SIAM J. Discrete Math.
Vol. 14, No. 3, pp. 381–407

TRANSFORMATIONS ON REGULAR NONDOMINATED COTERIES AND THEIR APPLICATIONS*

KAZUHISA MAKINO† AND TIKO KAMEDA‡

Abstract. A coterie under an underlying set U is a family of subsets of U such that every pair of subsets has at least one element in common, but neither is a subset of the other. A coterie C under U is said to be *nondominated* (ND) if there is no other coterie D under U such that, for every $Q \in C$, there exists $Q' \in D$ satisfying $Q' \subseteq Q$.

We introduce the operation σ which transforms a ND coterie to another ND coterie. A *regular* coterie is a natural generalization of a *vote-assignable* coterie. We show that any regular ND coterie C can be transformed to any other regular ND coterie D by judiciously applying the σ operation to C at most $|C| + |D| - 2$ times.

As another application of the σ operation, we present an incrementally polynomial-time algorithm for generating all regular ND coteries. We then introduce the concept of a g-*regular* functional as a generalization of availability. We show how to construct an optimum coterie C with respect to a g-regular functional in $O(n^3|C|)$ time, where $n = |U|$. Finally, we discuss the structures of optimum coteries with respect to a g-regular functional.

Key words. coterie, nondominatedness, regular coterie, availability, mutual exclusion, positive self-dual Boolean function, regular self-dual Boolean function, g-regular functional

AMS subject classifications. 68M14, 68M15, 68P15, 68Q25, 68R05

PII. S0895480100371110

1. Introduction.

A *coterie* C under an underlying set $U = \{1, 2, \dots, n\}$ is a family of subsets (called *quorums*) of U satisfying the *intersection property* (i.e., for any pair $S, R \in C$, $S \cap R \neq \emptyset$ holds) and *minimality* (i.e., no quorum in C contains any other quorum in C) [18, 23]. The concept of a coterie has applications in diverse areas, such as mutual exclusion in distributed systems [13, 18, 23], data replication protocols [14], name servers [27], selective dissemination of information [39], and distributed access control and signatures [30].

For example, to achieve mutual exclusion in a distributed system, let the elements in U represent the sites in the distributed system. A process is allowed to enter a critical section only if it can get permissions from all the members of a quorum $Q \in C$, where each site is allowed to issue at most one permission at a time. By the intersection property, it is guaranteed that at most one process can enter the critical section at any time.

A coterie C under U is said to *dominate* another coterie D ($\neq C$) under U if, for each quorum $Q \in D$, there is a quorum $Q' \in C$ satisfying $Q' \subseteq Q$. A coterie which is not dominated by any other coterie is said to be *nondominated* (ND) [18]. ND coteries are important in practical applications, since they have maximal "efficiency" in some sense [4, 18, 21].

*Received by the editors April 17, 2000; accepted for publication (in revised form) June 6, 2001; published electronically August 29, 2001. An extended abstract of this paper appears in *Proceedings of the Nineteenth ACM Symposium on Principles of Distributed Computing* (*PODC* 2000), Portland, OR, 2000, pp. 279–288. This work was supported in part by the Scientific Grant in Aid, by the Ministry of Education, Science, Sports, and Culture of Japan, and in part by the Natural Sciences and Engineering Research Council of Canada.

http://www.siam.org/journals/sidma/14-3/37111.html

†Division of Systems Science, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531, Japan (makino@sys.es.osaka-u.ac.jp)

‡School of Computing Science, Faculty of Applied Sciences, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada (tiko@cs.sfu.ca)

Given a family C of subsets of U, which is not necessarily a coterie, we define a *positive* (i.e., monotone) Boolean function f_C such that $f_C(x) = 1$ if the Boolean vector $x \in \{0,1\}^n$ is greater than or equal to the characteristic vector of some subset[1] in C, and 0 otherwise, where $n = |U|$. It was shown in [20] that C is a coterie if and only if f_C is *dual-minor*, and C is ND if and only if f_C is *self-dual*. (See section 2.2.) Based on this characterization, the methods developed in the rich field of Boolean functions can be exploited to derive various properties of coteries and ND coteries.

A coterie C is said to be *vote-assignable* if there exist a *vote assignment* $w : U \mapsto \mathbb{R}^+$ and a *threshold* $t \in \mathbb{R}^+$ such that $w(S) \geq t$ if and only if $S \supseteq Q$ for some $Q \in C$ [18, 19, 37], where $\mathbb{R}^+$ is the set of nonnegative real numbers and $w(S) = \sum_{i \in S} w(i)$. It is easy to see that there is a one-to-one correspondence between vote-assignable coteries (resp., ND coteries) C and dual-minor (resp., self-dual) threshold Boolean functions f_C. (For the definition of a threshold Boolean function, see section 2.) The vote-assignable coteries are important and have been used in many practical problems, since they can be handled efficiently (see, e.g., [18, 19, 37, 38]). We assume in this paper that a vote assignment w satisfies $w(i) \geq w(j)$ for all $i < j$, since we are interested in coteries which are nonequivalent under permutation on U. A coterie C is *equivalent to* a coterie C' *under permutation* if C can be transformed into C' by permuting the elements of U. For example, $C = \{\{1,2\},\{1,3\}\}$ is equivalent to $C' = \{\{2,3\},\{2,1\}\}$ under permutation. A coterie C is said to be *regular* if, for each $Q \in C$ and every pair $(i,j) \in U \times U$ with $i < j$, $i \notin Q$ and $j \in Q$, there exists $Q' \in C$ such that $Q' \subseteq (Q \setminus \{j\}) \cup \{i\}$.[2] By definition (and the discussion in section 2), a vote-assignable coterie C is always regular, though in general the converse is not true. The regular Boolean functions were defined as a generalization of the threshold functions [28]. It is known that most regular coteries are vote-assignable [28]; in particular, all regular ND coteries under U with $n = |U| \leq 9$ are vote-assignable.

Among the important problems regarding coteries are the following:

(i) decide whether a given coterie is ND (equivalently, whether a given positive dual-minor function is self-dual);

(ii) construct "optimal" ND coteries according to a certain criterion, such as availability and load [29] (equivalently, construct "optimal" positive self-dual functions); and

(iii) generate all ND coteries (equivalently, all positive self-dual functions) systematically.

Unfortunately, the complexity of problem (i) is still unknown [8, 16, 22], although a result by Fredman and Khachiyan [17] suggests that it is unlikely that the problem is NP-hard. [8, 16] give a number of interesting equivalent problems which arise in various fields of applications. However, it is known that if we restrict ourselves to regular coteries, (i) is polynomially solvable [6, 32].

Although (i) is an interesting problem, we do not consider (i) further in this paper. Instead, we focus on problems (ii) and (iii). As for (ii), let us consider the availability of a coterie, where the concept of availability has been extensively studied under different names in reliability theory (see, e.g., [35]). Assume that each element can be in either of two states, *operational* or *inoperational*, and takes on its state randomly and independently, element i being operational (resp., inoperational) with probability p_i (resp., $1 - p_i$). Given operational probabilities p_i, $i \in U$, where we assume without loss of generality that $1 \geq p_1 \geq p_2 \geq \cdots \geq p_n \geq 0$, the *availability* of a coterie C is

[1] The ith component of the characteristic vector is 1 (0) if $i \in U$ is (not) contained in the subset.

[2] This definition was motivated by the definition of *regular* Boolean functions. See section 2.3.

the probability that the set of operational elements contains at least one quorum in C. Availability is undoubtedly an important concept in practical applications, and hence it is natural to construct a coterie with the maximum availability.

The availability of coteries has been studied extensively [1, 5, 15, 33, 36, 38]. It is known [1, 36] that the elements $i \in U$ with $p_i < 1/2$ can be ignored; i.e., there exists a maximum-availability coterie C such that no quorum in C contains i. (In the case where $p_i < 1/2$ holds for all i, $C = \{\{1\}\}$ has the maximum availability [1, 15, 33].) Thus, we shall assume that

$$(1 \geq)\ p_1 \ \geq\ p_2 \ \geq\ \cdots\ \geq\ p_n \ \geq\ 1/2.$$

It is also known that if either $p_1 = 1$ or $p_1 \leq 1/2$, then $C = \{\{1\}\}$ has the maximum availability. If $1 \neq p_1 > 1/2$, on the other hand, it is demonstrated in [36, 38] that the coterie C_{max}, given below, maximizes availability. First define the weight for $i \in U$ by

$$w^*(i) = \log_2(p_i/(1 - p_i)) \tag{1}$$

and introduce the notation $w^*(S) = \sum_{i \in S} w^*(i)$ for $S \subseteq U$. Now, $Q \in C_{max}$ if

(a) $w^*(Q)$ $(= w^*(U \setminus Q)) = w^*(U)/2$ and $1 \in Q$ (1 is an element of U), or

(b) Q is a minimal subset of U satisfying $w^*(Q) > w^*(U)/2$, and Q does not contain any quorum of type (a).

Since this coterie C_{max} is vote-assignable, Amir and Wool [1], Spasojevic and Berman [36], and Tong and Kain [38] proposed algorithms to compute a vote assignment w from w^*, called *tie-breaking* algorithm, in order to remove case (a). An exponential algorithm is proposed in [38] to find the "optimal" tie-breaking rule, while Amir and Wool [1] and Spasojevic and Berman [36] present polynomial-time approximation algorithms for it. The main problem with the above definition of C_{max} is that there may exist a subset $S \subseteq U$ such that $w^*(S) = w^*(U \setminus S)$ (case (a)), because of which a simple vote assignment w (showing that C_{max} is vote-assignable) is not easily obtainable, and that the weight $w^*(i)$ is, in general, not a rational number; hence we cannot compute $w^*(S) = \sum_{i \in S} w^*(i)$ in polynomial time. For the above reasons, no polynomial algorithm for constructing maximum-availability coteries was known. In this paper, we present a polynomial-time algorithm for it. More precisely, we define a "g-regular" functional as a generalization of availability (see section 6) and then show that, given a g-regular functional Φ, we can compute a coterie C which maximizes Φ in $O(n^3|C|)$ time, where $|C|$ is the number of quorums in C.

Problem (iii) is known to be useful to solve (ii) [9, 18]. To solve (ii), we first enumerate all (or some) ND coteries efficiently and select the best one under a certain criterion, which is not easily computable. This procedure is useful when n is small or when we have enough time to compute it. We feel that (iii) is mathematically interesting, giving us an insight into the structure of ND coteries (or, equivalently, self-dual Boolean functions).

The generation of all ND coteries in a certain subclass of vote-assignable ND coteries was discussed in [28], which is used to give a lower bound on the number of all vote-assignable ND coteries. However, the procedure is not polynomial and computes a proper subclass of vote-assignable ND coteries. Garcia-Molina and Barbara [18] proposed an algorithm to generate all ND coteries in a certain superclass of regular ND coteries. However, it is also not polynomial. Bioch and Ibaraki [9] later came up with a polynomial-time algorithm to generate all ND coteries, and compiled a list containing all ND coteries of up to seven elements, which are essentially different (i.e., nonequivalent under permutation). We remark here that their algorithm is not polynomial if equivalent duplicates are to be deleted from the output. In fact, they compiled a list of all ND coteries under seven or fewer elements by first running their algorithm and then selecting nonequivalent representatives from among them. In this paper, we present a polynomial algorithm to generate all *regular* ND coteries. Since no regular ND coterie C is equivalent to any other regular ND coterie C' ($\neq C$) under permutation (see Lemma 2.2), our algorithm does not output ND coteries which are equivalent under permutation. Although our algorithm outputs only regular ND coteries, it is practically useful because all ND coteries under $n = 5$ or fewer elements are all regular (if we consider their representatives), and when n is relatively small, a large fraction of ND coteries are regular [28]. Moreover, if the objective function of (ii) cited above is g-regular (e.g., the availability of a coterie), then we can restrict our attention to regular coteries.

After defining necessary terminology in section 2 (we use Boolean terminology, which is simpler than that of set theory), we discuss in section 3 two operations, named ρ and σ, which transform the positive self-dual function f (representing a ND coterie) into another positive self-dual function (representing another ND coterie) by making a minimal change in the set of minimal true vectors of f. The ρ operation was introduced in [9], and σ was implicitly introduced in [18], where it is called *coterie transformation*.

