A Theory of Coteries: Mutual Exclusion in Distributed Systems

Toshihide Ibaraki, *Member, IEEE,* and Tiko Kameda

***Abstract*—A coterie under a ground set U consists of subsets (called quorums) of U such that any pair of quorums intersect each other. "Nondominated (ND)" coteries are of particular interest, since they are "optimal" in some sense. By assigning a Boolean variable to each element in U, we represent a family of subsets of U by a Boolean function of these variables. We characterize the ND coteries as exactly those families which can be represented by positive, self-dual functions. In this Boolean framework, we prove that any function representing an ND coterie can be decomposed into copies of the three-majority function, and this decomposition is representable as a binary tree. We also show that the class of ND coteries proposed by Agrawal and El Abbadi is related to a special case of the above binary decomposition, and that the composition proposed by Neilsen and Mizuno is closely related to the classical Ashenhurst decomposition of Boolean functions. A number of other results are also obtained. The compactness of the proofs of most of these results indicates the suitability of Boolean algebra for the analysis of coteries.**

***Index Terms*— Boolean algebra, coteries, decomposition of Boolean functions, distributed systems, mutual exclusion, self-dual Boolean functions.**

I. Introduction And Definitions

A. History and Motivation

A coterie consists of subsets of an underlying set, such that any pair of subsets in it has at least one element in common. (For a more precise definition see Section I-B.) The concept of coterie was introduced in [12] as a mathematical abstraction to model mutual exclusion in distributed systems [21], [30], in general, and a family of quorums in a replicated database system [13], [29], in particular.

Probably the best known example of a coterie consists of all the subsets of $\lceil (n+1)/2 \rceil$ elements, where n is the total number of elements in the underlying set. This idea can be readily generalized to the "vote-assignable" coterie [12], [13], [30]. In a vote-assignable coterie, each element u_i has weight w_i (positive integer), and a subset G belongs to the coterie if and only if it is a minimal subset satisfying $\sum_{u_i \in G} w_i > W/2$, where W is the sum of all the weights. There are also many other coteries, which are not vote-assignable. The main motivation for generalizing the "majority" idea to general coteries was to have a large number of choices in implementing distributed mutual exclusion, in order to select the most suitable one for the application in hand. Consider, for example, a replicated database, in which the copies of each data item are stored at different sites interconnected by a communication network. When the sites and/or the communication links are subject to failure, the availability of a data item is defined as the probability that all sites storing a quorum of copies are operating and can communicate with each other (so that the data item can be accessed for reading or writing). It is known that, for a general underlying network, the coterie with the highest availability may not be vote-assignable [25]. Therefore, we need to study other more general coteries. Also, if the underlying network has some special structure (e.g., mesh or hypercube), it is conceivable that the complexity of mutual exclusion protocols depends on the adopted coterie. In such a case, it would be advantageous to have a variety of coteries as candidates for implementation.

Consider a ground set U, consisting of n elements. Any subset of these elements can be represented by an n-vector, $X = (x_1, x_2, \cdots, x_n) \in \{0,1\}^n$, where x_i is 1 (0) if the ith element is (not) in the subset. In this paper, we represent a family of subsets of U by a Boolean function $f(X)$ such that $f(X) = 1$ if and only if the argument X represents a superset of some subset in the family. Using this representation, we can characterize an important class of coteries, called "nondominated (ND)" coteries [12]: a family of subsets forms an ND coterie if and only if $f(X)$ is positive and self-dual. Through this characterization, it becomes possible to utilize concepts and known results of Boolean functions (see, e.g., [14], [23]).

Besides the class of vote-assignable coteries (see Section VI) discussed in [12], there are several known classes of coteries. Maekawa [21] showed that, based on finite projective planes, nontrivial coteries can be constructed such that each quorum contains only $\lceil \sqrt{n} \rceil$ elements, where $n = |U|$. Agrawala and El Abbadi [1] proposed a new class of ND coteries based on complete binary trees. We generalize this class to "binary decomposable" (B-decomposable, for short) coteries, by removing the restrictions that the tree be complete and that each label be used only once. We then show that any ND coterie is B-decomposable. This follows from one of our main theorems which states that any positive, self-dual function can be realized by a binary tree composed of copies of the three-majority function (i.e., the three-input

Manuscript received October X, 1990; revised September X, 1992. This work was supported in part by Grant-in-Aid from the Ministriy of Education, Science, and Culture of Japan, and in part by grants from the Natural Sciences and Engineering Research Council of Canada and the Advanced Systems Institute of British Columbia.

T. Ibaraki is with the Department of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, Kyoto, Japan 606.

T. Kameda is with the School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6.

IEEE Log Number 9210517.

threshold function with the input weights equal to 1 and the threshold value equal to 2) one of whose inputs is a single variable. Another class of ND coteries has recently been identified by Neilsen and Mizuno [24]. In terms of our Boolean representation, it turns out that the decomposition operation used to describe this class is a special case of the classical Boolean function decomposition due to Ashenhurst [2], studied extensively in switching theory and other fields (see Remark 5.3).

A vote-assignable coterie has the desirable property that it can be represented by a set of n weights, where n is, as before, the number of elements in the underlying set. The only known general way of describing an arbitrary ND coterie, however, was to list all its quorums. This is not, in general, space-efficient, since a coterie can have an exponential number (in terms of n) of quorums. A composition of copies of the basic majority function that realizes a given positive, self-dual function (e.g., the B-decomposition mentioned above), on the other hand, often provides a compact method of describing an ND coterie.

Some concurrency control schemes for a replicated database system (e.g., [3], [7], [11], [15]) make use of a set of "read quorums" and a set of "write quorums" such that each write quorum intersects each write quorum and each read quorum but the read quorums need not intersect each other. To model such a system, we introduce "bicoteries" and "write-read coteries," and investigate their properties as well. They are closely related to "complementary quorum set" and "antiquorum set" defined in [3].

In the rest of this section we define terms that are used in this paper. Section II reviews, in the terminology of Boolean algebra, a number of known results as well as those that follow easily from definitions. In Section III, we investigate the relation among coteries, bicoteries, and write-read coteries. Our discussion of coterie decomposition begins in Section IV-A, and our main theorem mentioned above will be proved in Section IV-B. In Section IV-C, we consider the "nonredundantly B-decomposable" coteries, as a generalization of the class of coteries consisting of structured quorums [1]. Section IV-D considers decomposition of general coteries and its relevance to ND coteries. In Section V, we consider another kind of decomposition of ND coteries as a generalization of the method proposed in [24], and discuss its relevance to the Ashenhurst decomposition. Section VI focuses on vote-assignable coteries and their generalization to multidimensional vote-assignable coteries introduced in [8]. Preliminary versions of this paper appeared before as a technical report [16] and also in conference proceedings [17].

In this paper, we often omit the Boolean AND operator $\wedge$, and simply concatenate variables to be ANDed together. $\subseteq$ and $\subset$ will denote set inclusion and proper set inclusion, respectively.

B. Definitions

Coterie [12]: Let $U = \{u_1, u_2, \cdots, u_n\}$ be a ground set and C be a family of subsets of U. C is a *coterie* under U if

i) $G \in C \Rightarrow G \neq \phi$,

ii) [*minimality*] There are no $G, H \in C$ such that $G \subset H$, and

iii) [*intersection property*] $\forall G, H \in C$: $G \cap H \neq \phi$. □

According to the above definition, the empty family containing no subset is a coterie. The subsets belonging to a coterie are called *quorums*.

Domination [12]: Let C and D be coteries. C *dominates* D if $C \neq D$ and $\forall G \in D\ \exists H \in C$: $H \subseteq G$. A coterie C is *nondominated* (ND) if no coterie dominates C. □

To see that ND coteries are particularly useful in practical applications, let coterie D be dominated by C. Then if a partition (that results from a network partition failure) contains a quorum belonging to D, then it must contain a quorum belonging to C.

Bicoterie [11]: An ordered pair $B = (P, Q)$, where P and Q are families of subsets of U, is a *bicoterie* under U if

i) Both P and Q satisfy conditions i) and ii) for a coterie (with $C = P, Q$).

ii) $\forall G \in P\ \forall H \in Q$: $G \cap H \neq \phi$.

A bicoterie $B = (P, Q)$ is called a *write-read coterie* (*wr-coterie*, for short), if it satisfies

iii) $\forall G, H \in P$: $G \cap H \neq \phi$ (i.e., P is a coterie). □

In [3], families P and Q in a bicoterie $B = (P, Q)$ are said to be "complementary" to each other. A write-read coterie $B = (P, Q)$ models "write quorums" by P and "read quorums" by Q.

Domination [11]: Bicoterie $B = (P, Q)$ *dominates* bicoterie $B' = (R, S)$ if

i) $\forall G \in R\ \exists H \in P$: $H \subseteq G$,

ii) $\forall G \in S\ \exists H \in Q$: $H \subseteq G$, and

iii) $(P, Q) \neq (R, S)$, i.e., $P \neq R$ or $Q \neq S$.

A bicoterie B is said to be *nondominated* (ND) if no bicoterie dominates B. Similarly, a wr-coterie B is said to be *nondominated* (ND) if no wr-coterie dominates B. □

A nondominated bicoterie is called a "quorum agreement" in [3].

In this paper, we define the Boolean value 0 to be less than the Boolean value 1, i.e., $0 < 1$. For two Boolean vectors $X = (x_1, x_2, \cdots, x_n)$ and $Y = (y_1, y_2, \cdots, y_n)$, we write $X \leq Y$ if $x_i \leq y_i$ for all $i = 1, 2, \cdots, n$.

Positive Boolean function: A Boolean function f is *positive* (or *monotone increasing*) if $\forall X, Y \in \{0,1\}^n$: $X \leq Y \Rightarrow f(X) \leq f(Y)$. □

Minimal vectors: Let f be a positive Boolean function. $X \in \{0,1\}^n$ is a *minimal vector* of f if $f(X) = 1$ and $[(Y \leq X) \wedge (Y \neq X)] \Rightarrow f(Y) = 0$. The set of all minimal vectors (or the subsets they represent according to our convention given below) of function f will be denoted by $\mathcal{M}in\mathcal{S}et(f)$. □

Let C be a family of distinct nonempty subsets of U satisfying the minimality condition. In this paper, we do not distinguish subset $G \in C$ from its characteristic vector, $(g_1, g_2, \cdots, g_n)$, where for $i = 1, 2, \cdots, n$,

$$g_i = \begin{cases} 1 & \text{if } u_i \in G, \\ 0 & \text{if } u_i \notin G. \end{cases}$$

Define Boolean function f_C associated with a family C of subsets of U by

$$f_C(X) = \begin{cases} 1 & \text{if } X \geq G \text{ for some } G \in C, \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that $\mathcal{M}inSet(f_C) = C$ holds if C satisfies the minimality condition.

Remark 1.1: It is known (e.g., [23]) that any positive function f can be represented by a unique disjunctive form without complemented literals such that each term is the prime implicant corresponding to a minimal vector. This form is called the ***minimum sum expression*** for f. For f_C, it has the following form.

$$f_C(X) = \bigvee_{G \in C} \{ \bigwedge_{u_i \in G} u_i \}.$$

In the above expression, for simplicity, we abuse the notation and use u_i (which is an element of U) as the ith component of vector X as well. We use this convention throughout this paper. □

Example 1.1: For $U = \{w, x, y, z\}$, consider a family of subsets

$$C = \{\{w, x, y\}, \{w, x, z\}, \{y, z\}\}.$$

It is easy to check that this C is a coterie. The positive Boolean function representing this C is

$$f_C = wxy \vee wxz \vee yz.$$

There is an obvious correspondence between the prime implicants of f_C and the subsets in C. □

Property 1.1: The Boolean function f_C defined above is positive for any family C. The correspondence between the families C of subsets of U, satisfying the minimality condition, and the positive Boolean functions f_C is one-to-one. □

For two Boolean functions f and g on the same set of variables, we write $f \leq g$ if and only if $\forall X \in \{0,1\}^n$: $f(X) = 1 \Rightarrow g(X) = 1$, and we write $f < g$ if and only if $f \leq g$ and $f \neq g$. The following property is immediate from the definition of domination.

