \vee b \quad b]$
$[0 \quad b \quad \bar{a}]$	$[0 \quad 0 \quad \bar{a}]$	$[0 \quad a \quad \bar{a}]$	$[0 \quad a \vee b \quad \bar{a}]$

Fig. 13. A listing of all eight particular solutions of $\{g_2(\boldsymbol{X}) = \mathbf{1}\}$ subjected to the consistency condition $\{\boldsymbol{ab} = \mathbf{0}\}$.

5. Conclusion

This paper has two important contributions. The paper's first major contribution is to propose that the famous problem of Boolean satisfiability (SAT) be extended from the two-valued Boolean domain to cover big Boolean algebras. The new type of satisfiability may therefore be conveniently labeled as BigSAT. The paper's second major contribution is to find all possible solutions, if any, of BigSAT by constructing parametric general solutions of associated Boolean equations. The conventional method for constructing such solutions was reviewed and then superceded by a novel method that can immediately exhibit all particular solutions. The paper is, therefore, setting the stage for solving BigSAT *via* advanced strategies similar to those of the good old Davis-Putnam procedure [4] and its many successors.

This paper sets the stage for future work that is more application oriented. We have already proposed a novel cryptosystem that is based on the utilization of big Boolean algebras[36]. The basic idea is to dramatically extend the search space needed in SAT-based cryptography. The adversary will not only be obliged to traverse a search space (that can be arbitrarily huge), but might end up with several candidate answers, all of which are wrong except one. Another sequel of this paper entails new methods of digital circuit design utilizing equation solving over big Boolean algebras

Acknowledgement

The authors are greatly indebted to Dr. Ahmad Ali Rushdi, University of Texas, Austin, Texas, USA, for the significant help he kindly and professionally offered in the preparation of this manuscript.

References

[1] **Hooker, J.,** *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction,* John Wiley & Sons, New York, NY, USA (2000).

[2] **Chandru, V.** and **Hooker, J.N.,** *Optimization Methods for Logical Inference,* John Wiley & Sons, New York, NY, USA (1999).

[3] **Biere, A., Heule, M. J. H., van Maaren, H.** and **Walsh, T.** (Editors), *Handbook of Satisfiability*, Vol. **185**. Frontiers in Artificial Intelligence and Applications. IOS Press (2009).

[4] **Davis, M.,** and **Putnam, H.,** A computing procedure for quantification theory, *Journal of the Association of Computing Machinery (JACM)*, **7** (3): 201–215 (1960).

[5] **Davis, M., Logemann, G.** and **Loveland, D.,** A machine program for theorem-proving. *Communications of the ACM*, **5**(7): 394-397 (1962).

[6] **Bryant, R. E.,** Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers,* **100** (8): 677-691 (1986).

[7] **Selman, B., Levesque, H. J.** and **Mitchell, D. G.,** A new method for solving hard satisfiability problems. *Proceedings of the AAAI*, **92:** 440-446 (1992).

[8] **Marques-Silva, J. P.,** and **Sakallah, K. A.,** GRASP—a new search algorithm for satisfiability, *Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design*, pp. 220-227 (1997).

[9] **Bayardo, Jr., R. J.** and **Schrag, R.,** Using CSP look-back techniques to solve real-world SAT instances, *Proceedings of the AAAI/IAAI,* pp. 203-208 (1997).

[10] **Moskewicz, M. W., Madigan, C. F., Zhao, Y., Zhang, L.** and **Malik, S.,** Chaff: Engineering an efficient SAT solver. *Proceedings of the 38th ACM Annual Design Automation Conference*, pp. 530-535 (2001).

[11] **Goldberg, E.** and **Novikov, Y.,** A fast and robust SAT-solver. *Discrete Applied Mathematics*, **155** (12): 1549-1561 (2002).

[12] **Eén, N.** and **Sörensson, N.,** An extensible SAT-solver. In: *Theory and Applications of Satisfiability Testing,* pp. 502-518, Springer, Berlin-Heidelberg, Germany (2004).

[13] **Mahajan, Y. S., Fu, Z.** and **Malik, S.,** Zchaff2004,: An efficient sat solver. In: *Theory and Applications of Satisfiability Testing*, pp. 360-375, Springer, Berlin-Heidelberg (2005).

[14] **De Moura, L.** and **Bjørner, N.,** Z3: An efficient SMT solver. In: *Tools and Algorithms for the Construction and Analysis of Systems*, pp. 337-340. Springer, Berlin-Heidelberg, Germany (2008).

[15] **Järvisalo, M., Le Berre, D., Roussel, O.** and **Simon, L.,** The international SAT solver competitions, *AI Magazine,* **33** (1): 89-92 (2012).

[16] **Muroga, S.,** *Logic Design and Switching Theory,* Wiley, New York, NY, USA (1979).

[17] **Rushdi, A. M. A.** and **Ahmad. W.,** Finding all solutions of the Boolean satisfiability problem, if any, *via* Boolean-equation solving, *Journal of King Abdulaziz University: Engineering Sciences,* **27** (1): 19-34 (2016).