Section 4 shows that any regular self-dual function f (representing a regular ND coterie) can be transformed into any other regular self-dual function g (representing any other regular ND coterie) by judiciously applying the σ operation to f at most $|\min T(f)| + |\min T(g)| - 2$ times. (For the definition of $\min T(f)$, see section 2.) In sections 5 and 6, we consider the problems of generating all regular self-dual functions and of computing an optimal self-dual function with respect to a g-regular functional Φ (see the definition of g-regularity in section 6) as applications of the above transformation.

In addition to the theory of coteries, the concepts of self-duality and regularity play important roles in diverse areas such as computational learning theory (e.g., identification of positive Boolean functions [8, 10, 24, 25]), threshold logic [28], operations research [6, 11, 31, 32], clutters in set theory [7], minimal transversals in hypergraphs [16], and coherent systems of reliability theory [35]. The results of this paper are relevant to all these problems.

2. Definitions and basic properties.

A *Boolean function*, or a *function* in short, of n variables is a mapping $f : \{0,1\}^n \to \{0,1\}$, where $v \in \{0,1\}^n$ is called a *Boolean vector* (a *vector* in short). If $f(v) = 1$ (resp., 0), then v is called a *true* (resp., *false*) vector of f. The set of all true vectors (resp., false vectors) of f is denoted by $T(f)$ (resp., $F(f)$). Throughout this paper, the constant functions with $T(f) = \emptyset$ (empty) and $F(g) = \emptyset$ are denoted by $f = \perp$ and $g = \top$, respectively. For any two functions f and g, we write $f \leq g$ if $T(f) \subseteq T(g)$. For a vector $v = (v_1, v_2, \ldots, v_n)$, we define $ON(v) = \{j \mid v_j = 1\}$ and $OFF(v) = \{j \mid v_j = 0\}$.

The argument x of function f is represented as a vector $x = (x_1, x_2, \ldots, x_n)$, where each x_i is a Boolean *variable*. A variable x_i is said to be *relevant* if there exist two vectors v and w such that $f(v) \neq f(w)$, $v_i \neq w_i$, and $v_j = w_j$ for all $j \neq i$; otherwise, it is said to be *irrelevant*. The set of all relevant variables of a function f is denoted by $V_f \subseteq V = \{x_1, x_2, \ldots, x_n\}$. A *literal* is either a variable x_i or its complement $\overline{x}_i$, which are referred to as a *positive* or *negative* literal, respectively. The *complement* of vector $x = (x_1, x_2, \ldots, x_n)$ is defined by $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$. A *term* t is a conjunction $(\bigwedge_{i\in P(t)} x_i) \wedge (\bigwedge_{j\in N(t)} \overline{x}_j)$ of literals such that $P(t), N(t) \subseteq \{1, 2, \ldots, n\}$ and $P(t) \cap N(t) = \emptyset$. For example, $t_1 = x_1x_4\overline{x}_5x_6$ is a term, while $t_2 = x_2x_4\overline{x}_2$ is not. In particular, the term t with $P(t) = N(t) = \emptyset$ represents $\top$. A *disjunctive normal form* (DNF) is a disjunction of distinct terms. It is easy to see that any function f can be represented in DNF whose variable set is V_f. We sometimes do not distinguish a formula (e.g., DNF) from the function it represents if no confusion arises.

2.1. Positive functions. For a pair of vectors $v, w \in \{0,1\}^n$, we write $v \leq w$ if $v_j \leq w_j$ holds for all $j \in V$, and $v < w$ if $v \leq w$ and $v \neq w$, where we define $0 < 1$. For a set of vectors $S \subseteq \{0,1\}^n$, $\min_{\geq} S$ (resp., $\max_{\geq} S$) denotes the set of all minimal (resp., maximal) vectors in S with respect to $\geq$. For example, for a function f, $\min_{\geq} T(f)$ denotes the set of all minimal true vectors of f, and $\max_{\geq} F(f)$ denotes the set of all maximal false vectors of f. We sometimes use $\min S$ (resp., $\max S$) instead of $\min_{\geq} S$ (resp., $\max_{\geq} S$) if no confusion arises. A function f is said to be *positive* or *monotone* if $v \leq w$ always implies $f(v) \leq f(w)$. A *prime implicant* of a function f is a term (i.e., monomial) t such that $t \leq f$, but $t' \not\leq f$ for any proper subterm t' of t. There is a one-to-one correspondence between $\min T(f)$ and the set of all prime implicants of f such that a vector v corresponds to the term t_v defined by $t_v = x_{i_1}x_{i_2}\cdots x_{i_k}$ if $v_{i_j} = 1, j = 1, 2, \ldots, k$, and $v_i = 0$ otherwise. For example, the vector $v = (1010)$ corresponds to the term $t_v = x_1x_3$. In particular, if $v = (00\cdots 0)$, then $t_v = \top$. Note that $t_v \leq t_w$ (as functions) holds if and only if $v \geq w$. We also use the notation $t_{\overline{v}}$ to denote the term $x_{j_1}x_{j_2}\cdots x_{j_l}$, where $\{j_1, j_2, \ldots, j_l\} = \{1, 2, \ldots, n\} \setminus \{i_1, i_2, \ldots, i_k\}$. For the above $v = (1010)$, we have $t_{\overline{v}} = x_2x_4$.

It is known that a positive function f is uniquely determined by $\min T(f)$ (hence a positive function f can be represented by a string of length $\leq n|\min T(f)|$) and that f has the unique *minimal disjunctive normal form* (MDNF), consisting of all the prime implicants of f, where $N(t) = \emptyset$ for each prime implicant t. In this paper, we sometimes represent the MDNF of a positive function such as $f = x_1x_2 + x_2x_3 + x_3x_1$ in a simplified form $f = 12 + 23 + 31$, using only the subscripts of the literals. The set of minimal true vectors of this function is $\min T(f) = \{(110), (011), (101)\}$ if f is a 3-variable function. Coteries can be conveniently modeled by Boolean functions based on the fact that $\min T(f)$ can represent a family of subsets, none of which includes the other. For example, the above $\min T(f)$ represents a coterie $C = \{\{1,2\}, \{2,3\}, \{3,1\}\}$, while $T(f)$ represents the family of all subsets that contain a member of C.

2.2. Dual-comparable functions. The *dual* of a function f, denoted f^d, is defined by

$$f^d(x) = \overline{f}(\overline{x}),$$

where $\overline{f}$ and $\overline{x}$ denote the complement of f and x, respectively. As is well known, f^d is obtained from f by interchanging $+$ (OR) and $\cdot$ (AND), as well as the constants 0 and 1. Recall that for any two functions f and g, we write $f \leq g$ if $T(f) \subseteq T(g)$, and $f < g$ if $f \leq g$ and $f \neq g$. We say that f *is covered by* g if $f \leq g$. It is easy to see that $(f+g)^d = f^dg^d$, $(fg)^d = f^d + g^d$, $f \leq g$ if and only if $f^d \geq g^d$, and so on. A function is called *dual-minor* if $f \leq f^d$, *dual-major* if $f \geq f^d$, and *self-dual* if $f = f^d$. It is known [20] that

1. f is dual-minor if and only if at most one of v and $\overline{v}$ belongs to $T(f)$ for any $v \in \{0,1\}^n$;

2. f is dual-major if and only if at least one of v and $\overline{v}$ belongs to $T(f)$ for any $v \in \{0,1\}^n$; and
3. f is self-dual if and only if exactly one of v and $\overline{v}$ belongs to $T(f)$ for any $v \in \{0,1\}^n$.

For example, $f = 123$ is dual-minor since $f^d = 1 + 2 + 3$ satisfies $f \leq f^d$. The dual of $f = 12 + 23 + 31$ is

$$f^d = (1+2)(2+3)(3+1) \;=\; 12 + 23 + 31.$$

This function f is self-dual and is called the *basic majority function*; it is known to be the only positive self-dual function of three relevant variables. There is no positive self-dual function of exactly two relevant variables. However, each function $f = x_i$ is a positive self-dual function of one variable.

If f is positive, then f^d is also positive. In this case, an alternative definition of f^d is given by the condition that $v \in T(f^d)$ if and only if v is a *transversal* of $\min T(f)$; i.e., it satisfies $ON(v) \cap ON(w) \neq \emptyset$ for all $w \in \min T(f)$. Let

$\mathcal{C}_{SD}(n)$: the class of all positive self-dual functions of n variables,
$\mathcal{C}_{DMA}(n)$: the class of all positive dual-major functions of n variables,
$\mathcal{C}_{DMI}(n)$: the class of all positive dual-minor functions of n variables.

Note that in these definitions functions may have some irrelevant variables.

2.3. Regular, 2-monotonic, and threshold functions. A positive function f is said to be *regular* if, for every $v \in \{0,1\}^n$ and every pair (i,j) with $i < j$, $v_i = 0$ and $v_j = 1$, the following condition holds:

$$f(v) \leq f(v + e^{(i)} - e^{(j)}), \tag{2}$$

where $e^{(k)}$ denotes the unit vector which has a 1 in its kth position and 0 in all other positions.

In order to define an important partial order on $\{0,1\}^n$, we first define the concept of the *profile* of a vector $v \in \{0,1\}^n$ as follows:

$$prof_v(k) = \sum_{j \leq k} v_j,$$

where $k = 1, 2, \ldots, n$. If $v, w \in \{0,1\}^n$, where $v \neq w$, satisfy $prof_v(k) \leq prof_w(k)$ for all k, then we write $v \prec w$ (or $w \succ v$), and we say that w *majorizes* v. If $v \prec w$ or $v = w$, then we write $v \preceq w$ (or $w \succeq v$). It is helpful to visualize the profile as in Figure 2.1.

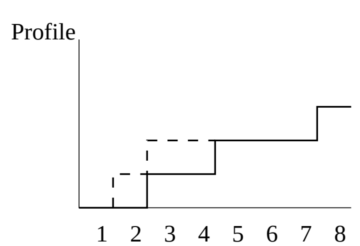

FIG. 2.1. *The profiles $prof_v(k)$ (solid lines) and $prof_w(k)$ (dashed lines).*

In Figure 2.1 the profiles of $v = (00101001)$ and $w = (01100001)$ are depicted in the solid staircase and dashed staircase, respectively. The dashed staircase is not visible where it overlaps with the solid staircase. It is easy to see that if v is majorized by w, then the profile of w does not go below the profile of v.

It is clear from the above definition that $v \prec w$ if and only if $\overline{v} \succ \overline{w}$, since $prof_{\overline{v}}(k) = k - prof_v(k)$. Note that $v \leq w$ implies $v \preceq w$, but the converse is not always true. A function f is said to be *profile-monotone* if $v \prec w$ implies $f(v) \leq f(w)$. The following lemma is proved in [28].

LEMMA 2.1 (see Muroga [28]). *A function f is regular if and only if f is* profile-monotone.

For two functions f and g, we say that f is *equivalent to g under permutation* if permuting variables of f produces g.