Property 1.2: Let C_1 and C_2 be two coteries under the same underlying set. Then C_1 dominates C_2 if and only if $f_{C_2} < f_{C_1}$. □

Dual Boolean function: For any Boolean function f, its *dual*, denoted f^d, is defined by

$$f^d(X) = \bar{f}(\bar{X})$$

where $\bar{X}$ is the complement[1] of $X \in \{0,1\}^n$, and $\bar{f}$ denotes the complement of f. □

Remark 1.2: Given an expression for function f, an expression for f^d is obtained by interchanging $\wedge$ and $\vee$. If f involves a constant 1 (0), it should be changed to 0 (1). For example, a positive function

$$f_C = \bigvee_{G \in C} \{ \bigwedge_{u_i \in G} u_i \}$$

has its dual

$$f_C^d = \bigwedge_{G \in C} \{ \bigvee_{u_i \in G} u_i \}.$$

(This relationship between f_C and f_C^d implies that $\mathcal{M}inSet(f_C^d)$ represents the set of all minimal transversals [12] of C.) □

Property 1.3: Let f_C be the Boolean function representing a family C of nonempty subsets of U, which satisfies the minimality condition. Then the dual of f_C is given by

$$f_C^d(X) = \begin{cases} 0 & \text{if } \bar{X} \geq G \text{ for some } G \in C \\ 1 & \text{otherwise} \end{cases}.$$

□

Property 1.4: If f is positive, then so is f^d. □

Self-duality: A Boolean function f is said to be ***dual-minor*** if $f \leq f^d$, ***dual-major*** if $f^d \leq f$, and ***dual-comparable*** if $f \leq f^d$ or $f^d \leq f$ holds. f is said to be ***self-dual*** if $f = f^d$. □

Property 1.5:

i) f is dual-minor $\Leftrightarrow \forall X \in \{0,1\}^n$: at least one of $f(X)$ and $f(\bar{X})$ takes value 0.
ii) f is dual-major $\Leftrightarrow \forall X \in \{0,1\}^n$: at least one of $f(X)$ and $f(\bar{X})$ takes value 1.
iii) f is self-dual $\Leftrightarrow \forall X \in \{0,1\}^n$: exactly one of $f(X)$ and $f(\bar{X})$ takes value 1. □

Example 1.2: Note that a simple product term, e.g., $f = x_1 x_2 x_3$, is dual-minor, and its dual, $x_1 \vee x_2 \vee x_3$, is dual-major. For f_C of Example 1.1, we have

$$f_C^d = (w \vee x \vee y)(w \vee x \vee z)(y \vee z) = wy \vee wz \vee xy \vee yz \vee xz.$$

For this f_C, it is easy to see that $f_C \leq f_C^d$ (i.e., f_C is dual-minor), since, for each prime implicant P in f_C, there is a prime implicant P' in f_C^d that contains a subset of literals in P. An example of self-dual function is $f = xy \vee yz \vee zx$, since

$$f^d = (x \vee y)(y \vee z)(z \vee x) = xy \vee yz \vee zx = f.$$

□

II. Basic Results

This section lists a number of basic results which either follow easily from definitions or are already known [3], [11], [12]. The theorems below are based on the following properties of Boolean functions. Let C be a family of nonempty subsets of an underlying set U.

i) For any two subsets G and H of U: $G \cap H = \phi \Leftrightarrow G \leq \bar{H}$ $(= U - H)$ (viewed as Boolean vectors). Therefore, for a family of subsets C satisfying the minimality condition,

$$\begin{aligned} &\forall G, H \in C\colon G \cap H \neq \phi \text{ (intersection property)} \\ &\quad \Leftrightarrow \forall G, H \in C : G \not\leq \bar{H} \text{ and } H \not\leq \bar{G} \\ &\quad \Leftrightarrow f_C \text{ is dual-minor.} \end{aligned}$$

The last line above follows from the second last line, since, by Property 1.5i), if $f_C(X) = 1$ and $f_C(\bar{X}) = 1$ hold, i.e., $X \geq G$ for some $G \in C$ and $\bar{X} \geq H$ for some

[1] The ith bit of vector $\bar{X}$ is 1 (0) if and only if the ith bit of X is 0 (1).

$H \in C$, then $\bar{G} \geq \bar{X} \geq H$, a contradiction. This proves $\Rightarrow$. The converse can be proved similarly. Theorem 2.1, given below, now immediately follows.

ii) Let C_1 and C_2 be coteries under U. By Property 1.2, C_1 dominates C_2 if and only if $f_{C_2} < f_{C_1}$, or equivalently, $f^d_{C_1} < f^d_{C_2}$. Since f_{C_1} is dual-minor (by point i) above), we have $f_{C_1} \leq f^d_{C_1}$, hence $f_{C_2} < f^d_{C_2}$, i.e., f_{C_2} is not self-dual. Therefore

$$f_C \text{ is self-dual } \Rightarrow C \text{ is an ND coterie.}$$

Conversely, if f_C is dual-minor but not self-dual, then $f_C < f^d_C$ and there is a vector $Y \in \{0,1\}^n$ such that $f_C(Y) = 0$ and $f^d_C(Y) = 1$. Let $C' = C \cup \{Y\}$, where Y is regarded as a subset of U. $f_{C'}$ is dual-minor, since it satisfies Property 1.5i). (In particular, $f_{C'}(\bar{Z}) = 0$ for any $Z \geq Y$, since $f^d_C(Y) = 1 \Rightarrow f_C(\bar{Y}) = 0 \Rightarrow f_C(\bar{Z}) = 0$ for any $\bar{Z} \leq \bar{Y} \Rightarrow f_{C'}(\bar{Z}) = 0$ for any $\bar{Z} \leq \bar{Y}$.) By Theorem 2.1 below, $f_{C'}$ represents a coterie. Since $f_C < f_{C'}$ holds, C is not ND. Therefore,

$$C \text{ is an ND coterie } \Rightarrow f_C \text{ is self-dual.}$$

Theorem 2.1: Let C be a family of nonempty subsets of U satisfying the minimality condition. Then C is a coterie if and only if f_C is dual-minor. □

Theorem 2.2: Let C be a family of nonempty subsets of U satisfying the minimality condition. Then C is an ND coterie if and only if f_C is self-dual. □

The following corollary follows easily from Theorem 2.2 and Property 1.5iii).

Corollary 2.1: Let C be a coterie under U. Then C is ND if and only if for $\forall X \subseteq U$ exactly one of X and $\bar{X}$ contains some subset $G \in C$. □

Example 2.1: The function $f_C = wxy \vee wxz \vee yz$ of Example 1.1 is dual-minor as we saw in Example 1.2. Therefore, C is a coterie by Theorem 2.1. However, it is not an ND coterie, since f_C is not self-dual. In fact, C is dominated by another coterie $C' = \{\{x,y\}, \{y,z\}, \{z,x\}\}$, which is an ND coterie since $f_{C'} = xy \vee yz \vee zx$ is self-dual as observed in Example 1.2. □

The above properties can be easily extended to bicoteries and wr-coteries as well.

Theorem 2.3: Let $B = (P, Q)$, where P and Q are families of nonempty subsets of U satisfying the minimality condition. Then B is a bicoterie if and only if $f_Q \leq f^d_P$ (or equivalently $f_P \leq f^d_Q$). Similarly, B is a wr-coterie if and only if $f_Q \leq f^d_P$ and f_P is dual-minor. □

Theorem 2.4: Let $B = (P, Q)$ be as defined in Theorem 2.3. Then B is an ND bicoterie if and only if $f_Q = f^d_P$ (or equivalently $f_P = f^d_Q$). Also, B is an ND wr-coterie if and only if $f_Q = f^d_P$ and f_P is dual-minor. □

We can summarize the main results given above in terms of $\mathcal{M}inSet(f)$.

i) f is positive and self-dual $\Leftrightarrow$ $\mathcal{M}inSet(f)$ is an ND coterie.
ii) f is positive $\Leftrightarrow$ ($\mathcal{M}inSet(f)$, $\mathcal{M}inSet(f^d)$) is an ND bicoterie.
iii) f is positive and dual-minor $\Leftrightarrow$ ($\mathcal{M}inSet(f)$, $\mathcal{M}inSet(f^d)$) is an ND wr-coterie.

As a corollary to the above, we can relate the numbers of ND coteries, ND bicoteries, and ND wr-coteries to those of some classes of Boolean functions.

Corollary 2.2: Let $n = |U|$.

a) The number of ND coteries under U is equal to the number of positive, self-dual Boolean functions of n variables.
b) The number of ND bicoteries under U is equal to the number of positive Boolean functions of n variables. (Note that (P, Q) and (Q, P) are two distinct bicoteries unless $P = Q$.)
c) The number of ND wr-coteries under U is equal to the number of positive, dual-minor Boolean functions of n variables. This is the same as the number of coteries under U by Theorem 2.1. Also, as we shall see in the next section (Theorem 3.1), this is equal to the number of positive, self-dual functions of $n+1$ variables. □

Remark 2.1: [20] contains the following asymptotic formula for the number, $\Psi(n)$, of positive Boolean functions of n variables:

$$\Psi(n) \sim 2^{\binom{n}{n/2}} \exp\left\{\binom{n}{n/2-1}\left(\frac{1}{2^{n/2}} + \frac{n^2}{2^{n+5}} - \frac{n}{2^{n+6}}\right)\right\}$$

where $\binom{n}{k}$ stands for "n choose k." Reference [23] contains a lower bound,

$$2^{\binom{n-1}{(n+1)/2}},$$

on the number of positive, self-dual Boolean functions of n variables (for odd n). This bound is also found in [12] as a lower bound on the number of ND coteries. □

Remark 2.2: Given a family of subsets C, whether it is a coterie, i.e., f_C is dual-minor, can be decided in time polynomial in the number n of variables and the number of minimal vectors $m = |\mathcal{M}inSet(f_C)|$; this can be done by simply checking if every pair of minimal vectors has a common variable with value 1. On the other hand, the complexity of deciding if f_C is dual-major is known to be coNP-complete. However, the complexity of deciding if a dual-minor f_C is also dual-major, and hence if f_C is self-dual, is not known, as pointed out in [12] in a hypergraph setting. In fact, it is currently known as one of the open problems, which are possibly co-NP-complete.