[18] **Marques-Silva, J**., Practical applications of Boolean satisfiability. *9th IEEE International Workshop on Discrete Event Systems (WODES 2008)*, pp. 74-80 (2008).

[19] **Feldman, A., Pietersma**, **J.,** and **van Gemund**, **A. J. C.,** A multi-valued SAT-based algorithm for faster model-based diagnosis. In: C. A. Gonzales, T. Escobert, and B. Pulido, (Editors), Proc. DX-06, pp. 93–100 (2006).

[20] **Navara, M**., Satisfiability in fuzzy logics, *Neural Network World*, **10**(5): 845-858 (2000).

[21] **Rushdi, M. A., Rushdi, A. M., Zarouan, M.** and **Ahmad, W.,** Generalization of the satisfiability problem to intuitionistic fuzzy logic with realistic tautology, *Kuwait Journal of Science*, **45** (2018).

[22] **Brown, F. M.**, *Boolean Reasoning: The Logic of Boolean Equations*, Kluwer Academic Publishers, Boston, MA USA (1990).

[23] **Rushdi, A. M.** and **Amashah**, **M. H.,** Using variable–entered Karnaugh maps to produce compact parametric general solutions of Boolean equations, *International Journal of Computer Mathematics*, **88** (15): 3136-3149 (2011).

[24] **Hammer, P. L.** and **Rudeanu**, **S.,** *Boolean Methods in Operations Research and Related Areas*, Springer Verlag, Berlin, Germany (1968).

[25] **Rudeanu, S.**, *Boolean Functions and Equations*, North-Holland Publishing Company & American Elsevier, Amsterdam, the Netherlands (1974).

[26] **Rudeanu, S.**, *Lattice Functions and Equations*, Springer, London UK (2001).

[27] **Rudeanu, S.,** Algebraic methods versus map methods of solving Boolean equations, *International Journal of Computer Mathematics*, **80**(7): 815-817 (2003).

[28] **Rushdi, A. M.**, Using variable-entered Karnaugh maps to solve Boolean equations, *International Journal of Computer Mathematics*, **78** (1): 23-38 (2001).

[29] **Rushdi, A. M.**, Efficient solution of Boolean equations using variable-entered Karnaugh maps, *Journal of King Abdulaziz University: Engineering Sciences*, **15** (2): 21-29 (2004).

[30] **Rushdi, A. M.** and **Amashah**, **M. H.,** Parametric general solutions of Boolean equations *via* variable-entered Karnaugh maps, *Journal of Qassim University: Engineering and Computer Sciences,* **3** (1): 59-71 (2010).

[31] **Rushdi, A. M.**, A comparison of algebraic and map methods for solving general Boolean equations, *Journal of Qassim University: Engineering and Computer Sciences*, **4**(2): 1-32 (2012).

[32] **Rushdi, A. M.** and **M. H. Amashah**, Purely-algebraic versus VEKM methods for solving big Boolean equations, *Journal of King Abdulaziz University: Engineering Sciences,* **23**(2): 75-85, (2012).

[33] **Rushdi**, **A. M. A.,** and **H. M. Albarakati**, Prominent classes of the most general subsumptive solutions of Boolean equations, *Information Sciences*, **281**: 53-65 (2014).

[34] **Rushdi, A. M.,** and **Albarakati, H. M.,** Construction of general subsumptive solutions of Boolean equations *via* complete-sum derivation, *Journal of Mathematics and Statist*ics, **10** (2): 155-168 (2014).

[35] **Rushdi, A. M. A.** and **Al-Qwasmi, M. A**., Formal derivation of a particular input of a single AND (OR) gate in terms of its output and other inputs, *Journal of King Abdulaziz University: Engineering Sciences,* **26**

(2): 51-62 (2015).

[36] **Rushdi, A. M. A.** and **Ahmad. W.,** A novel method for compact listing of all particular solutions of a system of Boolean equations, *British Journal of Mathematics & Computer Science*, **22**(6): 1-18 (2017).

[37] **Rushdi, A. M. A.** and **Ahmad. W.,** A new cryptographic scheme utilizing the difficulty of big Boolean satisfiability, *Journal of Mathematical, Engineering and Management Sciences* (*IJMEMS*), **3**(1): 47-61(2017).

[38] **Brown, F.M.,** Reduced solutions of Boolean equations, *IEEE Transactions on Computers*, **C-19**: 976-981 (1970).

[39] **Deschamps, J. P.,** Parametric solutions of Boolean equations, *Discrete Mathematics*, **3**(4): 333-342, (1972).

[40] **Brown, F.M.**, Segmental solutions of Boolean equations, *Discrete Applied Mathematics*, **4**: 87-96 (1982).

[41] **Banković, D.**, Some remarks on number of parameters of the solutions of Boolean equations, *Discrete Mathematics*, **79**(3): 229-234 (1990).