LEMMA 2.2. *Two different regular functions are not equivalent under permutation.*

Proof. Let f and g be regular functions such that g can be obtained from f by a permutation π, where we regard π as the permutation on indices; i.e., we write $\pi(i) = j$ instead of $\pi(x_i) = x_j$. We claim that $f = g$, which proves the lemma. Let i and j be indices satisfying $i < j$ and $\pi(i) > \pi(j)$. Note that if there exist no such indices, then π is the identity permutation, implying $f = g$. By the regularity of f, we have

$$f(v) \leq f(v + e^{(i)} - e^{(j)}) \tag{3}$$

for every $v \in \{0,1\}^n$ with $v_i = 0$ and $v_j = 1$, and by the regularity of g, we have $g(w) \leq g(w + e^{\pi(j)} - e^{\pi(i)})$ for every $w \in \{0,1\}^n$ with $w_{\pi(j)} = 0$ and $w_{\pi(i)} = 1$, that is,

$$f(v) \geq f(v + e^{(i)} - e^{(j)}) \tag{4}$$

for every $v \in \{0,1\}^n$ with $v_i = 0$ and $v_j = 1$. By combining (3) and (4),

$$f(v) = f(v + e^{(i)} - e^{(j)})$$

holds for every $v \in \{0,1\}^n$ with $v_i = 0$ and $v_j = 1$. This means that f is symmetric in variables x_i and x_j. Namely, the function f' obtained from f by the permutation

$$\pi'(k) = \begin{cases} i & \text{if } k = j, \\ j & \text{if } k = i, \\ k & \text{otherwise} \end{cases}$$

is identical to f (i.e., $f' = f$). Since π can be obtained by a concatenation of such permutations π', it follows by induction that $f = g$. ∎

For a set of vectors $S \subseteq \{0,1\}^n$, $\min_{\succeq} S$ (resp., $\max_{\succeq} S$) denotes the set of all minimal (resp., maximal) vectors in S with respect to $\succeq$. For any set of vectors $S \subseteq \{0,1\}^n$, we have $\min_{\succeq} S \subseteq \min S \,(= \min_{\geq} S)$ and $\max_{\succeq} S \subseteq \max S \,(= \max_{\geq} S)$, since $v \geq w$ implies $v \succeq w$. In particular, we have $\min_{\succeq} T(f) \subseteq \min T(f)$; i.e., any element of $\min T(f) \setminus \min_{\succeq} T(f)$ majorizes an element of $\min_{\succeq} T(f)$. It follows from Lemma 2.1 that a regular function f is uniquely determined by $\min_{\succeq} T(f)$.

A positive function f is called 2*-monotonic* if there exists a linear ordering on V for which f is regular. The 2-monotonicity and related concepts have been studied in various contexts in fields such as threshold logic [6, 12, 28, 32], game theory

[35], hypergraph theory [11], and learning theory [10, 24, 25]. The 2-monotonicity was originally introduced in conjunction with threshold functions (e.g., [28]), where a positive function f is a *threshold* function if there exist n nonnegative real numbers (weights) $w_1, w_2, \ldots, w_n$ and a nonnegative real number (threshold) t such that

$$f(x) = \begin{cases} 1 & \text{if } \sum w_i x_i \geq t, \\ 0 & \text{if } \sum w_i x_i < t. \end{cases}$$

As this f satisfies (2) by permuting variables so that $w_i > w_j$ implies $i < j$, a threshold function is always 2-monotonic, although the converse is not true [28].

It is known [18] that there are $\Omega(2^{2^{cn}})$ self-dual functions, where c denotes some positive constant, but only $O(2^{n^2})$ threshold self-dual functions. It is not known if 2-monotonic self-dual functions are substantially more than threshold self-dual functions.

3. The operations $\boldsymbol{\rho}$ and $\boldsymbol{\sigma}$. Let f be a positive function of n variables. Throughout this paper, we assume that f is *nontrivial* in the sense that $f \neq \bot, \top$ and $n \geq 1$. Given a vector $v \in \min T(f)$, the operation ρ_v applied to f removes v from $T(f)$ and then adds $\overline{v}$ to $T(f)$ [9]. More precisely, while adding $\overline{v}$, all the vectors larger than $\overline{v}$ are also added to $T(f)$. Therefore,

$$T(\rho_v(f)) = (T(f) \setminus \{v\}) \cup T_{\geq}(\overline{v}), \tag{5}$$

where

$$T_{\geq}(\overline{v}) = \{w \in \{0,1\}^n \mid w \geq \overline{v}\}.$$

An equivalent definition is

$$\rho_v(f) = f_{\setminus v} + t_{\overline{v}} + t_v t_{\overline{v}}^d, \tag{6}$$

where $f_{\setminus v}$ denotes the function defined by all the prime implicants of f except t_v, and $t_{\overline{v}}^d$ denotes the dual of $t_{\overline{v}}$. We note that if $t_v = x_{i_1} x_{i_2} \cdots x_{i_k}$ and $t_{\overline{v}} = x_{j_1} x_{j_2} \cdots x_{j_l}$, then

$$t_v t_{\overline{v}}^d = x_{i_1} x_{i_2} \cdots x_{i_k} (x_{j_1} + x_{j_2} + \cdots + x_{j_l})$$

represents all the vectors larger than v. As seen in Example 3.1, the expression (6) is not necessarily in MDNF, even if $f_{\setminus v}$ is represented by its MDNF, because some of the prime implicants in $t_{\overline{v}} + t_v t_{\overline{v}}^d$ may cover or may be covered by some prime implicants in $f_{\setminus v}$.

Let us note that the operation ρ is essentially the same as the coterie transformation (CT) in [18] except that CT assumes the following additional conditions: (i) $|OFF(v)| \geq 2$, and (ii) there is at least one prime implicant in $t_v t_{\overline{v}}^d$ which is not covered by $f_{\setminus v}$. In this sense, CT is a special case of the ρ operation.

Given a vector $v \in \min T(f)$ and a variable set I with $V_f \subseteq I \subseteq V$, we define the operation $\sigma_{(v;I)}$ by

$$\sigma_{(v;I)}(f) = f_{\setminus v} + t_{\overline{v[I]}} + t_{v[I]} t_{\overline{v[I]}}^d, \tag{7}$$

where $v[I]$ denote the *projection* of v on I; e.g., if $v = (1100)$, $I_1 = \{x_1, x_2, x_3\}$, and $I_2 = \{x_2, x_3\}$, then $v[I_1] = (110)$ and $v[I_2] = (10)$. By definition, we have $\sigma_{(v;V)} = \rho_v$. This operation $\sigma_{(v;I)}$ is implicitly used in [18].

Example 3.1. Consider a positive function of $n = 7$ variables,

$$f = 12 + 13 + 145 + 234 + 235.$$

For this function, we have $V_f = \{1, 2, 3, 4, 5\}$.[3] For $v = (1100000)$ and $w = (0111000)$, we show below how operations ρ and σ are applied.

$$\begin{aligned}
\rho_v(f) &= 13 + 145 + 234 + 235 + \mathbf{34567} + 12(\mathbf{3} + \mathbf{4} + \mathbf{5} + \mathbf{6} + \mathbf{7}) \\
&= 124 + 125 + 126 + 127 + 13 + 145 + 234 + 235 + 34567, \\
\rho_w(f) &= 12 + 13 + 145 + 235 + \mathbf{1567} + 234(\mathbf{1} + \mathbf{5} + \mathbf{6} + \mathbf{7}) \\
&= 12 + 13 + 1567 + 2346 + 2347 + 235, \\
\sigma_{(v;V_f)}(f) &= 13 + 145 + 234 + 235 + \mathbf{345} + 12(\mathbf{3} + \mathbf{4} + \mathbf{5}) \\
&= 124 + 125 + 13 + 145 + 234 + 235 + 345, \\
\sigma_{(w;V_f)}(f) &= 12 + 13 + 145 + 235 + \mathbf{15} + 234(\mathbf{1} + \mathbf{5}) \\
&= 12 + 13 + 15 + 235.
\end{aligned}$$

Let f be a function on the variable set $V = \{1, 2, \ldots, n\}$. For a variable set $I \subseteq V$, the *projection* of f on I, denoted by $Proj_I(f)$, is the function on I obtained from f by fixing $x_i = 0$ for all $x_i \in V \setminus I$, i.e.,

$$Proj_I(f)(x_1, x_2, \ldots, x_{|I|}) = f(x_1, x_2, \ldots, x_{|I|}, 0, 0, \ldots, 0)$$

if $I = \{x_1, x_2, \ldots, x_{|I|}\}$. For a variable set $J \supseteq V$, the *expansion* of f to J, denoted by $Exp_J(f)$, is the function on J obtained from f by adding irrelevant variables $x_i \in J \setminus V$. By definition, f and its expansion can be represented by the same DNF.

For $I \supseteq V_f$, we have

$$\sigma_{(v;I)}(f) = Exp_V(\rho_{v[I]}(Proj_I(f))). \tag{8}$$

Thus σ has properties similar to those of ρ. See, for example, Theorem 3.2 below.

Now, for a specified class $\mathcal{C}(n)$ of positive functions of n variables, we say that ρ (resp., σ) *preserves* $\mathcal{C}(n)$ if $\rho_v(f) \in \mathcal{C}(n)$ holds for all $f \in \mathcal{C}(n)$ and $v \in \min T(f)$ (resp., $\sigma_{(v;I)}(f) \in \mathcal{C}(n)$ holds for all $f \in \mathcal{C}(n)$, $v \in \min T(f)$, and $I \supseteq V_f$).

THEOREM 3.2. *The operations ρ and σ defined above preserve the classes $\mathcal{C}_{SD}(n)$, $\mathcal{C}_{DMA}(n)$, and $\mathcal{C}_{DMI}(n)$.*

Proof. This theorem is proved for ρ in [9]. Consider any function $f \in \mathcal{C}_{SD}(n)$ and any set I satisfying $V_f \subseteq I \subseteq V$. If $f = f^d$, then clearly $Proj_I(f) = Proj_I(f^d)$. We thus have $Proj_I(f) \in \mathcal{C}_{SD}(|I|)$, and hence $\rho_v(Proj_I(f)) \in \mathcal{C}_{SD}(|I|)$ by the above-cited result in [9]. It is clear that, for any $g \in \mathcal{C}_{SD}(|I|)$, we have $Exp_V(g) \in \mathcal{C}_{SD}(n)$. Thus by (5), $\sigma_{(v;I)}$ preserves $\mathcal{C}_{SD}(n)$, similarly for $\mathcal{C}_{DMA}(n)$ and $\mathcal{C}_{DMI}(n)$. ∎

Note that if f is self-dual, then $\rho_v(f)$, $v \in \min T(f)$, is specified simply by

$$T(\rho_v(f)) = (T(f) \setminus \{v\}) \cup \{\overline{v}\}, \tag{9}$$

i.e., by interchanging v with $\overline{v}$ in $T(f)$. This follows from (5) and the fact that $\rho_v(f) \in \mathcal{C}_{SD}(n)$, hence $|T(\rho_v(f)| = |T(f)| = 2^{n-1}$. To see the effect of $\sigma_{(v;I)}$ on $T(f)$, where $V_f \subseteq I \subseteq V$, define

$$v[I]\underline{*} = \{u \in \{0,1\}^n \mid u[I] = v[I]\}.$$

[3]We sometimes represent a variable set as an index set; e.g., $\{x_1, x_2\}$ is represented as $\{1, 2\}$.

It is easy to see that

$$T(\sigma_{(v;I)}(f)) = (T(f) \setminus v[I]\underline{*}) \cup \overline{v}[I]\underline{*}. \tag{10}$$

To see the difference between (9) and (10), refer to Example 3.1, where $I = V_f$.

Now consider a sequence of transformations from a positive self-dual function f to another positive self-dual function g,

$$\begin{array}{ccccccccc} f_0\,(=f) & \longrightarrow & f_1 & \longrightarrow & f_2 & \longrightarrow & \dots & \longrightarrow & f_{m_1}\,(=g), \\ g_0\,(=f) & \longrightarrow & g_1 & \longrightarrow & g_2 & \longrightarrow & \dots & \longrightarrow & g_{m_2}\,(=g), \end{array}$$

where $f_{i+1} = \rho_{v^{(i)}}(f_i)$, $v^{(i)} \in \min T(f_i)$, $g_{i+1} = \sigma_{(w^{(i)};I_i)}(g_i)$, $w^{(i)} \in \min T(g_i)$, and $I_i \supseteq V_{g_i}$. We can see that $m_1, m_2 \geq |\min T(f) \setminus \min T(g)|$ and $m_1 \geq |T(f) \setminus T(g)|$. The latter implies that m_1 might be exponential in n and $|\min T(f)|$, while m_2 might be small. In the next section, we consider the ρ and σ operations for regular self-dual functions, and give a transformation algorithm between any two regular self-dual functions f and g of n variables, which satisfy

$$m_2 \leq |\min T(f)| + |\min T(g)| - 2.$$

4. Transformation of regular self-dual functions. The goal of this section is to present an efficient algorithm, TRANS-REG-SD, which transforms a given regular self-dual function f to the one-variable regular self-dual function $g = x_1$. It applies a sequence of σ operations to f, generating a sequence of regular self-dual functions in the process. As we will show, this algorithm can be used to transform a given regular self-dual function of n variables to any other regular self-dual function of n variables, some of which may be irrelevant. We need to prove a number of lemmas to achieve this goal.