The problem becomes solvable in polynomial time if we restrict ourselves to some special classes. For example, [10] contains some such special classes of hypergraphs (it also contains a variety of interesting applications of coterie-related concepts). Also, it can be shown that self-duality of a class of positive functions can be checked in polynomial time if and only if it can be decided in polynomial time whether two positive functions g and h in the class are duals of each other. The latter problem has a polynomial time algorithm if and only if the functions in the class can be "dualized" in polynomial time of their input and output lengths. Some classes of positive functions, including positive threshold functions, are known to have such a polynomial-time dualization algorithm [4], [9], [26]. □

III. Construction of ND Coteries from Coteries

Let f be a positive boolean function and x be one of its variables. The following decomposition, known as Shannon's decomposition, is a standard tool in switching theory.

$$f = xf_{x=1} \vee f_{x=0} \tag{1}$$

where $f_{x=1}$ ($f_{x=0}$) denotes the function obtained from f by substituting $x = 1$ ($x = 0$) into f. Note that $f_{x=0} \leq f_{x=1}$ since f is positive. Let $g = f_{x=0}$. If f is self-dual, then $f_{x=1} = (f_{x=0})^d = g^d$. Therefore, f can be decomposed as

$$f = xg^d \vee g. \tag{2}$$

Obviously, the converse (i.e., this f is self-dual) is also true, since

$$f^d = (g \vee x) \wedge g^d = gg^d \vee xg^d = g \vee xg^d = f \tag{3}$$

which follows from $f_{x=0} \leq f_{x=1}$, i.e., $g \leq g^d$. Also, as g is dual-minor, g defines a coterie by Theorem 2.1. From this analysis, the following theorem follows.

Theorem 3.1: There is a one-to-one correspondence between the ND coteries under $U_n = \{u_1, u_2, \cdots, u_n\}$ and the coteries under $U_{n-1} = \{u_1, u_2, \cdots, u_{n-1}\}$, where $n \geq 1$. In particular, the number of ND coteries under U_n is equal to the number of coteries under U_{n-1} [which is equal to the number of ND wr-coteries under U_{n-1} by Corollary 2.2c)]. □

The above theorem slightly strengthens Lemma 4.1 and Theorem 4.1 in [12], since the correspondence here is one-to-one.

Remark 3.1: In terms of subsets, $f_{x=1}$ is characterized by

$$\mathcal{M}inSet(f_{x=1}) = \mathcal{M}inimal(\{G - \{x\} | G \in \mathcal{M}inSet(f)\})$$

where $\mathcal{M}inimal(D)$ for a family D of subsets denotes the family obtained from D by eliminating duplicates and deleting all subsets that properly contain another subset in D. Similarly, the subsets represented by $f_{x=0}$ is given by

$$\mathcal{M}inSet(f_{x=0}) = \{G | G \in \mathcal{M}inSet(f) \text{ and } x \notin G\}.$$

□

Given an arbitrary positive function g, $B = (\mathcal{M}inSet(g), \mathcal{M}inSet(g^d))$ is an ND bicoterie by Theorem 2.4. However, B is not necessarily a wr-coterie. There are the following three cases.

i) $g^d = g$: g is self-dual, which implies that $P = \mathcal{M}inSet(g)$ is an ND coterie and $B = (P, P)$ is an ND wr-coterie.
ii) $g^d \neq g$ and g is dual-comparable: Since either g or g^d is dual-minor, assume without loss of generality that $g \leq g^d$ (i.e., g is dual-minor). Then g represents a coterie by Theorem 2.1, $B = (\mathcal{M}inSet(g), \mathcal{M}inSet(g^d))$ is an ND wr-coterie by Theorem 2.4, and $f = xg^d \vee g$ represents an ND coterie by Theorem 2.2 and (3). However, g^d does not represent any coterie.
iii) $g^d \neq g$ and g is not dual-comparable: Neither g nor g^d represents a coterie, and $B = (\mathcal{M}inSet(g), \mathcal{M}inSet(g^d))$ is not a wr-coterie.

IV. Decomposing Coteries

A. General Decomposition of Self-Dual Functions

We start with the following important theorem, which states that the class of positive, self-dual (dual-minor, dual-major) functions is closed under composition.

Theorem 4.1:

a) Let f_i, $i = 1, 2, \cdots, k$, be positive and self-dual. Let $F(x_1, \cdots, x_k)$ be also positive and self-dual. Then
$$f = F(f_1, f_2, \cdots, f_k)$$
is positive and self-dual.
b) Let f_i, $i = 1, 2, \cdots, k$, be positive and dual-minor (resp., dual-major). Let $F(x_1, \cdots, x_k)$ be also positive and dual-minor (resp., dual-major). Then
$$f = F(f_1, f_2, \cdots, f_k)$$
is positive and dual-minor (resp., dual-major).

Proof: a) The positivity of f is obvious and $f^d = f$ can be shown as follows.

$$\begin{aligned}[F(f_1, f_2, \cdots, f_k)]^d &= F^d(f_1^d, f_2^d, \cdots, f_k^d)\\ &= F(f_1, f_2, \cdots, f_k).\end{aligned}$$

b) Again, the positivity of f is obvious. We only prove the case where $\{f_i\}$ and F are all dual-minor. The dual-major case can be analogously proved. Since $f_i \leq f_i^d$, we have

$$\begin{aligned}&\forall X \in \{0,1\}^n : (f_1(X), f_2(X),\\ &\cdots, f_k(X)) \leq (f_1^d(X), f_2^d(X), \cdots, f_k^d(X)).\end{aligned}$$

This, combined with the positivity of F and $F \leq F^d$, yields

$$F(f_1, f_2, \cdots, f_k) \leq F^d(f_1^d, f_2^d, \cdots, f_k^d).$$

□

Example 4.1: Consider the following four positive, self-dual functions.

$$\begin{aligned}F &= xy \vee yz \vee zx,\\ f_1 &= ab \vee bc \vee ca,\\ f_2 &= ac \vee cd \vee da,\\ f_3 &= e.\end{aligned}$$

After substitution and a routine simplification, one obtains

$$f = F(f_1, f_2, f_3) = ac \vee abd \vee abe \vee ade \vee bcd \vee bce \vee cde.$$

This f is positive and self-dual by Theorem 4.1a). It is not difficult to see that

$$\mathcal{M}inSet(f) = \{\{a, c\}, \{a, b, d\}, \cdots, \{c, d, e\}\}$$

is in fact an ND coterie. □

Remark 4.1: It is known [12], [23] that there are very few positive, self-dual functions (hence ND coteries) that depend on[2] exactly n variables if n = 1, 2, or 3. In fact,

i) $n = 1$: There is only one such function $f = x$, where $U = \{x\}$.

[2] f *does not* depend on a variable x if for any combination of the values of the other variables, f takes the same value regardless of the value of x.

ii) $n = 2$: None.
iii) $n = 3$: There is only one such function $f = xy \vee yz \vee zx$, where $U = \{x, y, z\}$. □

We shall call the function

$$M(x, y, z) = xy \vee yz \vee zx$$

the *basic majority function*. By the above remark, this function represents the simplest nontrivial ND coterie. Theorem 4.1 implies that any function composed of basic majority functions (without using constant 0 or 1) is positive and self-dual. An interesting question would be: what class of positive, self-dual boolean functions can be composed from copies of the basic majority function? To answer this question, we define the *decomposable*, positive, self-dual function f on U as follows.

1) $f = x$, where $x \in U$, or
2) $f = M(f_1, f_2, f_3)$, where f_1, f_2 and f_3 are decomposable functions on U.

We shall show in the next section that every ND coterie is decomposable. Thus the basic majority function is a "universal function" for positive, self-dual boolean functions, just like NAND and NOR functions are universal functions for all boolean functions. In fact, it turns out that every ND coterie has a more specific form of decomposition, which we call "binary decomposition" (B-decomposition, for short), as will be discussed in the next section.

B. B-Decomposition of ND Coteries

Define the function on two variables,

$$M_x(y, z) = M(x, y, z),$$

based on the basic majority function. We consider composition of f from f_1 and f_2 (or decomposition of f into f_1 and f_2) of the form,

$$f = M_x(f_1, f_2) = (f_1 \vee f_2)x \vee (f_1 \wedge f_2) \quad (4)$$

where f_1 and f_2 are positive, self-dual functions. Note that f_1 and f_2 may depend on x. By Theorem 4.1, f is positive and self-dual. By considering $\mathcal{M}inSet(f)$, i.e., the family of subsets that f represents, we obtain the following result.

Theorem 4.2: Let C_1 and C_2 be ND coteries under U. The following family C is also an ND coterie.

$$C = \mathcal{M}inimal(\{G \cup \{x\} | G \in C_1 \cup C_2\} \cup \{G_1 \cup G_2 | G_1 \in C_1, G_2 \in C_2\}).$$

Proof: Let $f_1 = f_{C_1}$ and $f_2 = f_{C_2}$. Then $C = \mathcal{M}inSet(f)$, where f is given by (4). The rest follows from point i) given just after Theorem 2.4. □

The above decomposition (4) can be conveniently represented by a binary tree, as illustrated in Fig. 1, where the label of each internal node indicates the subscript x of the binary operator, M_x, used for decomposition. By recursively applying this decomposition to the generated f_i's, we may eventually end up with functions of single variables. A single-variable function is represented by a leaf node labeled by the variable.

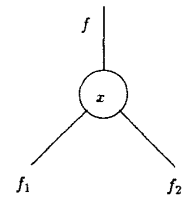


Fig. 1. Decomposition of f, using the binary operator M_x.

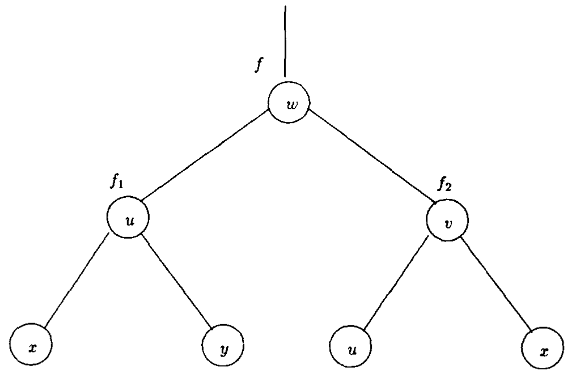


Fig. 2. Illustration for Example 4.2.

If all functions become single variables, then the generated finite tree is called the *binary decomposition tree* for f.

Example 4.2: Let f_1 and f_2 be defined by

$$f_1 = M_u(x, y) \text{ and } f_2 = M_v(u, x).$$

The decomposition tree T for the function f defined by

$$f = M_w(f_1, f_2) = (uy \vee xy \vee uv \vee vx)w \vee (ux \vee uvy \vee vxy)$$

is shown in Fig. 2. □

Remark 4.2: Different trees may represent the same ND coterie. For example, a tree consisting of the root labeled y and two child nodes labeled x and y represents an ND coterie $\{\{y\}\}$, which is also representable by a one-node tree labeled y. □

We say that a positive function f [or the ND coterie $\mathcal{M}inSet(f)$] has a *binary decomposition* (*B-decomposition*, for short) or f is *binary decomposable* (*B-decomposable*, for short) if

1) f is a single variable, or
2) $f = M_x(f_1, f_2)$ for some variable x, where both f_1 and f_2 have B-decompositions.

To prove that every ND coterie is B-decomposable, we start with some lemmas.

Lemma 4.1: Let a nonconstant function g be positive and dual-major (resp., dual-minor). Then g can be represented as

$$g = f_1 \vee f_2 \vee \cdots \vee f_k \text{ (resp.}, g = f_1 \wedge f_2 \wedge \cdots \wedge f_k) \quad (5)$$

for some $k > 0$, where each f_i is positive and self-dual.

Proof: We only prove the case where g is dual-major. The proof for the dual-minor case can be obtained by taking the dual of the dual-major case. Observe the following.

i) Since g is dual-major, at least one of $g(X)$ and $g(\bar{X})$ equals 1 for any vector X [by Property 1.5ii)].
ii) Since g is positive, $g(Y) = 1$ implies $g(X) = 1$ for any $X \geq Y$. (Or equivalently, $g(Y) = 0$ implies $g(X) = 0$ for any $X \leq Y$.)