[42] **Elliott, M.,** *Theory and Problems of Boolean Algebra and Switching Circuits*, Schaum's Outline Series, McGraw-Hill, New York, NY, USA (1970).

[43] **Givant, S.,** and **Halmos, P. R.**, *Lectures on Boolean Algebras*, Springer Science & Business Media, New York, NY, USA (2012).

[44] **Vladimirov, D. A.,** *Boolean Algebras in Analysis,* Springer Science & Business Media (2013).

[45] **Korfhage, R. R.,** *Discrete Computational Structures*, Academic Press, New York, NY, USA (2014).

[46] **Rushdi, A. M.**, Improved variable-entered Karnaugh map procedures, *Computers and Electrical Engineering*, **13** (1): 41-52 (1987).

[47] **Rushdi, A. M.** and **Al-Yahya, H. A.,** Further improved variable-entered Karnaugh map procedures or obtaining the irredundant forms of an incompletely-specified switching function, *Journal of King Abdulaziz University: Engineering Sciences,***13**(1): 111-152, (2001).

[48] **Rushdi, A. M.,** Map derivation of the minimal sum of a switching function from that of its complement, *Microelectronics and Reliability,* **25** (6):1055- 1065 (1985).

[49] **Rushdi, A. M.** and **Al-Yahya, H. A.,** A Boolean minimization procedure using the variable entered Karnaugh map and the generalized consensus concept, *International Journal of Electronics,* **87** (7): 769-794 (2000).

[50] **Crama, Y.** and **P. L. Hammer, P. L.,** *Boolean Functions*: *Theory, Algorithms, and Applications*, Cambridge University Press, Cambridge, United Kingdom (2011).

دراسة المشبعية في أنواع الجبر البولاني الكبير بواسطة حل المعادلات البولانية

علي محمد علي رشدي و وليد أحمد

قسم الهندسة الكهربائية وهندسة الحاسبات، كلية الهندسة، جامعة الملك عبدالعزيز، ص.ب ٨٠٢٠٤.، جدة ٢١٥٨٩، المملكة العربية السعودية

arushdi@kau.edu.sa

المستخلص. تمت دراسة المشبعية (ش ب ع) في أنواع الجبر البولاني الذري المحدود الأكبر من الجبر ثنائي القيمة B_2، وهي الأنواع المسماة بالجبر البولاني الكبير. وخلافا للصيغة الجبرية$g(\boldsymbol{X})$ المدروسة في مسألة المشبعية على الجبر ثنائي القيمة B_2، والتي قد تكون إما مشبعة أو غير مشبعة، ، فإن مثل هذه الصيغة في مسألة المشبعية على جبر بولاني كبير يكون لها حالات ثلاث، فهي قد تكون مشبعة دون شروط، أو مشبعة شرطيا أو غير مشبعة، ويعتمد ذلك على طبيعة شرط الاتساق للمعادلة المدروسة$\{g(\boldsymbol{X}) = 1\}$ ، حيث إن لهذا الشرط ثلاثة أحوال، فهو قد يأخذ شكل متطابقة، وقد يكون معادلة أصيلة، كما أنه قد يأتي على هيئة متناقضة. تناول البحث هذه مسألة المشبعية الأخيرة باستعمال طريقة تقليدية وأخرى مبتكرة لاشتقاق الحلول العامة المعلمية، ومن ثم توظيف أشجار المفكوكات لتوليد جميع الحلول الخاصة للمعادلة البولانية سالفة الذكر. إن أيا من هاتين الطريقتين يمكن أن تصاغ في صورة جبرية بحتة، ولكنها تصبح أوضح في التصور وأيسر في الفهم إذا عرضت من خلال الخريطة الطبيعية لدوال الجبر البولاني الكبير، وهي الخريطة المعروفة (لأسباب تاريخية) باسم خريطة كارنوه متغيرة المحتويات (خ ك غ ح). في الطريقة التقليدية يتم تصغير عدد المعالم تصغيرا أعظميا، فتنشأ حلول ملمومة موجزة، غير أن المعالم تنتمي للجبر البولاني الكبير محل الدراسة. وعلى النقيض من ذلك، لا تسعى الطريقة المبتكرة لتصغير عدد المعالم تصغيرا أعظميا، بل تستخدم معالم مستقلة تنتمي إلى الجبر ثنائي القيمة، وذلك لكل ذرة مؤكدة في مفكوك بول-شانون للصيغة $g(\boldsymbol{X})$. ورغم أن هذه الطريقة تنتج تعبيرات غير ملمومة، فإنها أسرع كثيرا في توليدها للحلول الخاصة حيث تصبح شجرة المفكوكات شجرة ثنائية. يتم شرح الطريقتين بمثالين تفصيليين.

كلمات مفتاحية: المشبعية، أنواع الجبر البولاني الكبير، حل المعادلات البولانية، الحلول المعلمية، الحلول الخاصة، طريقة مبتكرة.