We start with the following lemma, which shows that ρ_v preserves profile-monotonicity (i.e., regularity) if v satisfies a certain condition. (We have already seen that ρ_v preserves self-duality.)[4] Recall that $\rho_v(f)$ is specified by (9), and therefore in the proof we concentrate on the vectors v and $\overline{v}$.

LEMMA 4.1. *Let f be a regular self-dual function, and let $v \in \min T(f)$. $\rho_v(f)$ is regular if and only if $v \in \min_{\succeq} T(f)$ and $\overline{v} \not\prec v$.*

Proof. By definition, $\rho_v(f)$ is regular (i.e., profile-monotone) if

$$\rho_v(f)(u) \leq \rho_v(f)(w) \text{ for any } u \prec w. \tag{11}$$

Let us first consider the only-if part. Recall that $\rho_v(f)$ can be specified by (9). Thus we have $\rho_v(f)(v) = 0$ and $\rho_v(f)(\overline{v}) = 1$, which, together with (11), implies $\overline{v} \not\prec v$. Moreover, since $v \in \min T(f)$, if $v \notin \min_{\succeq} T(f)$, then there exists a vector $u \in \min_{\succeq} T(f)$ majorized by v, i.e., $u \prec v$. (See the paragraph after Lemma 2.1.) Now we have $\rho_v(f)(v) = 0$ and $\rho_v(f)(u) = 1$, which contradicts (11) with $w = v$. Thus $v \in \min_{\succeq} T(f)$ holds.

We now turn to the proof of the if part and show that $\rho_v(f)$ is profile-monotone. Equation (11) clearly holds if $u, w \notin \{v, \overline{v}\}$, since f is profile-monotone. (See (9).) If $u = v$ in (11), then the left-hand side becomes $\rho_v(f)(v) = 0$ by (9), and (11) holds for any w. Similarly, if $w = \overline{v}$ in (11), then the right-hand side becomes $\rho_v(f)(\overline{v}) = 1$ by (9), and (11) holds for any u. Now assume that $u = \overline{v}$ $(\prec w)$, in which case

[4]As we commented before, the ρ operation is a special case of the σ operation.

$w \neq v$ by the condition in the lemma. Then we have $v \succ \overline{w}$, and $v \in \min_{\succeq} T(f)$ implies $f(\overline{w}) = 0$, hence $f(w) = \rho_v(f)(w) = 1$. Thus (11) holds. Finally, assume that $w = v$ ($\succ u$), in which case $u \neq \overline{v}$ by the condition in the lemma. $v \in \min_{\succeq} T(f)$ implies $f(u) = 0$, hence $\rho_v(f)(u) = 0$. Thus (11) again holds. □

The following lemma shows how to choose v to be used in $\rho_v(f)$ to guarantee that $\rho_v(f)$ is regular.

LEMMA 4.2. *Let f be a regular self-dual function of $n\,(\geq 2)$ variables. If $v \in \min_{\succeq} T(f)$ and $v_n = 1$, then $\rho_v(f)$ is regular.*

Proof. By Lemma 4.1 we have only to show $\overline{v} \not\prec v$. We have $f(v - e^{(n)}) = 0$ from $v \in \min_{\succeq} T(f)$. Thus, the self-duality of f implies $f(\overline{v} + e^{(n)}) = 1$, which, together with $v \in \min_{\succeq} T(f)$, in turn implies $\overline{v} + e^{(n)} \not\prec v$. It follows from $\overline{v} + e^{(n)} \not\prec v$ and $v_n = 1$ that $\overline{v} \not\prec v$. □

Interestingly, the existence of a vector v satisfying the condition in Lemma 4.2 is equivalent the relevance of x_n to f, as proved in the following lemma.

LEMMA 4.3. *For a regular function f, x_n is relevant to f if and only if there exists a vector $v \in \min_{\succeq} T(f)$ such that $v_n = 1$.*

Proof. If such a vector v exists, then we have $f(v) = 1$ and $f(v - e^{(n)}) = 0$ (by $v \in \min_{\succeq} T(f)$). Thus x_n is relevant to f.

Conversely, if x_n is relevant, then there exists a vector $w \in \min T(f)$ such that $w_n = 1$, since, otherwise, the MDNF of f does not contain variable x_n, and hence x_n is irrelevant. The proof is complete if we show $w \in \min_{\succeq} T(f)$. Assume that $w \notin \min_{\succeq} T(f)$. Then there exists $u \in \min_{\succeq} T(f)$ such that $u \prec w$. Note that $w \in \min T(f)$ and $u \in \min_{\succeq} T(f)$ imply $u \not\preceq w - e^{(n)}$. Otherwise, by the profile-monotonicity of f, we would have $f(w - e^{(n)}) \geq f(u) = 1$, a contradiction to $w \in \min T(f)$. From $u \not\preceq w - e^{(n)}$ and $u \prec w$, it follows that $u_n = 1$ (possible only if $n \geq 2$), implying the only-if part. □

Lemma 4.2 deals with the case where x_n is relevant to f. Before dealing with the case where x_n is irrelevant to f, we first prove the following proposition.

PROPOSITION 4.4. *Let f be a regular function. For any $i, j \in V$ such that $i < j$, if x_j is relevant to f, then so is x_i.*

Proof. Assume that x_j is relevant, but x_i is not. Then there must be two vectors, v and w, such that $f(v) > f(w)$, where $v_k = w_k$ for all k ($1 \leq k \leq n$) except $k = j$, $v_j = 1$, and $w_j = 0$. Now define two vectors, $v' = (v_1, \ldots, v_{i-1}, 0, v_{i+1}, \ldots, v_n)$ and $w' = (w_1, \ldots, w_{i-1}, 1, w_{i+1}, \ldots, w_n)$. We thus have $v' \prec w'$. Since x_i is irrelevant, we should have

$$f(v') = f(v) > f(w) = f(w'),$$

a contradiction to the profile-monotonicity (regularity) of f. □

The above proposition implies that x_i is relevant to f if and only if $V_f \supseteq \{1, 2, \ldots, i\}$; in particular, x_n is relevant to f if and only if $V_f = \{1, 2, \ldots, n\} = V$. Corollary 4.5 generalizes Lemma 4.2 to the case where x_n may be irrelevant to f.

COROLLARY 4.5. *Let f be a regular self-dual function such that $|V_f| = i$ (≥ 2). If $v \in \min_{\succeq} T(f)$ and $v_i = 1$, then $\sigma_{(v;V_f)}(f)$ is regular.*

Proof. Let $I = V_f$ in (8). Then $\rho_v(Proj_{V_f}(f))$ is a regular function on V_f by Lemma 4.2. This completes the proof, since $Exp_V()$ preserves regularity. □

We now have the theoretical foundation for TRANS-REG-SD. By Lemma 4.2 and Corollary 4.5, if x_n is relevant to a given f, we can use transformation $\rho_v(f)$, with some v, to generate a new regular self-dual function and repeat this procedure as long as x_n is relevant. Once x_n becomes irrelevant to the newly generated function, f', we use the σ operation with respect to $V_{f'}$, and so forth.

What remains is the discussion of data we need to keep track of in implementing a sequence of σ transformations. It will be used later in computing the complexity of TRANS-REG-SD. To represent the sequence of regular self-dual functions $\{f'\}$ that TRANS-REG-SD generates, we represent each such function f' in terms of $\min T(f')$ and $\min_{\succeq} T(f')$ (see Lemma 4.8). The following proposition and corollary will prepare us for Lemma 4.8. For a vector v, let us introduce the following notation:

$$T_{\succ}(v) = \{w \mid w \succ v\} \text{ and } T_{\prec}(v) = \{w \mid w \prec v\}.$$

PROPOSITION 4.6. *$u \in T_{\succ}(v)$ is a minimal member with respect to $\prec$ in $T_{\succ}(v)$ if and only if $prof_u(i) = prof_v(i) + 1$ for some i, $1 \le i \le n$, and $prof_u(k) = prof_v(k)$ for all $k \neq i$.*

Proof. For simplicity, we present an informal "picture proof" using Figure 4.1. In Figure 4.1 (a) and (b), it is clear that the vector v whose profile is represented by the solid staircase is majorized by the vector u whose profile is represented by the dashed staircase and that u is a minimal vector with respect to $\prec$ in $T_{\succ}(v)$. The dashed staircase in Figure 4.1 (c) shows a nonminimal vector w with $prof_w(n) = |ON(w)| = prof_v(n) = |ON(v)|$. w is nonminimal, since it majorizes another vector $u \in T_{\succ}(v)$, whose profile satisfies $prof_u(3) = 1$ and $prof_u(4) = 2$. Similarly, it is easy to see that any member of $T_{\succ}(v)$ violating the conditions of this proposition majorizes another member of $T_{\succ}(v)$. □

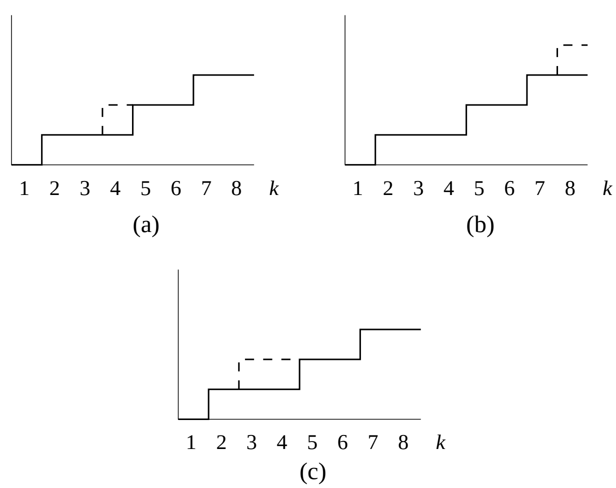

FIG. 4.1. *A "picture proof" of Proposition* 4.6.

Formula (12) in the following corollary follows immediately from the above proposition. Formula (13) is dual to (12).

COROLLARY 4.7.

$$(12)\quad \min_{\succeq} T_{\succ}(v) = \begin{cases} \{v + e^{(j)} - e^{(j+1)} \mid v_j = 0, v_{j+1} = 1, 1 \le j \le n-1\} \\ \qquad \cup \{v + e^{(n)}\} & \text{if } v_n = 0, \\ \{v + e^{(j)} - e^{(j+1)} \mid v_j = 0, v_{j+1} = 1, 1 \le j \le n-1\} & \text{if } v_n = 1. \end{cases}$$

$$(13)\quad \max_{\succeq} T_{\prec}(v) = \begin{cases} \{v - e^{(j)} + e^{(j+1)} \mid v_j = 1, v_{j+1} = 0, 1 \le j \le n-1\} \\ \qquad \cup \{v - e^{(n)}\} & \text{if } v_n = 1, \\ \{v - e^{(j)} + e^{(j+1)} \mid v_j = 1, v_{j+1} = 0, 1 \le j \le n-1\} & \text{if } v_n = 0. \end{cases}$$

We now show the effect of operation ρ_v on $\min T(f)$ and $\min_{\succeq} T(f)$.

LEMMA 4.8. *Let f be a regular self-dual function of $n\,(\geq 2)$ variables, and let $v \in \min_{\succeq} T(f)$ with $v_n = 1$. Then we have*

$$(14)\quad \min T(\rho_v(f)) = \min T(f) \setminus (\{v\} \cup \{\overline{v} + e^{(j)} \mid \max OFF(v) < j \le n\}) \cup \{\overline{v}\},$$

$$(15)\quad \min_{\succeq} T(\rho_v(f)) = \min_{\succeq} T(f) \setminus (\{v\} \cup \min_{\succeq} T_{\succ}(\overline{v})) \cup \{\overline{v}\} \\ \cup \{u \in \min_{\succeq} T_{\succ}(v) \mid u \not\succ z \text{ for all } z \in (\min_{\succeq} T(f) \setminus \{v\}) \cup \{\overline{v}\}\}.$$

Proof. By (9), we need to consider only the influence of (i) adding $\overline{v}$ to $T(f)$ and (ii) removing v from $T(f)$. To follow this proof, it will be helpful to have the following example: For a function $f = 12+13+145+234+235$ of five variables, if $v = (10011)$, then $\overline{v} = (01100)$, $\max OFF(v) = 2$, $\rho_v(f) = 12 + 13 + 234 + 235 + 23 + 145(2 + 3) = 12+13+23$, $\min_{\succeq} T(f) = \{(10100), (10011), (01101)\}$, and $\min_{\succeq} T(\rho_v(f)) = \{(01100)\}$.