Given a dual-major g, we first present a formal procedure for constructing, for an arbitrary $Y \in \mathcal{M}inSet(g)$, a positive, self-dual function f_Y, satisfying $f_Y(Y) = 1$ and $f_Y \leq g$. The following procedure constructs f_Y in terms of its true domain and false domain, T and F, respectively [i.e., $f_Y(X) = 1$ (resp., 0) for all $X \in T$ (resp., F)].

1) Let $T := \{X | X \geq Y\}$ and $F := \{Z | g(Z) = 0\}$. (This makes sure that $f_Y(Y) = 1$ and $f_Y \leq g$ hold initially.)
2) If $\exists Z \notin T \cup F$ such that $\bar{Z} \in T$, then $F := F \cup \{X | X \leq Z\}$. (Since it is self-dual, $f_Y(X)$ must satisfy $f_Y(\bar{Z}) = 1 \Rightarrow f_Y(Z) = 0$.) Repeat this step as long as it is applicable.
3) If $T \cup F = \{0,1\}^n$, then stop. Otherwise, choose a vector $Z \notin T \cup F$ ($T \cup F$ contains neither Z nor $\bar{Z}$), and let $T := T \cup \{X | X \geq Z\}$. Return to step 2.

Clearly, the above procedure terminates, since T always gets enlarged in step 3. The fact that the resulting f_Y is indeed self-dual follows from the property of T and F that, for any vector X, exactly one of X and $\bar{X}$ belongs to T (and the other belongs to F). We now prove this property. Since $T \cup F = \{0,1\}^n$ upon termination, it suffices to show that neither T nor F contains both X and $\bar{X}$ during the construction. That this is true initially, i.e., for the sets T and F of step 1 above, can be shown as follows. Since $X \geq Y (\neq 0)$ obviously means $\bar{X} \not\geq Y$, $X \in T$ implies $\bar{X} \notin T$. Also, $X \in F$ implies $g(X) = 0$ and hence $g(\bar{X}) = 1$, i.e., $\bar{X} \notin F$, since g is dual-major. Next, consider the case where F is augmented in step 2. The resulting set F does not contain both X and $\bar{X}$ for any X, since for any new vector $X \leq Z$ in F, $\bar{X} \in T$ holds because $\bar{X} \geq \bar{Z}$. Finally, in step 3, the resulting set T has this property, since for any new vector $X \geq Z$ in T, $\bar{X} \notin T$ holds because $\bar{X} \leq \bar{Z}$ and $\bar{Z} \notin T$ (if $\bar{Z} \in T$ then $Z \in F$ by step 2). The positivity of f_Y is obvious from the construction.

Applying the above procedure to every minimal vector Y of g, we obtain

$$g = \bigvee_{Y \in \mathcal{M}inSet(g)} f_Y.$$

□

Remark 4.3: In the above construction, many f_Y can be redundant. It is usually possible to make k in (2) fairly small. However, the true minimization is left open. □

Lemma 4.2: Let f be a positive, self-dual function on U. Then f can be decomposed as follows.

$$f = (f_1 \vee f_2 \vee \cdots \vee f_k)x \vee (f_1 \wedge f_2 \wedge \cdots \wedge f_k)$$

where $x \in U$ and each f_i is a positive, self-dual function, which does not depend on x.

Proof: Take any $x \in U$ and decompose f with respect to x as in (3) of Section III, i.e.,

$$f = xf_{x=1} \vee f_{x=0} = xg^d \vee g,$$

where $g = f_{x=0}$ and g is dual-minor. By Lemma 4.1, g^d can be represented as

$$g^d = f_1 \vee f_2 \vee \cdots \vee f_k$$

and g as

$$g = f_1^d \wedge f_2^d \wedge \cdots \wedge f_k^d = f_1 \wedge f_2 \wedge \cdots \wedge f_k$$

for some self-dual functions, $f_1, f_2, \cdots, f_k$, on $U - \{x\}$. □

The next lemma shows that f can be composed from functions $f_1, f_2, \cdots, f_k$, using only the basic majority function.

Lemma 4.3: Consider the decomposition

$$f = (f_1 \vee f_2 \vee \cdots \vee f_k)x \vee (f_1 \wedge f_2 \wedge \cdots \wedge f_k)$$

in Lemma 4.2. This can be represented as

$$f = M_x(f_1, F_2)$$

and

$$F_i = M_x(f_i, F_{i+1}), \qquad \text{for } i = 2, 3, \cdots, k-1,$$

where $F_k = f_k$.

Proof: Easy if one treats the two cases, $x = 0$ and $x = 1$, separately. □

Theorem 4.3: Any ND coterie is B-decomposable.

Proof: By Lemmas 4.2 and 4.3, a self-dual function f has a B-decomposition into smaller self-dual functions $\{f_i\}$, which do not depend on the variable x used for decomposition. Apply this decomposition recursively to all f_i until only single variables result. □

Corollary 4.1: Any ND coterie is decomposable. □

Now, B-decomposition discussed so far defines a single-output circuit made up of copies of the basic majority gate, whose inputs are variables from U (no constants) and whose output is f (Fig. 3). To check whether a given subset $S \subseteq U$ contains a quorum of the corresponding ND coterie, we can simply feed (the vector representing) S into this circuit. If the output is 1, then S contains a quorum of the ND coterie; otherwise, it does not. In this sense, the circuit is a compact representation of an ND coterie, which often requires less space than a list of all the quorums of the ND coterie.

Remark 4.4: B-decomposition can be generalized if we substitute the basic majority function of (4.2) by any positive self-dual $(k+1)$-ary function F and represent

$$F(x, f_1, f_2, \cdots, f_k)$$

by a node that is labeled by x and has $k (\geq 2)$ child nodes, $f_1, f_2, \cdots, f_k$. In particular, if F represents a vote-assignable ND coterie (see Section VI for definition), in which each f_i has weight 1 and x has weight $k-1$, it is given by

$$F(x, f_1, f_2, \cdots, f_k) = (f_1 \vee f_2 \vee \cdots \vee f_k)x \vee (f_1 \wedge f_2 \wedge \cdots \wedge f_k), \tag{6}$$

and may be considered as a direct extension of (4). □

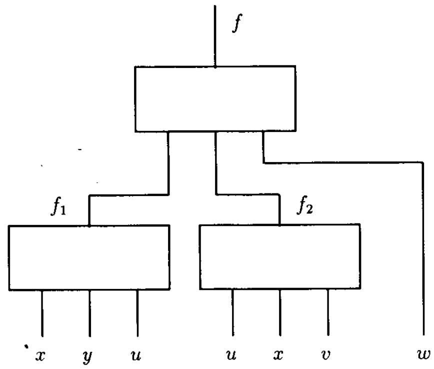


Fig. 3. A network of basic majority functions corresponding to Fig. 2.

C. Nonredundant B-Decomposition of ND Coteries

An ND coterie is said to be *nonredundantly B-decomposable* or to have a *nonredundant B-decomposition*, if it has a B-decomposition in which all node labels are different. In this case, we need to consider a decomposition

$$f = M_y(f_1, f_2) = (f_1 \vee f_2)y \vee (f_1 \wedge f_2) \quad (7)$$

where both f_1 and f_2 are positive, self-dual and, furthermore, their variable sets are mutually disjoint and do not contain y. In decomposition (7), we say that y is the *root* of the corresponding decomposition tree.

Lemma 4.4: Assume that a positive, self-dual function f has a decomposition of the form (7) and that f_1 depends on a variable w. Then f has another decomposition tree with root w if and only if $f_1 = w$.

Proof: If: If $f_1 = w$, we have

$$f = M_y(w, f_2) = M(y, w, f_2) = M_w(y, f_2).$$

Therefore, f has a decomposition tree whose root is w.

Only-if Assume that f_1 in decomposition (7) depends on variable w and some others. Assume further that w is the root of some decomposition tree for f. Using a theorem of Boolean algebra, we derive

$$\begin{aligned} f &= (f_1 \vee f_2)y \vee (f_1 \wedge f_2) \\ &= ((f_1)_{w=1} \vee f_2)yw \vee ((f_1)_{w=1} \wedge f_2)w \\ &\quad \vee ((f_1)_{w=0} \vee f_2)y \vee ((f_1)_{w=0} \wedge f_2) \\ &= [((f_1)_{w=1} \vee f_2)y \vee ((f_1)_{w=1} \wedge f_2)]w \\ &\quad \vee ((f_1)_{w=0} \vee f_2)y \vee ((f_1)_{w=0} \wedge f_2). \end{aligned}$$

Define g by

$$g^d = ((f_1)_{w=1} \vee f_2)y \vee ((f_1)_{w=1} \wedge f_2),$$

i.e., the coefficient of w in the expression for f. Since w is the root (i.e., there is a decomposition of type (7), $f = (g_1 \vee g_2)w \vee (g_1 \wedge g_2)$), g^d must be of the form

$$g^d = g_1 \vee g_2$$

where g_1 and g_2 are positive, self-dual functions defined on two disjoint sets of variables. Now, as noted in Remark 1.1, a positive function has a unique disjunctive form, called the minimum sum, consisting of all of its prime implicants. From the definition of g^d, it is easy to see that the minimum sum of g^d consists of prime implicants of the form

$$yP_1, yP_2 \text{ or } P_1P_2,$$

where P_1 (resp., P_2) is a prime implicant of $(f_1)_{w=1}$ (resp. f_2). Since both g_1 and g_2 cannot depend on y, assume without loss of generality that all terms of the form yP_1 or yP_2 belong to g_1. Then g_2 must contain at least one term P_1P_2, and g_1 and g_2 share all the variables in P_1 and P_2, a contradiction to the assumption on $\{g_j\}$. □

Lemma 4.5: Let f be a positive, self-dual function that depends on more than three variables[3] and has a nonredundant B-decomposition. A variable y can be the root of the tree representing a B-decomposition of f, if and only if one of the following conditions holds, where k is the size of the smallest subset in $\mathcal{M}in\mathcal{S}et(f)$.[4]

1) y is the unique variable that appears in all subsets in $\mathcal{M}in\mathcal{S}et(f)$ of sizes $k, k+1, \cdots, k+j$ for some $j \geq 0$.
2) $k = 2$ and there is exactly one subset of size 2 in $\mathcal{M}in\mathcal{S}et(f)$, and it contains y.

Proof: Consider decomposition (7), where y is the root and f_1 and f_2 are defined on disjoint sets of variables. First, assume that both f_1 and f_2 depend on more than one variable, and, for $i = 1, 2$, let k_i denote the size of the smallest subset in $\mathcal{M}in\mathcal{S}et(f_i)$. Without loss of generality, let $k_2 \geq k_1$, where $k_1 \geq 2$ by assumption on f_1 and f_2. Since every subset in $\mathcal{M}in\mathcal{S}et(f_1 \wedge f_2)$ has size at least $k_1 + k_2$, any subset $G \in \mathcal{M}in\mathcal{S}et(f)$ with a size between $k_1 + 1 (= k)$ and $k_2 + 1$ $(< k_1 + k_2)$, inclusive, satisfies $G \in \mathcal{M}in\mathcal{S}et((f_1 \vee f_2)y)$, and hence contains y. The variable y with this property is unique (hence must be the root), since there is no variable other than y that appears in all subsets $G \in \mathcal{M}in\mathcal{S}et(f)$ with $|G| \leq k_2+1$. (Recall that f_1 and f_2 are defined on disjoint sets of variables.)