(14)(i). By the self-duality of f, we have $\overline{v} \in \max F(f)$, hence $\overline{v} \in \min T(\rho_v(f))$. (See, e.g., the prime implicant 23 of $\rho_v(f)$ in Example 3.1.) Let us next consider vectors of the form $\overline{v}+e^{(j)}$, $j \in ON(v)\,(= OFF(\overline{v}))$, which may cease to be a minimal member as a result of operation (i). Note that these vectors belong to $T(f)$, since $\overline{v} \in \max F(f)$. We claim that $\overline{v} + e^{(j)} \in \min T(f)$ if and only if $j > \max OFF(v)$. (See then the prime implicants 234 and 235 of f in Example 3.1.) If $j > \max OFF(v)$, $\overline{v} + e^{(j)} \in \min T(f)$ holds, since otherwise there exists a vector $w \in \min T(f)$ such that $w < \overline{v} + e^{(j)}$. Since $w \neq \overline{v}$ and $w < \overline{v} + e^{(j)}$, $w \le \overline{v} + e^{(j)} - e^{(i)}$ holds for some $i \in OFF(v)\,(= ON(\overline{v}))$. Since $i < j$, we have $w \prec \overline{v}$. Since $f(\overline{v}) = 0$ by the self-duality of f, it follows from Lemma 2.1 that $f(w) = 0$ holds, a contradiction to $w \in \min T(f)$. This proves the if part of our claim.

Let $j \in ON(v)$ $(= OFF(\overline{v}))$ with $j \le \max OFF(v)$ and consider the vector $w = \overline{v} + e^{(j)} - e^{(\max OFF(v))}$. Since $j \in ON(v)$, we have $j < \max OFF(v)$. Thus $w \succ \overline{v}$, and hence $v \succ \overline{w}$ holds. If $f(w) = 0$, then $f(\overline{w}) = 1$ holds by the self-duality of f. Since $v \succ \overline{w}$, we have $v \notin \min_{\succeq} T(f)$, a contradiction. Thus $f(w) = 1$, implying that $\overline{v} + e^{(j)} \notin \min T(f)$ holds in this case.

(14)(ii). Since $v \notin \min T(\rho_v(f))$, there may be a $j \in OFF(v)$ such that $u = v+e^{(j)}$ belongs to $\min T(\rho_v(f))$. Since $f(v) = 1$, $f(v + e^{(j)} - e^{(n)}) = 1$, i.e., $v + e^{(j)} - e^{(n)} \in T(f)$, follows from the regularity of f. Thus no vector of the form $v + e^{(j)}$ belongs to $\min T(\rho_v(f))$, since there obviously exists a $w \in \min T(f)$ satisfying $w \le v + e^{(j)} - e^{(n)} < v + e^{(j)}$, a contradiction.

(15)(i). It is easy to see that $\overline{v} \in \min_{\succeq} T(\rho_v(f))$ holds, since otherwise $f(\overline{v}) = 0$ and there exists a vector w such that $w \prec \overline{v}$ and $f(w) = 1$, a contradiction to the regularity of f. Now consider any vector $w \in \min_{\succeq} T(f)$ majorizing $\overline{v}$, i.e., satisfying $w \succ \overline{v}$. Such a w must be removed from $\min_{\succeq} T(f)$ to construct $\min_{\succeq} T(\rho_v(f))$. Clearly, each such w is contained in $\min_{\succeq} T_{\succ}(\overline{v})$ (e.g., $w = (01110) \in \min_{\succeq} T(f)$ corresponding to the prime implicant 234 of f in Example 3.1).

(15)(ii). As noted in (14)(ii), we have $v \notin \min T(\rho_v(f))$, a fortiori, $v \notin \min_{\succeq} T(\rho_v(f))$. Let us consider the vectors in $\min_{\succeq} T_{\succ}(v)$, given by (12), since, besides $\overline{v}$, only they may be contained in $\min_{\succeq} T(\rho_v(f)) \setminus \min_{\succeq} T(f)$. Note that a vector $u \in \min_{\succeq} T_{\succ}(v)$ belongs to $\min_{\succeq} T(\rho_v(f))$, provided there is no vector $z \in \min_{\succeq} T(f) \setminus \{v\} \cup \{\overline{v}\}$ such that $z \prec u$. This is because $\min_{\succeq} T(f) \setminus (\{v\} \cup \min_{\succeq} T_{\succ}(\overline{v})) \cup \{\overline{v}\} \subseteq \min_{\succeq} T(\rho_v(f))$ (see (15)(i) above) and all vectors in $\min_{\succeq} T_{\succ}(\overline{v})$ majorizing $\overline{v}$. $\square$

From the proof of Lemma 4.8 (case (14)(i)), we can see that $\overline{v} + e^{(n)} \in \min T(f)$. Since $v_n = 1$ implies $n > \max OFF(v)$, $\{\overline{v} + e^{(j)} \mid \max OFF(v) < j \leq n\}$ is nonempty, and (14) implies the following lemma.

LEMMA 4.9. *Let f be a regular self-dual function of $n\,(\geq 2)$ variables, and let $v \in \min_{\succeq} T(f)$ with $v_n = 1$. Then*

$$|\min T(\rho_v(f))| \leq |\min T(f)| - 1, \tag{16}$$

$$\min T(\rho_v(f))_{x_n=1} \cup \{v, \overline{v} + e^{(n)}\} = \min T(f)_{x_n=1}, \tag{17}$$

where $S_{x_n=1}$ denotes the set $\{v \in S \mid v_n = 1\}$.

We are now ready to describe the transformation algorithm, TRANS-REG-SD. If we repeatedly apply the ρ_v operation (with different v's, of course) to a regular self-dual function f, until there is no vector $v \in \min_{\succeq} T(f)$ with $v_n = 1$, then by Lemmas 4.2, 4.3, and 4.9 we have a regular self-dual function f' to which x_n is irrelevant. This, together with (17), implies that $|\min T(f)_n|$ is even. Note that f' is not unique; i.e., it depends on the sequence of vectors $v \in \min_{\succeq} T(f)$ with $v_n = 1$ that are used in ρ_v. For example, consider a function $f = 12+13+145+234+235$ of five variables. For vectors $v = (10011)$ and $w = (01101)$, $\rho_v(f) = 12 + 13 + 23$ and $\rho_w(f) = 12 + 13 + 14 + 234$, respectively.

Now $V_{f'} = \{1, 2, \dots, j_1\}$ holds for some $j_1 \leq n - 1$. If $j_1 = 1$, we have $f' = x_1$ and we are done. If $j_1 \neq 1$, on the other hand, we apply $\sigma_{(v;V_{f'})}$ operations to f' instead of $\sigma_{(v;V_f)}\,(= \rho_v)$ until there is no vector $v \in \min_{\succeq} T(f')$ with $v_{j_1} = 1$. Since all the lemmas presented in this section are still valid for $\sigma_{(v;V_{f'})}$ and $v_{j_1} = 1$ in place of $\sigma_{(v;V_f)}\,(= \rho_v)$ and $v_n = 1$, we obtain a regular self-dual function f'' whose relevant variable set is $V_{f''} = \{1, 2, \dots, j_2\}$ with $j_2 < j_1$. By repeating this argument, we reach the one-variable regular self-dual function x_1. Formally, this sequence of transformations can be stated as follows.

Algorithm TRANS-REG-SD
Input: $\min T(f)$, where f is a regular self-dual function.
Output: Regular self-dual functions $f_0\,(= f)$, f_1, $f_2, \dots,$ $f_m\,(= x_1)$.
Step 0: Let $i = 0$ and $f = f_0$.
Step 1: Output f_i. If $f_i = x_1$, then halt.
Step 2: $f_{i+1} = \sigma_{(v^{(i)};V_{f_i})}(f_i)$, where $v^{(i)} \in \min_{\succeq} T(f_i)$ and $v^{(i)}_{\max V_{f_i}} = 1$. $i := i+1$. Return to Step 1.

By (16), the number m in the output from algorithm TRANS-REG-SD satisfies $m \leq |\min T(f)| - 1$. Since every self-dual function f satisfies $\rho_{\overline{v}}(\rho_v(f)) = f$ (see (9)), we can transform x_1 into any regular self-dual function g by repeatedly applying the σ operation to x_1 at most $|\min T(g)| - 1$ times. Thus we have the following theorem.

THEOREM 4.10. *Let f and g be any two regular self-dual functions. Then f can be transformed into g by repeatedly applying σ operations to f at most $|\min T(f)| + |\min T(g)| - 2$ times.*

In the subsequent sections, we consider the problems of generating all regular self-dual functions and of computing an optimum self-dual function with respect to a "g-regular" functional Φ (for the definition of g-regularity, see section 6) as applications of algorithm TRANS-REG-SD.

5. Generation of all regular self-dual functions. Let $\mathcal{C}_{R\text{-}SD}(n)$ denote the class of all regular self-dual functions of n variables. We present in this section an algorithm to generate all functions in $\mathcal{C}_{R\text{-}SD}(n)$ by applying the operation σ. The algorithm is *incrementally polynomial* [22] in the sense that the ith function $\phi_i \in \mathcal{C}_{R\text{-}SD}(n)$ is output in polynomial time in n and $\sum_{j=0}^{i-1} |\min T(\phi_j)|$ for $i = 1, 2, \ldots, |\mathcal{C}_{R\text{-}SD}|$.

To visualize the algorithm, we first define an undirected graph $G_n = (\mathcal{C}_{R\text{-}SD}(n), E)$, where $(g, f) \in E$, if there exists a vector $v \in \min_{\succeq} T(g)$ such that $\sigma_{(v;I)}(g) = f$ for some $I \supseteq V_g$.

Example 5.1. Figure 5.1 shows the graph G_5.[5] (Ignore the arrows on some edges for now.)

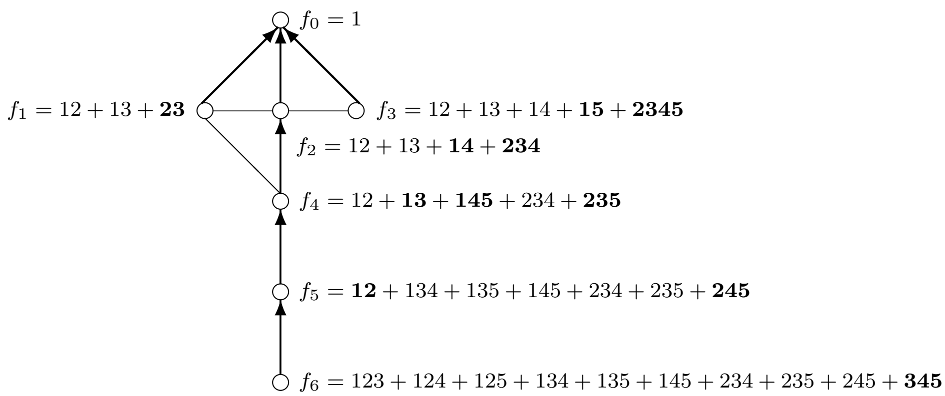

FIG. 5.1. *The graph G_5: the prime implicants corresponding to the vectors in $\min_{\succeq} T(f_i)$ are set in boldface.*

Theorem 4.10 implies that G_n is connected. Moreover, the condition $(g, f) \in E$ holds if and only if $(f, g) \in E$; i.e., G_n is indeed undirected, as shown by the following proposition. For a vector $v \in \{0, 1\}^n$ and $I \subseteq V$, let $v[I]\underline{0}$ denote the vector u defined by $ON(u) = ON(v) \cap I$; i.e., $u[I] = v[I]$ and the remaining components of u, if any, are all set to 0's.