Next, assume that f_1 or f_2 depends on exactly one variable. Without loss of generality, let f_1 depend on $w (\neq y)$ only. (In this case $k_2 \geq 2$, since f depends on more than three variables.) Then, $\{w, y\} \in \mathcal{M}in\mathcal{S}et(f)$ and hence $k = 2$. This subset $\{w, y\}$ is the only subset of size 2 in $\mathcal{M}in\mathcal{S}et(f)$, since f can be expressed as

$$f = (w \vee f_2)y \vee wf_2,$$

and f_2 depends on more than one variable, which is different from w. Then, as noted in the proof of the if-part of Lemma 4.4, either y or w (but not others) can be the root. This corresponds to the case 2 in the statement of the lemma. □

Example 4.3: Consider the positive, self-dual function f of Example 4.2,

$$f = ux \vee wxy \vee vwx \vee vxy \vee uvw \vee uvy \vee uwy.$$

The above lemma (condition 2) asserts that either one of u and x (and no other variable) can be the root of a decomposition tree. □

[3] Recall from Remark 4.1 that the positive, self-dual functions of up to three variables are very simple.

[4] Note that $k \geq 2$ follows from the assumptions on f.

Based on Lemma 4.5, it can be decided by the following algorithm whether or not a given positive function f represents a nonredundantly B-decomposable ND coterie.

Algorithm *DECOMPOSE:*

1. If f depends on at most three variables, then f represents a nonredundantly B-decomposable ND coterie if and only if $f = x$ for some variable x or f is the basic majority function. Stop. If f depends on more than three variables, go to step 2.
2. Check if f is dual-minor (i.e., $\mathcal{M}inSet(f)$ satisfies the intersection property). If not, f does not represent a coterie. Stop. Otherwise, check if there exists a variable y that satisfies one of the conditions in Lemma 4.5. If not, f does not represent a nonredundantly B-decomposable ND coterie. Stop. Otherwise, take such a y and go to step 3.
3. Compute $f_{y=1}$ [i.e., $S_1 = \mathcal{M}inimal(\{G - \{y\} | G \in \mathcal{M}inSet(f)\})$]. Decompose $f_{y=1}$ into a disjunction of two positive, dual-minor functions, f_1 and f_2, having disjoint sets of variables as follows. Starting with an arbitrarily chosen subset in family S_1, form the largest subfamily of S_1 by successively choosing subsets from S_1 which intersect some subset already chosen. Let f_1 represent this subfamily and let f_2 represent the subfamily of the remaining subsets. Then check if $f_{y=0}$ (i.e., $S_0 = \{G | G \in \mathcal{M}inSet(f)$ and $y \notin G\}$) satisfies
$$f_{y=0} = f_1 \wedge f_2. \tag{8}$$
If not, f does not represent a nonredundantly B-decomposable ND coterie. Stop. Otherwise, go to step 4.
4. Apply steps 1 to 3 recursively to test if both f_1 and f_2 represent nonredundantly B-decomposable ND coteries. f represents a nonredundantly B-decomposable ND coterie if and only if both of them represent nonredundantly B-decomposable ND coteries. □

Example 4.4: Let us apply *DECOMPOSE* to the positive, self-dual function f of Example 4.3. In step 2, we see that f is dual-minor by comparing every pair of prime implicants of f, and that each of u and x can be the root (as observed in Example 4.3). Choosing x as the root in step 2, we obtain in step 3

$$f_{x=1} = (vw \vee vy \vee wy) \vee u,$$

i.e., $f_1 = vw \vee vy \vee wy$ and $f_2 = u$, by starting with vw, for example. It is easy to check that

$$f_{x=0} = uvw \vee uvy \vee uwy = f_1 \wedge f_2.$$

Since f_1 is the basic majority function and f_2 is a single-variable function, this gives the nonredundant B-decomposition shown in Fig. 4(a). □

We show below that condition (8) in step 3 of *DECOMPOSE* implies that f is self-dual under the assumption that f_1 and f_2 are self-dual. Since $f_{y=1} = f_1 \vee f_2$, we have

$$(f_{y=1})^d = (f_1 \vee f_2)^d = f_1^d \wedge f_2^d = f_1 \wedge f_2.$$

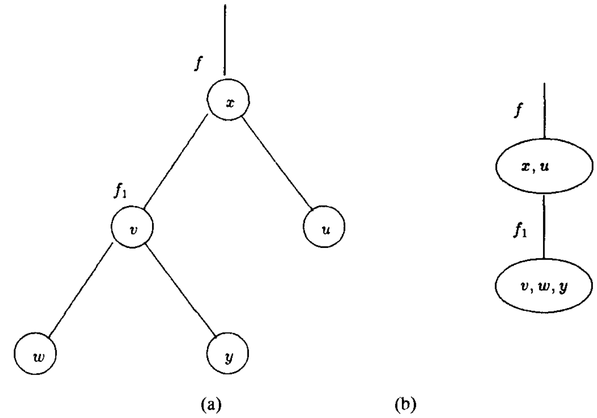


Fig. 4. Nonredundant B-decomposition of f in Examples 4.3 and 4.4. (a) Binary tree representation. (b) Compact representation.

Therefore,

$$f = yf_{y=1} \vee f_{y=0} \text{ [see(1)]}$$

becomes self-dual [see (3)] if and only if

$$f = yf_{y=1} \vee (f_{y=1})^d,$$

i.e., $f_{y=0} = f_1 \wedge f_2$ holds.

We now show that the time complexity of *DECOMPOSE* is polynomial. It is clear (see Remark 2.2) that steps 1 and 2 can be carried out in polynomial time in the number n of variables and $m = |\mathcal{M}inSet(f)|$. To see that step 3 is a polynomial time computation, first note that, by definition, S_1 and S_0 can be constructed from $\mathcal{M}inSet(f)$ in polynomial time. Decomposition of f into f_1 and f_2 can also be done in polynomial time, just by following the procedure stated therein. Finally, condition (8) can be tested in polynomial time by directly checking if $\mathcal{M}inSet(f_{y=0}) = \mathcal{M}inSet(f_1 \wedge f_2)$, where $\mathcal{M}inSet(f_1 \wedge f_2)$ can be obtained in polynomial time, using the relation

$$\begin{aligned}&\mathcal{M}inSet(f_1 \wedge f_2)\\ &= \mathcal{M}inimal(\{(G_1 \cup G_2) \mid G_1 \in \mathcal{M}inSet(f_1),\\ &\qquad G_2 \in \mathcal{M}inSet(f_2)\}).\end{aligned}$$

Step 4 of *DECOMPOSE* applies steps 1–3 recursively to f_1 and f_2. Since each decomposition of f into f_1 and f_2 partitions the set of variables into two disjoint subsets, the entire process can clearly be completed in polynomial time.

Theorem 4.4: Given a positive function f in the form of $\mathcal{M}inSet(f)$ (equivalently, as a set of its prime implicants), it can be decided in time polynomial in the number of variables n and $|\mathcal{M}inSet(f)|$ whether f represents a nonredundantly B-decomposable ND coterie. □

Some positive, self-dual functions are *not* nonredundantly B-decomposable. For example, the ND coterie consisting of all the three-element subsets of $U = \{u_1, u_2, \cdots, u_5\}$ gives rise to such a function.

Remark 4.5: The class of nonredundantly B-decomposable coteries proposed by Agrawal and El Abbadi [1] is constructed by the following algorithm. The input to the algorithm is a

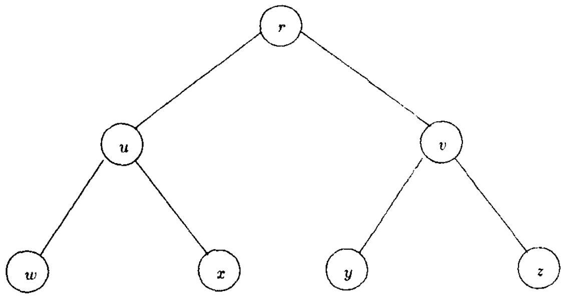


Fig. 5. A labeled complete binary tree used in the Agrawal–El Abbadi construction.

complete binary tree, in which all nodes have distinct labels from U.

Algorithm *GENERATE:*

1. Start with the set $\{r\}$, where $r \in U$ is the root. r is *unscanned.*
2. If no set contains an unscanned nonleaf node, then halt. The current family of sets is an ND coterie. Otherwise, choose an arbitrary set S containing an unscanned nonleaf node x, and go to step 3.
3. Let y and z be the child nodes of x. Generate from S the following three sets: $S \cup \{y\}$, $S \cup \{z\}$, and $(S \cup \{y, z\}) - \{x\}$. x is now scanned but y and z are not. Return to step 2.

As an example, consider the complete binary tree given in Fig. 5. Algorithm *GENERATE* generates sets S as illustrated in Table I. Read it from left to right. The underlined labels indicate those nodes which are unscanned and nonleaf. Therefore, the 15 sets containing only labels without underlines constitute the constructed coterie.

In view of the definition of B-decomposition and Theorem 4.2, it is not difficult to see that the coterie constructed as above is equal to $\mathcal{M}inSet(f)$, where f is the Boolean function associated with the given complete binary tree. The condition that the tree be complete can be easily incorporated into our algorithm *DECOMPOSE* by testing in step 2 whether two disjoint subsets of variables of f_1 and f_2 have the same cardinality.

It is clear that algorithm *GENERATE* can be applied not only to a complete binary tree but also to any binary tree with distinct node labels. Given such a tree as input, *GENERATE* will generate the nonredundantly B-decomposable ND coterie associated with the tree. *GENERATE* can also be generalized to nonbinary trees with distinct node labels, in which the Boolean function associated with each node is of type (6) stated in Remark 4.4. In this case, the family of sets generated from S in step 3 of *GENERATE* should read

$$S \cup \{y_1\}, S \cup \{y_2\}, \cdots, S \cup \{y_k\}, (S \cup \{y_1, y_2, \cdots, y_k\}) - \{x\},$$

if the current node has label x and k child nodes $y_1, y_2, \cdots, y_k$. The generalization of this type was already mentioned in [1]. □

Remark 4.6: Lemma 4.4 asserts that the nonredundant B-decomposition is unique if we do not distinguish the two roots which are possible only under the circumstance stated in that

TABLE I
APPLYING *GENERATE* TO THE EXAMPLE IN Fig. 5

<table>
<tr><td>$\{\underline{r}\}$</td><td>$\{r, \underline{u}\}$</td><td colspan="2">$\{r, u, w\}, \{r, u, x\}, \{r, w, x\}$</td></tr>
<tr><td></td><td>$\{r, \underline{v}\}$</td><td colspan="2">$\{r, v, y\}, \{r, v, z\}, \{r, y, z\}$</td></tr>
<tr><td></td><td>$\{\underline{u}, \underline{v}\}$</td><td>$\{u, w, \underline{v}\}$</td><td>$\{u, w, v, y\}, \{u, w, v, z\}, \{u, w, y, z\}$</td></tr>
<tr><td></td><td></td><td>$\{u, x, \underline{v}\}$</td><td>$\{u, x, v, y\}, \{u, x, v, z\}, \{u, x, y, z\}$</td></tr>
<tr><td></td><td></td><td>$\{w, x, \underline{v}\}$</td><td>$\{w, x, v, y\}, \{w, x, v, z\}, \{w, x, y, z\}$</td></tr>
</table>

lemma. To visualize this, introduce notations

$$M_{xy}(g) = xy \vee xg \vee yg$$

and

$$M_{xyz} = xy \vee yz \vee zx$$

where x, y, and z are variables, and represent $M_{xy}(g)$ by a node with label (x, y), having an edge to the node with label g, and M_{xyz} by a leaf node with label (x, y, z). Fig. 4(b) shows the resulting tree for the function f of Example 4.3. The binary tree representation of a nonredundantly B-decomposable function f with this compact notation is unique. □

Remark 4.7: The number of labeled rooted trees on n nodes, in which each node has either 0 or 2 child nodes, gives an upper bound on the number of nonredundantly B-decomposable ND coteries, as different trees may represent the same Boolean function. Such trees exist only for odd $n = 2m+1$, and their number is given as follows (see problem 21 in Section 3.4.4 of [19]).

$$\binom{2m+1}{m}(2m)!/2^m.$$

This is asymptotically much smaller than the number of ND coteries cited in Remark 2.1. □

D. B-Decomposition of General Coteries

The B-decomposition of ND coteries discussed in Section IV-B can be generalized to all other coteries with a slight modification. This will reveal an interesting structural relation between a coterie and the ND coteries that dominate it.