PROPOSITION 5.2. *Let $f \in \mathcal{C}_{R\text{-}SD}(n)$ and $g = \sigma_{(w;I)}(f)$ for $w \in \min_{\succeq} T(f)$ such that $I \supseteq V_f$ and $\overline{w}[I] \not\succ w[I]$. Then $g \in \mathcal{C}_{R\text{-}SD}(n)$ and $f = \sigma_{(u;I)}(g)$, where $u = \overline{w}[I]\underline{0} \in \min_{\succeq} T(g)$.*

Proof. $g \in \mathcal{C}_{R\text{-}SD}(n)$ follows from Theorem 3.2 and Lemma 4.1. From (10), we have

$$T(g) = (T(f) \setminus w[I]\underline{*}) \cup \overline{w}[I]\underline{*}.$$

Let $f' = \sigma_{(u;I)}(g)$. Then

$$T(f') = (T(g) \setminus \overline{w}[I]\underline{*}) \cup w[I]\underline{*}.$$

We thus have $T(f') = T(f)$, hence $f' = f$. $u = \overline{w}[I]\underline{0} \in \min_{\succeq} T(g)$ follows from Lemma 4.1 and the fact that f is regular. $\square$

[5] Note that in this section the subscripts of the functions $\{f_i\}$ are reversed from those used in TRANS-REG-SD; for example, f_0 now denotes the function x_1.

Example 5.3. For example, consider the function $f = f_2 = 12 + 13 + 14 + 234$ in Figure 5.1, where we assume $n = 5$. We have $\min_{\succeq} T(f_2) = \{(10010), (01110)\}$. Pick $w = (10010) \in \min_{\succeq} T(f_2)$, which satisfies $\overline{w}[I] \not\prec w[I]$ for $I = V_{f_4}$. It is easy to see that $g = \sigma_{(w;I)}(f_2) = 12+13+14(2+3+5)+235+234 = 12+13+145+234+235 = f_4$. For $u = \overline{w} = (01101) \in \min_{\succeq} T(g)$, we have $\sigma_{(u;I)}(g) = 12+13+234+235(1+4)+14 = f_2$. Note that $\overline{w}[I]\underline{0} = \overline{w}[I]$ since $I = V$.

For two distinct vectors $u = (u_1, \ldots, u_n)$ and $v = (v_1, \ldots, v_n)$, we say that u is *lexicographically smaller* than v, written $u \lessdot v$, if for some k $(1 \le k \le n)$ $u_k < v_k$ and $u_i = v_i$ for all i $(1 \le i < k)$. Thus among the vectors in $\{0,1\}^3$, for example, we have $(000) \lessdot (001) \lessdot (010) \lessdot (011) \lessdot (100) \lessdot (101) \lessdot (110) \lessdot (111)$.

Let $f_0 = x_1$ be the designated function in $\mathcal{C}_{R\text{-}SD}(n)$ and consider the problem of transforming an arbitrary function $g \in \mathcal{C}_{R\text{-}SD}(n)$ to f_0 by repeatedly applying the σ operation as in algorithm TRANS-REG-SD. Note that the transformation path from a given g to f_0 is not unique. Thus, to make the path unique, we choose for the σ operation the lexicographically smallest vector $\tilde{v} \in \min_{\succeq} T(g)$ such that $\tilde{v}_{\max V_g} = 1$. Let μ be such an operation, i.e.,

$$\mu(g) = \sigma_{(\tilde{v};V_g)}(g). \tag{18}$$

In this way, we define a directed spanning tree of G_n, $RT_n = (\mathcal{C}_{R\text{-}SD}(n), A_{RT})$, such that (g, f) is a directed arc in A_{RT} if and only if $\mu(g) = f$. If $\mu()$ is applied recursively to g, we eventually reach a function h such that $V_h \subset V_g$ (see (17)). For example, in Figure 5.1, we have $V_{f_2} \subset V_{f_4}$. Thus, RT_n is an in-tree rooted at $f_0 = x_1$. In Figure 5.1, A_{RT} is indicated by the thick arcs.

Our algorithm GEN-REG-SD presented later in this section, which generates all regular self-dual functions of n variables for a given n, will traverse RT_n from f_0 in a depth-first manner in the reverse direction of the arcs in A_{RT}, outputting each regular self-dual function f when it first visits f. This type of enumeration is called *reverse search* in [2, 3] and has been applied to many other enumeration problems such as the extreme points of a convex polyhedron, the arrangements of hyperplanes, the triangulations of a polygon, matroid bases, and so on. When RT_n is traversed from f_0, for each arc $(g, f) \in A_{RT}$, the end node f (nearer the root) is visited before the end node g (farther from the root). Unfortunately, when we first visit node f we cannot identify the incoming arcs (in A_{RT}) towards node f from among the edges in E (of G_n). In other words, knowing f, we cannot find g such that $(g, f) \in A_{RT}$. Note that (18) computes f given g, not the other way around. In Lemma 5.5 below, we find the "inverse" of (18) in the sense that u in Proposition 5.2 coincides with $\tilde{v}$ in (18).

The following lemma is essentially a restatement of Lemma 4.8 in a slightly generalized form.

LEMMA 5.4. *Let $f, g \in \mathcal{C}_{R\text{-}SD}(n)$ satisfy $f = \mu(g) = \sigma_{(\tilde{v};V_g)}(g)$, where $\tilde{v}$ is the lexicographically smallest vector in $\min_{\succeq} T(g)_{x_{\max V_g}=1}$. With $w = \overline{(\tilde{v})}[V_g]\underline{0}$, we have*

$$\min T(f) = \min T(g) \setminus (\{\tilde{v}\} \cup \{w + e^{(j)} \mid, \max(OFF(\tilde{v}) \cap V_g) < j \le \max V_g\}) \;\cup\; \{w\}, \tag{19}$$

$$\min T(g) = (\min T(f) \setminus \{w\}) \cup \{\tilde{v}\} \cup \{w + e^{(j)} \mid \max(ON(w) \cap V_g) < j \le \max V_g\}, \tag{20}$$

$$\min_{\succeq} T(f) = \min_{\succeq} T(g) \setminus (\{\tilde{v}\} \cup \min_{\succeq} T_{\succ}(w; V_g)) \;\cup\; \{w\} \cup \{u \in \min_{\succeq} T_{\succ}(\tilde{v}; V_g) \mid u \not\succ z \text{ for all } z \in (\min_{\succeq} T(g) \setminus \{\tilde{v}\}) \cup \{w\}\}, \tag{21}$$

$$(22)\quad \min_{\succeq} T(g) = \min_{\succeq} T(f) \setminus (\{w\} \cup \min_{\succeq} T_{\succ}(\tilde{v}; V_g)) \cup \{\tilde{v}\} \\ \cup \{u \in \min_{\succeq} T_{\succ}(w; V_g) \mid u \not\succ z \ \ \textit{for all} \ \ z \in (\min_{\succeq} T(f) \setminus \{w\}) \cup \{\tilde{v}\}\},$$

where $T_{\succ}(v; I) = \{u \mid u \succ v, ON(u) \subseteq I\}$ *for a vector* v *and an index set* $I \subseteq V$.

Proof. The proof follows from Lemma 4.8, which is a special case of this lemma ($V_g = V$). ∎

Note that by (20) we have $w + e^{(\max V_g)} \in \min T(g)$, implying $\tilde{v} \lessdot w + e^{(\max V_g)}$, hence $\tilde{v}_1 = 0$ and $w_1 = 1$, since $\overline{w}_1 = \tilde{v}_1$. In Lemma 5.4, conceptually, $\tilde{v}$ was explicitly chosen first, and w was specified in terms of $\tilde{v}$. The following lemma will enable us to choose w explicitly, so that $\tilde{v}$ is chosen implicitly. Note that condition (c) in Lemma 5.5 involves the lexicographic order in $\min_{\succeq} T(f)$, which one can compute given f, while $\tilde{v}$ in (18) is defined in terms of the lexicographically smallest vector in $\min_{\succeq} T(g)$, which one can compute given g. Note also that $\tilde{v}$ is unique for a given regular self-dual function g, but vector w which satisfies the conditions of Lemma 5.5 is in general not unique for a given regular self-dual function f. This reflects the fact that a nonroot node in graph RT_n has one parent but in general has more than one child node.

LEMMA 5.5. *Let* $f \in \mathcal{C}_{R\text{-}SD}(n)$ *and* $g = \sigma_{(w;V_g)}(f)$ *for* $w \in \min_{\succeq} T(f)$[6] *such that* $\overline{w}[V_g] \not\lessdot w[V_g]$ *and* $V_g \supseteq V_f$. *Then* $f = \mu(g)\,(= \sigma_{(\tilde{v};V_g)}(g))$ *i.e.,* $(g, f) \in A_{RT}$ *(the arc set of* RT_n*), if and only if*

(a) $w_{\max V_g} = 0$,
(b) $w_1 = 1$, *and*
(c) $\overline{w}[V_g]\underline{0}$ *is lexicographically smaller than any vector in* $\min_{\succeq} T(f)_{x_{\max V_g}=1}$.

Proof. Let us first consider the only-if part, assuming $f = \sigma_{(\tilde{v};V_g)}(g)$, where $\tilde{v}$ is the lexicographically smallest vector in $\min_{\succeq} T(g)_{x_{\max V_g}=1}$ as in Lemma 5.4. Then $g = \sigma_{(w;V_g)}(f)$ implies that $w = \overline{(\tilde{v})}[V_g]\underline{0}$. Since $\tilde{v}_{\max V_g} = 1$ by definition, we have $w_{\max V_g} = 0$, hence (a) holds.

(b) was shown above in the comment immediately after Lemma 5.4. To prove (c), by (21), it suffices to show that $\tilde{v} \lessdot w$ and that $\tilde{v}$ is lexicographically smaller than any vector in the set $\min_{\succeq} T_{\succ}(\tilde{v}; V_g)$, since any vector u in the first term in (21) with $u_{\max V_g} = 1$ must be lexicographically larger than $\tilde{v}$ by definition of $\tilde{v}$. The former follows immediately from (b) and $\tilde{v} = \overline{w}[V_g]\underline{0}$, and the latter is obvious, since $u \prec v$ implies that $u \lessdot v$ for any vectors u and v.

To prove the if part, let $\tilde{v} = \overline{w}[V_g]\underline{0}$. We want to show that $\tilde{v}$ is lexicographically the smallest in $\min_{\succeq} T(f)_{x_{\max V_g}=1}$. By (c) and (22), it suffices to show that $\tilde{v}$ is lexicographically smaller than any vector in $\min_{\succeq} T_{\succ}(w; V_g)$ (the last term in (22)). This is obvious since $w_1 = 1$ and $\tilde{v}_1 = 0$. ∎

Example 5.6. Using Lemma 5.5, we can construct $RT_n = (\mathcal{C}_{R\text{-}SD}(n), A_{RT})$. $RT_6 = (\mathcal{C}_{R\text{-}SD}(6), A_{RT})$ is shown in Figure 5.2.

The function numbers in the figure denote those regular self-dual functions of six variables shown in the following table, which was derived from the work by Bioch and Ibaraki [8]. (Some function numbers have been changed.)

[6] This means that $(g, f) \in E$ (the edge set of G_n). g depends on the choice of w.

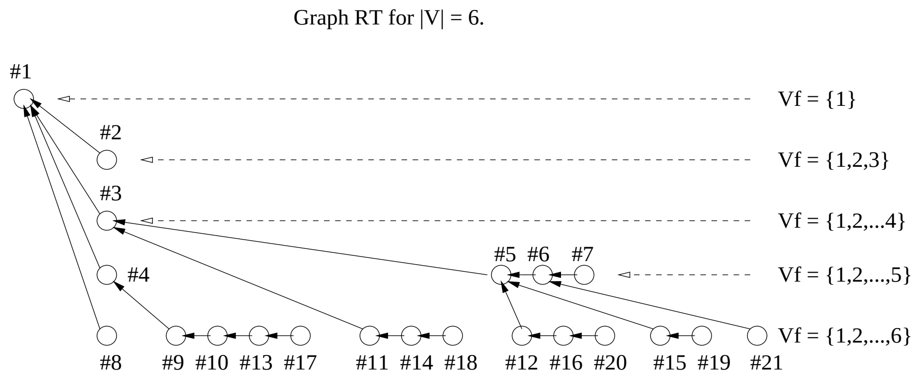

FIG. 5.2. $RT_6 = (\mathcal{C}_{R\text{-}SD}(6), A_{RT})$.