Consider the basic majority function, $M(x, y, z) = xy \vee yz \vee zx$. It is easily seen that $M_0(y, z) = M(0, y, z) = yz$ (AND function) and $M_1(y, z) = M(1, y, z) = y \vee z$ (OR function), which are conveniently represented by nodes with labels 0 and 1, respectively. Clearly, yz is dual-minor and $y \vee z$ is dual-major.

Now, consider B-decomposition with the modification that some nodes may now have labels 0 (but not 1). Since every node represents a positive, dual-minor function (recall that a self-dual function is also dual-minor), repeated applications of Theorem 4.1b) show that the function f it represents is positive and dual-minor, i.e., $C = \mathcal{M}inSet(f)$ is a coterie.

Example 4.5: The tree shown in Fig. 6 represents

$$\begin{aligned} f &= (f_1 \vee f_2)w \vee f_1 f_2 \\ &= (uv \vee ux \vee vx \vee xy)w \vee uxy \vee vxy \end{aligned}$$

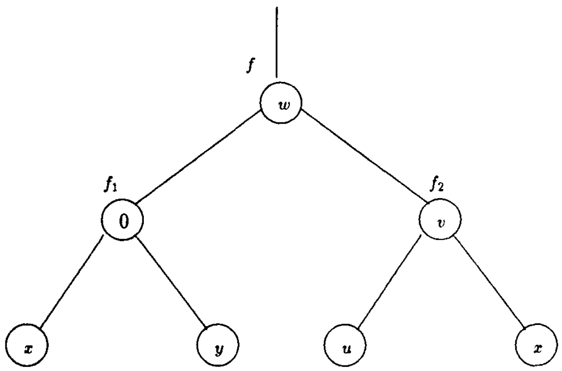


Fig. 6. A B-decomposition with basic majority functions and an AND function.

where $f_1 = xy$ and $f_2 = uv \vee vx \vee xu$. It is easy to see that $\mathcal{M}inSet(f)$ is in fact a coterie. □

The next theorem says that the above decomposition is general, i.e., any coterie has such a decomposition.

Theorem 4.5:

a) A function, represented by a binary tree composed of copies of the basic majority function and/or AND function, is positive and dual-minor.

b) Any positive, dual-minor function f has a B-decomposition composed of copies of the basic majority function and/or AND function.

Proof: As a) was already shown, we prove b) only. Since f is positive and dual-minor, Lemma 4.1 states that f can be represented as

$$f = f_1 \wedge f_2 \wedge \cdots \wedge f_k$$

for some $k > 0$, where each f_i is positive and self-dual. Recall that each f_i has a B-decomposition into basic majority functions by Theorem 4.3, and the conjunction of k f_i's in the above expression can be realized by using $k-1$ AND functions $M_0(y,z)$ as follows.

$$f_1 \wedge f_2 \wedge \cdots \wedge f_k = (\cdots((f_1 \wedge f_2) \wedge f_3) \cdots \wedge f_k).$$

Therefore, f has a B-decomposition as stated. □

Assume now that a B-decomposition T of the above theorem is given for a positive, dual-minor function f. Since the basic majority function is self-dual and the dual of AND is OR, i.e.,

$$(M_0(y,z))^d = M_1(y,z),$$

the B-decomposition T^d, obtained by switching all labels 0 to 1, represents f^d. Therefore, this can be used as a convenient way of defining an ND wr-coterie,

$$B = (\mathcal{M}inSet(f), \mathcal{M}inSet(f^d)),$$

as well as its representation by two trees.

A more interesting observation is that a self-dual function can be defined from T if all labels 0 in T are replaced by some variables. Since $M(0,y,z) \le M(u,y,z)$ for any variable u (which is possibly equal to y or z), the resulting self-dual function g satisfies $f \le g$. As observed in Section II, this means that g represents an ND coterie that dominates the original coterie $\mathcal{M}inSet(f)$. By assigning different variables to labels 0, we can generate many ND coteries that dominate coterie $\mathcal{M}inSet(f)$.

As an example, note that the B-decomposition of Fig. 2 can be obtained by assigning variable u to the node with label 0 in Fig. 6. It is easy to check that the ND coterie of Fig. 2, given in Example 4.2, dominates the coterie of Fig. 6, given in Example 4.5.

V. Decomposition of ND Coteries into Smaller Functions

Motivated by the recent work by Neilsen and Mizuno [24], we investigate in this section a special case of the general decomposition in Theorem 4.1a).

A. Decomposition Using a Self-Dual Function

Consider a function of the form

$$f = F(h, x_1, x_2, \cdots, x_n) \tag{9}$$

where F and h are positive and self-dual. We rewrite f as

$$f = F(x_0, x_1, x_2, \cdots, x_n),$$
$$x_0 = h(x_1, x_2, \cdots, x_n),$$

and expand F on x_0 as in (1). We thus obtain

$$\begin{aligned} f(X) &= F(1,X)x_0 \vee F(0,X) \\ &= g^d(X)x_0 \vee g(X) \\ &\quad \text{[by (2) since } f \text{ is self-dual by Theorem 4.1a)]} \\ &= g^d(X)h(X) \vee g(X) \end{aligned} \tag{10}$$

where $X = (x_1, x_2, \cdots, x_n)$, $g(X) = F(0,X)$, and g is dual-minor. This decomposition may be considered as a generalization of (2) in the sense that positive, self-dual "function" x_0 is replaced by a general positive, self-dual function h.

Decomposition (10) is useful if g and h are constructed from smaller functions; more precisely, if g depends on less than $n-1$ variables (i.e., F depends on less than n variables) and h is composed of a known class of functions, each depending on less than n variables. In the following subsections, we shall discuss two cases that meet this requirement.

B. Decomposition into Positive Self-Dual and Positive Functions

Before presenting the main result of this subsection, we describe some properties of positive functions $f(X)$, where $X = (x_1, x_2, \cdots, x_n)$ and $n \ge 3$. Expanding f on x_{n-1} and x_n by (3), we obtain

$$\begin{aligned} f(X) =& f_{11}(X')x_{n-1}x_n \vee f_{10}(X')x_{n-1} \vee f_{01}(X')x_n \\ &\vee f_{00}(X') \end{aligned} \tag{11}$$

where

$$X' = (x_1, x_2, \cdots, x_{n-2}) \text{ and}$$
$$f_{ab}(X') = f(X', a, b) \text{ for } a = 0, 1 \text{ and } b = 0, 1.$$

By the positivity of f, $\{f_{ab}\}$ are all positive, and we have

$$f_{00} \le f_{10} \le f_{11} \text{ and } f_{00} \le f_{01} \le f_{11}.$$

Note that f_{10} and f_{01} are, in general, not comparable.

Lemma 5.1: Consider decomposition (11) of a positive function $f(X)$ of n variables, where $n \ge 3$. f is self-dual if and only if

$$\begin{aligned} f_{00}(X') &= f_{11}^d(X') \text{ [equivalently, } f_{00}^d(X') = f_{11}(X')] \\ f_{01}(X') &= f_{10}^d(X') \text{ [equivalently, } f_{01}^d(X') = f_{10}(X')]. \end{aligned} \tag{12}$$

Proof: From (11), we get

$$f^d = (f_{11}^d \vee x_{n-1} \vee x_n)(f_{10}^d \vee x_{n-1})(f_{01}^d \vee x_n) f_{00}^d.$$

Since $f_{00} \le f_{10} \le f_{11}$ implies $f_{00}^d \ge f_{10}^d \ge f_{11}^d$, the above expression can be simplified to

$$f^d = f_{00}^d x_{n-1} x_n \vee f_{01}^d x_{n-1} \vee f_{10}^d x_n \vee f_{11}^d.$$

By considering all possible value assignments to x_{n-1} and x_n, and by noting that $f_{00} = f_{11}^d$ implies $f_{00}^d = f_{11}$, and so forth, it follows that $f = f^d$ holds if and only if condition (12) holds. □

Define $p(X')$ and $q(X')$ as follows.

$$p^d(X') = f_{11}(X') \text{ and } q(X') = f_{10}(X').$$

Lemma 5.1 implies that a positive function f is self-dual if and only if it has a representation

$$f(X) = p^d(X')x_{n-1}x_n \vee q(X')x_{n-1} \vee q^d(X')x_n \vee p(X'), \tag{13}$$

such that $p(X')$ and $q(X')$ are positive functions satisfying

$$p(X') \le q(X') \le p^d(X') \tag{14}$$
$$\text{(i.e., } p(X') \le q^d(X') \le p^d(X').)$$

Note that (13) can be considered as a rule for constructing a self-dual function f from smaller functions $p(X')$ and $q(X')$, where $p(X')$ and $q(X')$ can be chosen independently as long as (14) is satisfied. A special case of this decomposition is

$$f(X) = x_{n-1}x_n \vee q(X')x_{n-1} \vee q^d(X')x_n \tag{15}$$

(i.e., $p^d(X') = 1$ and $p(X') = 0$). In this case, condition (14) is trivially satisfied for any positive function $q(X')$.

Remark 5.1: Relation (15) implies the following inequality:

(The number of positive, self-dual functions of up to n variables) $\ge$ (The number of positive functions of up to $n-2$ variables).

However, this does not give any better lower bound on the number of positive, self-dual functions than that given in Remark 2.1. □

Now take any positive, self-dual function $f(X)$ and consider its decomposition (13). It can be rewritten as follows.

$$\begin{aligned} f(X) &= p^d(X')x_{n-1}x_n \vee q(X')x_{n-1} \vee q^d(X')x_n \vee p(X') \\ &= p^d(X')(x_{n-1}x_n \vee q(X')x_{n-1} \vee q^d(X')x_n) \\ &\quad \vee p(X') \text{ [by (2)]} \\ &= g^d(X)h(X) \vee g(X). \end{aligned}$$

where

$$\begin{aligned} g(X) &= p(X') \text{ and} \\ h(X) &= x_{n-1}x_n \vee q(X')x_{n-1} \vee q^d(X')x_n, \end{aligned} \tag{16}$$

proving the next theorem.