#	$\lvert V_f \rvert$	Function
1	1	**1***
2	3	12+13+**23**†
3	4	12+13+**14***+**234**†
4	5	12+13+14+**15***+**2345**†
5		12+**13***+**145***+234+**235**†
6		**12***+134+135+145+234+235+**245**†
7		123+124+125+134+135+145+234+235+245+**345**†
8	6	12+13+14+15+**16**+**23456**†
9		12+13+**14***+**156**+2345+**2346**†
10		12+**13***+145+146+**156**+2345+2346+**2356**†
11		12+**13***+145+**146**+**234**+**2356**†
12		12+**13***+234+235+**236**† +**1456**
13		**12***+134+135+136+145+146+**156**+2345+2346+2356 +**2456**†
14		**12***+134+135+136+145+**146**+**234**+2356+**2456**†
15		**12***+134+135+**136**+**145**+234+**235**+**2456**†
16		**12***+134+135+136+234+235+**236**+1456+**2456**†
17		123+124+125+126+134+135+136+145+146+**156**+2345+2346+2356 +2456+**3456**†
18		123+124+125+126+134+135+136+145+**146**+**234**+2356+2456+**3456**†
19		123+124+125+126+134+135+**136**+**145**+234+**235**+2456+**3456**†
20		123+124+125+126+134+135+136+234+235+**236**+1456+2456+**3456**†
21		123+124+125+**126**+134+135+145+234+235+**245**+**3456**†

In the table, the prime implicants corresponding to the vectors in $\min_{\succeq} T(f)$ are shown in boldface. Let t_w denote the prime implicant corresponding to vector $w \in \min T(f)$; i.e., t_w contains the variable x_i as a factor if and only if $i \in ON(w)$. In other words, $i \in P(t_w)$ if and only if $i \in ON(w)$ and $N(t_w) = \emptyset$. (For the definition of $P()$ and $N()$, see the beginning of section 2.) * indicates a prime implicant t_w such that vector w satisfies the conditions of Lemma 5.5. When each function is called g, the prime implicant t_v with † corresponds to the unique vector $\tilde{v}$.

In general, each vector w with $w_{\max V_f} = 1$ implies that f has $(n - \max V_f)$ children due to w in RT_n, one each for $|V_g| = |V_f| + 1, \ldots, n$. Similarly, each vector w with $w_{\max V_f} = 0$ implies that f has $(n - \max V_f + 1)$ children due to w, one each for $|V_g| = |V_f|, |V_f| + 1, \ldots, n$. For example, it is observed in Figure 5.2 that $t_w = 14$ in function #3 gives rise to $6-4 = 2$ children, while $t_w = 15$ in function #4 gives rise to just 6−5=1 child. Similarly, $t_w = 145$ in function #5 gives rise to just one child, while $t_w = 13$ in function #5 gives rise to 6−5+1 = 2 children, and so forth.

Note that if $V_g \neq V_f$ (i.e., $V_g \supset V_f$), Lemma 5.5 implies that $f = \mu(g)$ if and only if $w_1 = 1$, since $V_g \supset V_f$ and $w \in \min_{\succeq} T(f)$ imply conditions (a) and (c) are vacuous in this case. Thus, for an index set $I \supset V_f$, any vector $w \in \min_{\succeq} T(f)$ that satisfies $w_1 = 1$ and $\overline{w}[I] \not\leq w[I]$ produces $g = \sigma_{(w;I)}(f)$ such that $f = \mu(g)$.

We now discuss the data structures for $\min T(f)$ and $\min_{\succeq} T(f)$. The set $\min T(f)$ is represented by a *binary tree*, denoted by $B(\min T(f))$, of height n, in which the left edge (resp., right edge) from a node at depth $j-1$ (the root is at depth 0) represents the case $x_j = 1$ (resp., $x_j = 0$). A leaf node t of $B(S)$ at depth n stores the vector $v \in S \,(\subseteq \{0,1\}^n)$, the components of which correspond to the edges of the path from the root to t. In order to have a compact representation, the edges with no descendant leaves are removed from $B(S)$.

Example 5.7. Figure 5.3 shows $B(\min T(f))$ for $\min T(f) = \{v^{(1)} = (110000), v^{(2)} = (101000), v^{(3)} = (100110), v^{(4)} = (011100), v^{(5)} = (011010)\}$ (i.e., $f = 12 + 13 + 145 + 234 + 235$).

With this data structure, it is easy to see that we can apply operations MEMBER (i.e., check if $v \in S$), INSERT (i.e., update $S := S \cup \{v\}$), and DELETE (i.e., update $S := S \setminus \{v\}$) all in $O(n)$ time. Moreover, since the rightmost (resp., leftmost) path in $B(S)$ represents the lexicographically smallest (resp., largest) vector in S, we can output from $B(S)$ the lexicographically smallest/largest vector in S in $O(n)$ time.

Let $V_i = \{1, 2, \ldots, i\}$ and define $\alpha : \{0,1\}^n \to \mathbb{Z}^+$ by

$$\alpha(v) = \min(\{\, i \mid \overline{v}[V_i] \not\leq v[V_i],\ ON(v) \subseteq V_i \subseteq V\} \cup \{n+1\}). \tag{23}$$

By definition, if no i satisfies the condition on the right-hand side, $\alpha(v) = n+1$ holds. We can easily see that $\overline{v}[V_i] \not\leq v[V_i]$ holds for all $i \geq \alpha(v)$, if $\alpha(v) < n+1$. Based on this $\alpha(v)$ and $\max ON(v)$, we decompose $\min_{\succeq} T(f)$ into many subsets as follows: $\min_{\succeq} T(f) = \bigcup_{\substack{j \leq i \leq n+1 \\ j=1,2,\ldots,n}} \min_{\succeq} T(f)_{(i,j)}$, where

$$\min{}_{\succeq} T(f)_{(i,j)} = \{v \in \min{}_{\succeq} T(f) \mid \alpha(v) = i,\ \max ON(v) = j\}.$$

For the above example function $f = 12 + 13 + 145 + 234 + 235$, we have $\min_{\succeq} T(f) = \{v^{(2)} = (101000), v^{(3)} = (100110), v^{(5)} = (011010)\}$. From this, we get $\min_{\succeq} T(f)_{(5,3)} = \{v^{(2)}\}$, $\min_{\succeq} T\ (f)_{(5,5)} = \{v^{(3)}, v^{(5)}\}$, and $\min_{\succeq} T(f)_{(i,j)} = \emptyset$, otherwise.

Our algorithm keeps $\min_{\succeq} T(f)$ as $\bigcup_{\substack{j \leq i \leq n+1 \\ j=1,2,\ldots,n}} B(\min_{\succeq} T(f)_{(i,j)})$ and $\bigcup_{j=1}^{n} B(\min_{\succeq} T(f)_j)$, where $\min_{\succeq} T(f)_j = \bigcup_{i=j}^{n+1} \min_{\succeq} T(f)_{(i,j)}$.

We start depth-first search from the root f_0. Note that $\min_{\succeq} T(f_0)_{(1,1)} = (10\ldots0)$ and $V_{f_0} = \{1\}$. During the depth-first search, when we visit node f, we first set up $I := V_f$ as the index set. In the order $(i,j) = (1,1), \ldots, (1, \max I–1), (2,1), \ldots, (2, \max I–1), \ldots, (\max I, 1), \ldots, (\max I, \max I–1)$, we then check if the lexicographically largest $w^{(i,j)}$ in $\min_{\succeq} T(f)_{(i,j)}$ satisfies (b) $w_1^{(i,j)} = 1$ and (c) that the vector $v^{(i,j)}$, defined by $ON(v^{(i,j)}) = OFF(w^{(i,j)}) \cap I$, is lexicographically smaller than any vector in $\min_{\succeq} T(f)_{\max I}$ (i.e., $\min_{\succeq} T(f)_{x_{\max I = 1}}$). If there exists such a vector $w^{(i,j)}$, we move to $g = \sigma_{(w^{(i,j)};I)}(f)$. Since $w^{(i,j)}$ satisfies conditions (a), (b), and (c) of Lemma 5.5, we have $f = \mu(g)$. Moreover, if $w^{(i,j)}$ does not satisfy (b) (resp., (c)), then no vector in $u \in \min_{\succeq} T(f)_{(i,j)}$ satisfies (b) (resp., (c)). This means that we do not have to check the vectors in $\min_{\succeq} T(f)_{(i,j)}$ other than $w^{(i,j)}$. Thus, if no $w^{(i,j)}$ ($i = 1, 2, \ldots, \max I$,

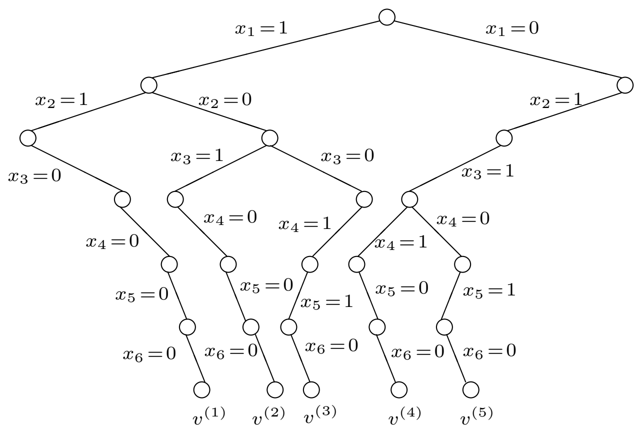

FIG. 5.3. *A binary tree* $B(\min T(f))$*, where* $f = 12 + 13 + 145 + 234 + 235$ *(a function of six variables).*

$j = 1, 2, \dots, \max I - 1$) satisfies (b) and (c), then we check if $\max I = n$. If so, we return to $h = \mu(f)$ from f (if $f = f_0 = x_1$, then halt); otherwise, we update $I := I \cup \{\max I + 1\}$ and again check if $w^{(i,j)}$ satisfies (b) and (c).

When the depth-first search returns to f from g by backtracking (i.e., $f = \mu(g) = \sigma_{(v^{(i^*,j^*)};V_g)}(g)$), we set $I := V_g$ and move on to the vector w' which lexicographically follows $w^{(i^*,j^*)}$ in $\min_{\succeq} T(f)_{(i^*,j^*)}$. If this w' satisfies conditions (b) and (c) of Lemma 5.5, then we move to $g' = \sigma_{(w';I)}(f)$; otherwise, check if $w^{(i,j)}$ satisfies (b) and (c), according to the ordering $(i,j) = (i^*, j^*+1), \dots, (i^*, \max I - 1), (i^*+1, 1), \dots, (i^*+1, \max I - 1), \dots, (\max I, 1), \dots, (\max I, \max I - 1)$.

This procedure has the advantage that there is no need to maintain the data of the entire search tree RT_n but only the information about the current function is sufficient. Our algorithm can be stated formally as follows:

Algorithm GEN-REG-SD

Input: A positive integer n.

Output: All regular self-dual functions of n variables.

Step 0: Let $f := f_0$, $I := V_f$, $w := (11\cdots1)$, and output f. /* Note that $(11\cdots1)$ is the special vector, indicating that no σ operation is applied to f.*/

Step 1: If $w = (11\cdots1)$, then go to Step 2. /* In this case, no σ operation was applied to f.*/

Else if $I = V_f$, then go to Step 3. /* In this case, $V_g = V_f$ holds for $g = \sigma_{(w,I)}(f)$; i.e., we have applied the σ operation with $I = V_f$ previously.*/

Else, go to Step 4. /* In this case, $V_g \supset V_f$ holds for $g = \sigma_{(w,I)}(f)$; i.e., we have applied the σ operation with $I \supset V_f$ previously, and hence we already applied the σ operation for with V_i, where $V_f \subseteq V_i \subset I$.*/

Step 2: Try to find the lexicographically largest $w' = w^{(i,j)}$ in $\min_{\succeq} T(f)_{(i,j)}$ satisfying the two conditions of Lemma 5.5, (b) $w_1^{(i,j)} = 1$ and (c) the vector $v^{(i,j)}$, defined by $ON(v^{(i,j)}) = OFF(w^{(i,j)}) \cap I$, is lexicographically smaller than any vector in $\min_{\succeq} T(f)_{\max I}$, in the order $(i,j) = (1,1), \dots, (1, \max I - 1), (2,1), \dots, (2, \max I - 1), \dots, (\max I, 1), \dots, (\max I, \max I - 1)$. /* Recall

$I = V_f$. First apply σ with the index set $I = V_f$.*/

Then consider the following four cases:

(i) There is a w' satisfying the above conditions: Let $f := \sigma_{(w';I)}(f)$ and $w := (11\cdots 1)$, output f, and return to Step 1 (downward move).