Theorem 5.1: For any positive, self-dual function of n variables, $f(X)$, the decomposition (10) decomposes $f(X)$ into smaller functions in the sense that $h(X)$ is defined by a positive function of $n-2$ variables [i.e., $q(X')$ in (16)] and $g(X)$ [$= p(X')$] contains $n-2$ variables. □

C. Decomposition into Disjoint Sets of Variables (A-Decomposition)

Another interesting case arises when the set of variables can be decomposed into two disjoint sets. In this case, (10) becomes

$$f(X) = g^d(Z)h(Y) \vee g(Z) \tag{17}$$

where

$$\begin{aligned} &X = (x_1, x_2, \cdots, x_n), \\ &Y = (x_{i_1}, x_{i_2}, \cdots, x_{i_k}), \\ &Z = (x_{i_{k+1}}, x_{i_{k+2}}, \cdots, x_{i_n}), \text{ and} \\ &(x_{i_1}, x_{i_2}, \cdots, x_{i_n}) \textit{ is a permutation of } (x_1, x_2, \cdots, x_n). \end{aligned} \tag{18}$$

Note that decomposition (2) is a special case of (17), where $h(Y)$ is a single variable. Decomposition (17) is not universal, since, for example, the ND coterie consisting of all the three-element subsets of $U = \{u_1, u_2, \cdots, u_5\}$ does not have such decomposition.

Remark 5.2: In [24], Neilsen and Mizuno proposed a coterie join algorithm that constructs an ND coterie C_3 over ground set $U = (U - \{x_0\}) \cup U'$ from two ND coteries C_1 and C_2 under disjoint ground sets U and U', respectively, where x_0 is a designated element in U. That is, $G_3 \in C_3$ holds if and only if

$$\begin{aligned} &G_3 = G_1 \textit{ for some } G_1 \in U \text{ with } x_0 \notin G_1, \textit{ or} \\ &G_3 = (G_1 - \{x_0\}) \cup G_2 \text{ for some } G_2 \in C_2 \text{ and some} \\ &\qquad G_1 \in U \text{ with } x_0 \in G_1. \end{aligned}$$

They show, among other things, that C_3 is an ND coterie if and only if so are C_1 and C_2.

Now, let $F(z_0, Z)$ [resp., $h(Y)$] be the positive, self-dual function associated with C_1 (resp., C_2) given in the following form:

$$F(z_0, Z) = g^d(Z)z_0 \vee g(Z),$$

where

$$g^d(Z) = F(1, Z) \text{ and } g(Z) = F(0, Z).$$

Then it is not difficult to see that the ND coterie given by $f(X)$ of (17) is nothing but the above ND coterie C_3. In this sense, (17) is a Boolean description of Neilsen–Mizuno join algorithm. An advantage of the Boolean approach is that the NDness is an immediate consequence of the self-duality of f, though a rather involved argument was necessary in [24] to show it directly from the join algorithm.

Moreover, the construction of C_3 as above is valid even if Y and Z are not disjoint, as it can be considered as an algorithm to compute $\mathcal{M}inSet(f)$ of general decomposition (10). However, in this case, some generated sets may be included in other sets, and, therefore, operation $\mathcal{M}inimal(C_3)$ is necessary to obtain the corresponding ND coterie. □

We point out here the relationship of decomposition (17) to the classical decomposition of Ashenhurst [2] (referred to henceforth as *A-decomposition*), which can be applied to any Boolean function. A Boolean function $f(X)$ of n (≥ 3) variables has an A-decomposition if it can be represented as

$$f(X) = F(h(Y), Z) \tag{19}$$

where F is a function of $n-k+1$ variables, h is a function of k (≥ 2) variables, and Y and Z are given by (18). The next lemma shows that decomposition (17) is equivalent to A-decomposition applied to the functions representing ND coteries.

Lemma 5.2: Let a Boolean function $f(X)$ that depends on n (≥ 3) variables be positive and self-dual. Then f has a decomposition of the form (17) if and only if it has an A-decomposition of the form (19).

Proof: Decomposition (17) is obviously a special case of A-decomposition. Conversely, if f has an A-decomposition, it can be decomposed as follows.

$$f(Y, Z) = F(1, Z)h(Y) \vee F(0, Z)\bar{h}(Y). \tag{20}$$

Since f depends on Y, there is an assignment $Z = A$ such that $F(1, A) = 1$ and $F(0, A) = 0$. (If not, there is an assignment such that $F(0, A) = 1$ and $F(1, A) = 0$. In this case, interchange the roles of h and $\bar{h}$ by redefining $\bar{h}$ as h.) Then

$$f(Y, A) = h(Y),$$

and h is positive since f is positive. Next, consider assignments $Y = B_1$ and $Y = B_2$ such that $B_1 > B_2$, and $h(B_1) = 1, h(B_2) = 0$, that is $f(B_1, Z) = F(1, Z)$ and $f(B_2, Z) = F(0, Z)$. Since f is positive in Y, this implies that

$$F(1, Z) \geq F(0, Z),$$

and (20) reduces to

$$f(Y, Z) = F(1, Z)h(Y) \vee F(0, Z). \tag{21}$$

Now,

$$\begin{aligned} f^d(Y, Z) &= (F^d(1, Z) \vee h^d(Y))F^d(0, Z) \\ &= F^d(0, Z)h^d(Y) \vee F^d(1, Z), \end{aligned}$$

and since f is self-dual,

$$F^d(0, Z)h^d(Y) \vee F^d(1, Z) = F(1, Z)h(Y) \vee F(0, Z).$$

Choosing an assignment (e.g., $Y = 0$) to Y such that $h(Y) = h^d(Y) = 0$, we have

$$F(0, Z) = F^d(1, Z) \text{ (i.e.,} F(1, Z) = F^d(0, Z)),$$

and choosing an assignment to Z such that $F(1, Z) = 1$ and $F(0, Z) = 0$, we have

$$h(Y) = h^d(Y) \text{ (i.e.,} h \text{ is self dual).}$$

Therefore, (21), hence (20), is a decomposition of the form (17). □

Remark 5.3: Properties and algorithms related to A-decomposition have been extensively studied in the literature of Boolean functions (e.g., [2], [14]). Interestingly, this subject has also been studied in other areas under various names, such as "committees" in game theory [28], "clutters" in set systems [5], and "modules" in coherent systems of reliability theory [6]. Based on the result of Billera [5], Möhring [22] developed an algorithm to test the existence of A-decomposition for a positive Boolean function (in Boolean terminology), which runs in polynomial time in n and $m = |\mathcal{M}inSet(f)|$. The time complexity was further improved to $O(n^2m^3)$ in [27]. □

As a special case of (17), assume that

$$g^d(Z) = g'(Z') \vee z \text{ and } g(Z) = g'(Z') \wedge z \tag{22}$$

where z is a variable in Z, and Z' denotes the vector of variables Z from which z is deleted, and $g'(Z')$ is positive and self-dual. Then (17) becomes

$$f(X) = (g'(Z') \vee h(Y))z \vee (g'(Z') \wedge h(Y))$$

which is in the same form as (7). Therefore, nonredundantly B-decomposable ND coteries in Section IV-C are contained in the class of ND coteries which are obtained by recursively applying decomposition (17) to all the generated functions (as long as they contain more than three variables).

VI. Vote-Assignable Coteries

A. Vote-Assignable Coteries and Threshold Functions

A Boolean function $f(X)$ is called a *threshold function* [23] if there is a set of *weights*, $w_1, \cdots, w_n$, and a *threshold* T such that $f(X) = 1$ if and only if $\sum_i w_i x_i \geq T$. Here, all weights and thresholds are nonnegative integers. Any threshold function with nonnegative weights is positive. The basic majority function is a threshold function with weights $w_1 = w_2 = w_3 = 1$ and threshold $T = 2$.

Since $f^d(X) = \bar{f}(\bar{X})$ by definition, we have $f^d(X) = 1 \Leftrightarrow f(\bar{X}) = 0 \Leftrightarrow \sum_i w_i(1 - x_i) < T \Leftrightarrow \sum_i w_i x_i \geq \sum_i w_i + 1 - T$.

From this observation, it follows that

i) If f is a threshold function, then f^d is also a threshold function, which is realizable with the same set of weights as f and a threshold of $T^d = W + 1 - T$, where $W = \sum_i w_i$,

ii) A threshold function f is dual-comparable, and is dual-minor (resp., dual-major) if $T \geq T^d$ (resp., $T \leq T^d$), and

iii) A threshold function f is self-dual if $W = \sum_i w_i$ is odd and $T = (W + 1)/2$.

The coterie represented by a function f is said to be *vote-assignable* [12] if there is a set of nonnegative integer weights, $w_1, \cdots, w_n$, such that $W = \sum_i w_i$ is odd and $f(X) = 1$ if and only if $\sum_i w_i x_i \geq (W+1)/2$. A bicoterie (P, Q) is said to be *vote-assignable* [11] if both f_P and f_Q are positive threshold functions. The following results now easily follow from the above three facts [11], [12].

Theorem 6.1:

a) $\mathcal{M}inSet(f)$ is a vote-assignable ND coterie if and only if f is a positive, self-dual threshold function.
b) Let $B = (\mathcal{M}inSet(g), \mathcal{M}inSet(g^d))$. B is a vote-assignable ND bicoterie if and only if g is a positive threshold function.
c) Let $B = (\mathcal{M}inSet(g), \mathcal{M}inSet(g^d))$. B is a vote-assignable ND wr-coterie if and only if g is a positive, dual-minor threshold function. □

Theorem 6.2: Let $B = (P, Q)$ be a vote-assignable ND bicoterie. Then either B or $B' = (Q, P)$ is a vote-assignable wr-coterie.

Proof: Follows from fact ii) above, Theorem 2.4, and Theorem 6.1b). □

Theorem 6.3: Let f represent any vote-assignable ND coterie under $U_n = \{u_1, u_2, \cdots, u_n\}$. f can be constructed from a vote-assignable ND wr-coterie $B = (\mathcal{M}inSet(g), \mathcal{M}inSet(g^d))$ under $U_{n-1} = \{u_1, u_2, \cdots, u_{n-1}\}$ by $f = g^d x_n \vee g$. Moreover, the correspondence between g and f is one-to-one. □

Corollary 6.1: Let $|U| = n$.

a) The number of vote-assignable ND coteries under U is equal to the number of positive, self-dual threshold functions of n variables.
b) The number of vote-assignable ND bicoteries under U is equal to the number of positive threshold functions of n variables. (Note that (P, Q) and (Q, P) are two distinct bicoteries unless $P = Q$.)
c) The number of vote-assignable ND wr-coteries under U is equal to the number of positive, self-dual threshold functions of $n + 1$ variables. □

Remark 6.1: The number of positive threshold functions of n variables $N(n)$ (which is equal to the number of positive, self-dual threshold functions of $n + 1$ variables) is bounded from below and above as follows [31], [32].

$$2^{n(n-3)/2} \leq N(n) \leq 2^{1-n} \sum_{i=0}^{n} \binom{2^n - 1}{i} (\leq 2^{n^2}).$$

Recently it has been shown in [33] that

$$\log_2 N(n) \to n^2,$$

as n tends to infinity. Comparing this with the number of positive, self-dual functions, cited in Remark 2.1, we see that the vote-assignable ND coteries form only a very small fraction of all ND coteries. □

B. Multidimensional Vote-Assignable Coteries

In [8], the concept of multidimensional voting is introduced as a generalization of vote-assignable coteries. In our terminology, a family C of subsets of U is said to have an $MD(l, k)$ *voting* if $C = \mathcal{M}inSet(f)$ for some Boolean function

$$f = F(f_1, f_2, \cdots, f_k) \tag{23}$$

where F is a threshold function with weights $w_{0i} = 1$ $(i = 1, 2, \cdots, k)$ and threshold $T_0 = l$, and each f_i $(i = 1, 2, \cdots, k)$ is a positive threshold function of $n = |U|$ variables with weights w_{ij} $(j = 1, 2, \cdots, n)$ and threshold T_i. Here, all weights and thresholds are nonnegative integers. They have shown, among other things, the following properties.

i) Any coterie has an $MD(l, k)$ voting for some l and k.
ii) Let $f = F(f_1, f_2, \cdots, f_k)$ be an $MD(l, k)$ voting that represents a coterie P, and let Q be the family of subsets defined by the $MD(k + 1 - l, k)$ voting $f' = F'(f_1^d, f_2^d, \cdots, f_k^d)$. Then, $B = (P, Q)$ is a wr-coterie.

One problem here is that it is not obvious whether the family of subsets represented by an $MD(l, k)$ voting is a coterie, or whether it is an ND coterie, unless we check all subsets along the line discussed so far. An easy way to generate only coteries or only ND coteries by $MD(l, k)$ voting is given below.

Theorem 6.4:

a) Let an $MD(l, k)$ voting (23) satisfy
 - $l \geq \lceil (k + 1)/2 \rceil$, and
 - each f_i $(i = 1, 2, \cdots, k)$ is dual-minor.

 Then $\mathcal{M}inSet(f)$ is a coterie.
b) Let an $MD(l, k)$ voting (23) satisfy
 - k is odd and $l = (k + 1)/2$, and
 - each f_i $(i = 1, 2, \cdots, k)$ is self-dual.

 Then $\mathcal{M}inSet(f)$ is an ND coterie.

Proof: Immediate from Theorems 4.1, 2.1, and 2.2, since function F with $l \geq \lceil (k + 1)/2 \rceil$ (resp., $l = (k + 1)/2$ for an odd k) is dual-minor (resp., self-dual) as observed in Section VI-A. □

Unfortunately, it is not known whether the $MD(l, k)$ votings described in Theorem 6.4 are universal [i.e., can generate all coteries (ND coteries)] or not. However, it can be shown that $MD(l, k)$ votings satisfying the conditions of Theorem 6.4a) are universal for coteries if and only if those satisfying the conditions of Theorem 6.4b) are universal for ND coteries. This follows from the observation that the self-dual function f defined by (2) (Section III) from a dual-minor function g is a positive threshold function if and only if g is a positive threshold function.

Property ii) above can be strengthened slightly by Theorem 2.4, since f' referred to in ii) is equal to f^d by the fact that f is an $MD(l, k)$ voting and f' is an $MD(k + 1 - l, k)$ voting (see also Property i) in Section VI-A).

Theorem 6.5: Let $B = (P, Q)$ be defined as in property ii) above. Then B is an ND wr-coterie. □

VII. Conclusions

We have shown that Boolean functions provide an elegant way of modeling coteries, bicoteries, and write-read coteries. ND coteries, ND bicoteries, and ND wr-coteries have been characterized in a very simple way in this framework, and many properties have been proved and/or reproved from these characterizations. We have also discussed decompositions of ND coteries extensively. Shannon decomposition (1) and Ashenhurst decomposition (19) provided theoretical basis for these coterie decompositions.

One open problem we are currently investigating is: to describe a given ND coterie, how should one design the corresponding network of basic majority functions? Although the lemmas in Section IV-B give a construction of such a network, it is desirable to design an optimal network (e.g., using the minimum number of copies of the basic majority function). This is reminiscent of the minimization of the number of NAND gates in a NAND-realization of a Boolean function. Furthermore, the network need not be a tree; this may help reduce the network complexity further.

Most of these design optimization problems are likely to be NP-hard. However, as in the case of logic design of computer hardware, it should be possible to develop reasonable heuristics to generate "good" networks.

ACKNOWLEDGMENT

We have benefited from the comments given by the anonymous reviewers, and from the discussions with T. Eiter of Technical University of Vienna, M. Yamashita of Hiroshima University, M. L. Neilsen and M. Mizuno of Kansas State University, Y. Crama of Limburg University and H. Nagamochi of Kyoto University. In particular, a comment by T. Eiter has led to Theorem 4.4, and Remark 4.6 was provided by one of the anonymous reviewers.

REFERENCES

[1] D. Agrawal and A. El Abbadi, "Efficient solution to the distributed mutual exclusion problem," in *Proc. 8th ACM Symp. Principles of Distributed Computing,* Edmonton, Aug. 1989, pp. 193–200.

[2] R. L. Ashenhurst, "The decomposition of switching functions," in *Proc. Int. Symp. Theory of Switching,* Harvard Univ., 1957, pp. 74–116.

[3] D. Barbara and H. Garcia-Molina, "Mutual exclusion in partitioned distributed systems," *Distributed Comput.,* vol. 1, pp. 119–132, 1986.

[4] P. Bertolazzi and A. Sassano, "An $O(mn)$ time algorithm for regular set-covering problems," *Theoret. Comput. Sci.,* vol. 54, pp. 237–247, 1987.

[5] L. J. Billera, "On the composition and decomposition of clutters," *J. Combinatorial Theory,* vol. 11, pp. 234–245, 1971.

[6] Z. W. Birnbaum and J. D. Esary, "Modules of coherent systems," *SIAM J. Appl. Math.,* vol. 13, pp. 444–462, 1965.

[7] S. Y. Cheung, M. Ahamad, and M. H. Ammar, "Optimizing vote and quorum assignments for reading and writing replicated data," *IEEE Trans. Knowledge Data Eng.,* to be published.

[8] ——, "Multi-dimensional voting: A general method for implementing synchronization in distributed systems," Tech. Rep. GIT-ICS-89/35, School of Inform. Comput. Sci., Georgia Institute of Technology, 1989.

[9] Y. Crama, "Dualization of regular Boolean functions," *Discrete Appl. Math.,* vol. 16, pp. 79–85, 1987.

[10] T. Eiter and G. Gottlob, "Identifying the minimal transversals of a hypergraph and related problems," Tech. Rep. CD-TR91/16, Technische Universität Wien, 1991.

[11] A. Fu, "Enhancing concurrency and availability for database systems," Ph.D. dissertation, Simon Fraser Univ., 1990.

[12] H. Garcia-Molina and D. Barbara, "How to assign votes in a distributed system," *J. ACM,* vol. 32, pp. 841–860, Oct. 1985.

[13] H. Gifford, "Weighted voting for replicated data," in *Proc. 7th ACM Symp. Operat. Syst.,* Dec. 1979, pp. 150–162.

[14] M. A. Harrison, *Introduction to Switching and Automata Theory.* New York: McGraw-Hill, 1965.

[15] M. Herlihy, "Dynamic quorum adjustment for partitioned data," *ACM Trans. Database Syst.,* vol. 12, pp. 170–194, June 1987.

[16] T. Ibaraki and T. Kameda, "A theory of coteries," Tech. Rep. CSS/LCCR TR90-09, Lab. of Computer and Communications Research, Simon Fraser Univ., Aug. 1990.

[17] T. Ibaraki and T. Kameda, "A boolean theory of coteries," in *Proc. 3rd IEEE Symp. Parallel and Distributed Processing,* Dallas, TX, Dec. 1991, pp. 150–157.

[18] D. Kleitman and G. Markowsky, "On Dedekind's problem: The number of isotone Boolean functions," *Trans. AMS,* vol. 214, pp. 373–390, 1975.

[19] D. E. Knuth, *The Art of Computer Programming Vol. 1: Fundamental Algorithms.* Reading, MA: Addison-Wesley, 1973.

[20] A.D. Koršunov, "Solution of Dedekind's problem on the number of monotone Boolean functions," *Soviet Math. Dokl.,* vol. 18, pp. 442–445, 1977.

[21] M. Maekawa, "A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems," *ACM Trans. Comput. Syst.,* vol. 3, pp. 145–159, May 1985.

[22] R.H. Möhring, "Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and Boolean functions," *Algorithms and Software for Optimization, edited by C.L. Monma, Ann. Operat. Res.,* vol. 4, pp. 195–225, 1985.

[23] S. Muroga, *Threshold Logic and Its Applications.* New York: Wiley-Interscience, 1971.

[24] M. L. Neilsen and M. Mizuno, "Coterie join algorithm," *IEEE Trans. Parallel Distributed Syst.,* vol. 3, no. 5, pp. 582–590, Sept. 1992.

[25] M. Obradovic and P. Berman, "Voting as the optimal static pessimistic scheme for managing replicated data," in *Proc. 9th Symp. Reliable Distributed Syst.,* 1990, pp. 126–135.

[26] U. N. Peled and B. Simeone, "Polynomial-time algorithm for regular set-covering and threshold synthesis," *Discrete Appl. Math.,* vol. 12, pp. 57–69, 1985.

[27] K. G. Ramamurthy, "A new algorithm to find the smallest committee containing a given set of players," *Opsearch,* vol. 25, pp. 49–56, 1988.

[28] L.S. Shapley, "On committees," in *New Methods of Thought and Procedures,* F. Zwycky and A.G. Wilson, Eds. New York: Springer, 1967.

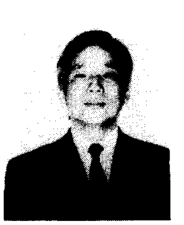
[29] D. Skeen, "A quorum-based commit protocol," in *Proc. 6th Berkeley Workshop Distributed Data Management and Computer Networks,* Feb. 1982, pp. 69–80.

[30] R. H. Thomas, "A majority consensus approach to concurrency control," *ACM Trans. Database Syst.,* vol. 4, pp. 180–209, June 1979.

[31] R. O. Winder, "Threshold logic asymptotes," *IEEE Trans. Comput.,* vol. C-19, pp. 349–353, 1970.

[32] S. Yajima and T. Ibaraki, "A lower bound on the number of threshold functions," *IEEE Trans. Electron. Comput.,* vol. EC-14, pp. 926–929, 1965.

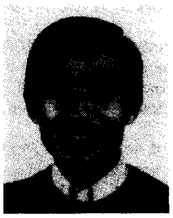
[33] Yu. A. Zuev, "Asymptotics of the logarithm of the number of threshold functions of the algebra of logic," *Soviet Math. Dokl.,* vol. 39, pp. 512–513, 1989.



Toshihide Ibaraki (S'63–M'68) received the B.E., M.E., and Dr. E. degrees from Kyoto University, Kyoto, Japan, in 1963, 1965, and 1970, respectively.

Since 1969, he has been with the Department of Applied Mathematics and Physics, Kyoto University, except for two years from 1983 to 1985, during which time he was with the Department of Computer and Information Sciences, Toyohashi University of Technology. Currently he is a Professor of Applied Mathematics and Physics. He has held a number of visiting appointments with University of Illinois, University of Waterloo, Simon Fraser University, and Rutgers University. His research interest includes combinatorial optimization, computational complexity, database systems, and algorithms.

Dr. Ibaraki is a member of the Association for Computing Machinery, the Mathematical Programming Society, the Operations Research Society of Japan, the Institute of Electronics, Information and Communication Engineers of Japan, the Information Processing Society of Japan, the Japan Association for Automatic Control Engineers, and the Japanese Society for Artificial Intelligence.



Tiko Kameda received the B.E. and M.E. degrees from the University of Tokyo in 1961 and 1963, respectively, and the Ph.D. degree from Princeton University in 1968.

From 1967 to 1980, he was with the Department of Electrical Engineering, University of Waterloo. Since 1981, he has been with Simon Fraser University, Burnaby, B.C., Canada, where he is a Professor of Computing Science and is the Director of Computer and Communications Research Laboratory. He has held a number of visiting appointments with Universities of Erlangen-Nürnberg, Bonn, Frankfurt, and Braunschweig, and Gesellschaft für Mathematik und Datenverarbeitung, all in Germany, and also with Kyoto University, Japan. His current research interests include database systems, combinatorial algorithms, distributed computing, and random testing of VLSI circuits.

Dr. Kameda is a member of the Association for Computing Machinery.