(ii) There is no such w' and $\max I < n$:[7] Try to find the lexicographically largest $w'' = w^{(i,j)}$ in $\min_{\succeq} T(f)_{(i,j)}$ satisfying (b) $w_1^{(i,j)} = 1$, according to the ordering $(i,j) = (1,1), \ldots, (1, \max V_f), (2,1), \ldots, (2, \max V_f), \ldots, (n,1), \ldots, (n, \max V_f)$.
If there is no such w'', then go to either case (iii) or (iv). Else, let $w'' \in \min_{\succeq} T(f)_{(i^*,j^*)}$. Then let $I := V_{i^*}$ $(= \{1,2,\ldots,i^*\})$ if $i^* \geq \max I + 2$; otherwise, let $I := I \cup \{\max I + 1\}$, $f := \sigma_{(w'';I)}(f)$, and $w := (11\cdots 1)$, output f, and return to Step 1 (downward move).

(iii) There is no such w' or w'', and $f = f_0$: Halt (all functions have been output).

(iv) There is no such w' or w'', and $f \neq f_0$: Let v be the lexicographically smallest vector in $\min_{\succeq} T(f)_{\max V_f}$. Update f, I, and w by letting $I := V_f$, $f := \mu(f) = \sigma_{(v;V_f)}(f)$, and $ON(w) = OFF(v) \cap I$, respectively. Return to Step 1 (backtrack).

Step 3: Let $w \in \min_{\succeq} T(f)_{(i^*,j^*)}$. Try to find the vector w' satisfying conditions (b) and (c) of Lemma 5.5, according to the ordering (1) the vector that lexicographically follows w in $\min_{\succeq} T(f)_{(i^*,j^*)}$, followed by (2) $w^{(i,j)}$, where $(i,j) = (i^*, j^*+1), \ldots, (i^*, \max I - 1), (i^*+1, 1), \ldots, (i^*+1, \max I - 1), \ldots, (\max I, 1), \ldots, (\max I, \max I - 1)$. /*Try to apply σ with the current I with $I = V_f$.*/

Then consider the four cases of Step 2.

Step 4: Let $w \in \min_{\succeq} T(f)_{(i^*,j^*)}$. Try to find the vector w' satisfying (b) $w'_1 = 1$, according to the ordering (1) the vector that lexicographically follows w in $\min_{\succeq} T(f)_{(i^*,j^*)}$, followed by (2) $w^{(i,j)}$, where $(i,j) = (i^*, j^*+1), \ldots, (i^*, \max V_f), (i^*+1, 1), \ldots, (i^*+1, \max V_f), \ldots, (\max I, 1), \ldots, (\max I, \max V_f)$. /*Try to apply σ with the current I with $I \supset V_f$.*/

Then consider the four cases of Step 2.

In case (ii), we check if there exist a set $I'' \supset I$ and a vector $w'' \in \min_{\succeq} T(f)$ such that $f = \mu(g)$, where $g = \sigma_{(w'';I'')}(f)$. Since $I'' \supset I \supseteq V_f$, we just check if $w''_1 = 1$, as noted in the second paragraph after Example 5.6. According to the orderings on w' and w'', algorithm GEN-REG-SD traverses RT_n depth-first; i.e., each arc in RT_n is traversed only twice, downward and upward.

To analyze the time complexity of GEN-REG-SD, we need two more lemmas.

LEMMA 5.8. *Given the data structures $B(\min T(f))$, $B(\min_{\succeq} T(f)_{(i,j)})$ $(j = 1,2,\ldots,n,\ i = j, j+1, \ldots, n+1)$ and $B(\min_{\succeq} T(f)_j)$ $(j = 1,2,\ldots,n)$, each iteration of Steps* $1 \sim 4$ *in algorithm* GEN-REG-SD *computes either w' or w'' (or concludes that no such vector exists) in $O(n^3)$ time.*

Proof. Since we can check if a given vector u satisfies (b) and (c) of Lemma 5.5 in $O(n)$ time, and since there are $2n^2$ candidates for either w' or w'', each iteration requires $O(n \times n^2) = O(n^3)$ time. ∎

LEMMA 5.9. *Let $f, g \in \mathcal{C}_{R\text{-}SD}(n)$ satisfy $f = \mu(g)$. Let $f = \sigma_{(v;I)}(g)$ and $g = \sigma_{(w;I)}(f)$. Given the data structures for g (i.e., $B(\min T(g))$, $B(\min_{\succeq} T(g)_{(i,j)})$ $(i = 1,2,\ldots,n+1,\ j = 1,2,\ldots,n)$, and $B(\min_{\succeq} T(g)_j)$ $(j = 1,2,\ldots,n)$), v, and I,*

[7] Note that $I \supseteq V_f$, and hence $\min_{\succeq} T(f)_j = \emptyset$ holds for all $j \geq \max I + 1$.

we can compute data structures of f in $O(n^3)$ time. Similarly, given data structures of f, w, and I, we can compute data structures of g in $O(n^3)$ time.

Proof. We shall prove only the first assertion of the lemma, since the second assertion can be proved analogously. It directly follows from (19) in Lemma 5.4 that we can compute $B(\min T(f))$ in $O(n^2)$ time.

As for the rest, it suffices to show that, given a vector u, the membership $u \in \min_{\succeq} T(f)$ can be checked in $O(n^2)$ time, since at most n vectors are deleted from or added to $\min_{\succeq} T(g)$ to construct $\min_{\succeq} T(f)$ by (21) and (12). Note that a vector u is contained in $\min_{\succeq} T(f)$ if and only if $f(z) = 0$ holds for all vectors $z \in \max_{\succeq} T_{\prec}(u)$. From (13), we have at most n such vectors z. Moreover, we can check if a given vector z satisfies $f(z) = 1$ in $O(n)$ time if $B(\min T(f))$ is prepared [34]. This completes the proof. □

By Lemmas 5.8 and 5.9, each iteration of Step 1 can be carried out in $O(n^3)$ time, except for the outputting of function f. Since each arc $(g, f) \in A_{RT}$ is traversed twice, algorithm GEN-REG-SD requires $O(n^3|\mathcal{C}_{R\text{-}SD}(n)|)$ time, plus the time for outputting all functions in $\mathcal{C}_{R\text{-}SD}(n)$, i.e., $O(nM_{sum})$ time, where

$$M_{\text{sum}} = \sum_{f \in \mathcal{C}_{R\text{-}SD}(n)} |\min T(f)|.$$

Algorithm GEN-REG-SD clearly requires $O(nM_{\max})$ space, where

$$M_{\max} = \max_{f \in \mathcal{C}_{R\text{-}SD}(n)} |\min T(f)|.$$

Thus we have the following theorem.

THEOREM 5.10. *Algorithm* GEN-REG-SD *generates all functions in $\mathcal{C}_{R\text{-}SD}(n)$ and is incrementally polynomial. It requires $O(n^3|\mathcal{C}_{R\text{-}SD}(n)| + nM_{sum})$ time and $O(nM_{\max})$ space.*

COROLLARY 5.11. *All functions in $\mathcal{C}_{R\text{-}SD}(n)$ can be scanned in $O(n^3|\mathcal{C}_{R\text{-}SD}(n)|)$ time.*

By Lemma 2.2, the regular functions are all *representatives of equivalence classes under permutation*; i.e., there is no regular function f that is equivalent to another regular function $g\,(\neq f)$ under permutation. Therefore, our algorithm generates the nonequivalent functions. Let us remark that the algorithms in [9, 18] are not polynomial if we try to output only nonequivalent functions.

It is known that the positive self-dual functions of up to $n = 5$ variables are all threshold functions (and hence regular if we consider the representatives), but there are many nonregular self-dual functions for $n \geq 6$, even if we consider the representatives (see Example 5.6). Moreover, it is known [28] that all regular positive self-dual functions for $n \leq 9$ are threshold functions.

6. Optimum self-dual function for regular functional Φ. Let φ be a *pseudo-Boolean function*; i.e., φ is a mapping from $\{0,1\}^n$ to the set of real numbers $\mathbb{R}$. A pseudo-Boolean function φ is said to be g-*regular* if $\varphi(v) \geq \varphi(w)$ holds for all pairs of vectors v and w with $v \succeq w$. For a Boolean function f, let

$$\Phi(f) = \sum_{v \in T(f)} \varphi(v), \tag{24}$$

where φ is a pseudo-Boolean function. The functional Φ is also said to be g-*regular* if φ is g-regular. As an example of a g-regular pseudo-Boolean functional of interest, we cite the *availability* of a Boolean function, defined as follows. Each element $i \in V$ has the operational probability p_i $(0 \le p_i \le 1)$; i.e., the ith element is *operational* with the probability p_i, while it is *failed* with the probability $1 - p_i$. We assume that each element takes on its state independently. Then the *availability* $A(f)$ of a Boolean function f is defined by

$$A(f) = \sum_{v \in T(f)} \left(\prod_{i \in ON(v)} p_i \prod_{i \in OFF(v)} (1 - p_i) \right). \tag{25}$$

If we interpret $T(f)$ as the set of states in which the n-element system defined by the Boolean function f is working, then $A(f)$ represents the probability that the system represented by f is working. The availability has been studied extensively [35], especially in the case where f represents ND coteries (i.e., f is positive self-dual) [1, 5, 15, 33, 36, 38]. It is known [1, 36] that any element i with $p_i < 1/2$ can be ignored; i.e., x_i is irrelevant for all positive self-dual functions f having the maximum availability. The only exception is when all the elements have $p_i < 1/2$. In this case, it is known [1] that $f = x_j$ has the maximum availability, where j is the element such that $p_j \ge p_i$ for all i. Thus, we can assume that $p_i \ge 1/2$ holds for all i. Moreover, we assume without loss of generality that

$$p_1 \;\ge\; p_2 \;\ge\; \cdots \;\ge\; p_n \;\ge\; 1/2.$$

Now, let $\varphi(v) = \prod_{i \in ON(v)} p_i \prod_{i \in OFF(v)} (1 - p_i)$. Then we have $\Phi(f) = A(f)$. It follows from the assumption on the order of probabilities that $A(f)$ is g-regular.

In this section, we consider the functions f that maximize g-regular functional Φ among all self-dual functions.

LEMMA 6.1. *Given a* g*-regular function* φ*, let* Φ *be a* g*-regular functional defined by* (24)*. Then the following statements regarding* f *are equivalent.*

(i) $\Phi(f)$ *is maximum among all self-dual functions.*

(ii) *All vectors* $v \in T(f)$ *satisfy* $\varphi(v) \ge \varphi(\overline{v})$*.*

(iii) *All vectors* $v \in \min_{\succeq} T(f)$ *satisfy* $\varphi(v) \ge \varphi(\overline{v})$*.*

Proof. Let us prove (i) $\Longrightarrow$ (ii) $\Longrightarrow$ (iii) $\Longrightarrow$ (i). Since (ii) $\Longrightarrow$ (iii) is obvious, we show (i) $\Longrightarrow$ (ii) and (iii) $\Longrightarrow$ (i).

(i) $\Longrightarrow$ (ii): If there exists a vector $v \in T(f)$ such that $\varphi(v) < \varphi(\overline{v})$, then the function g defined by $T(g) = (T(f) \setminus \{v\}) \cup \{\overline{v}\}$ satisfies $\Phi(g) > \Phi(f)$. Since g is self-dual, $\Phi(f)$ is not maximum among all self-dual functions.

(iii) $\Longrightarrow$ (i): Assume that $\Phi(f)$ is not maximum among all self-dual functions. Then there exists a self-dual function g such that $\Phi(g) > \Phi(f)$. Since $\Phi(g) > \Phi(f)$, some vector $v \in T(f) \setminus T(g)$ satisfies $\varphi(v) < \varphi(\overline{v})$. For this v, let w be a vector in $\min_{\succeq} T(f)$ such that $w \preceq v$. By the g-regularity of φ, $\varphi(w) \le \varphi(v)$ holds. Moreover, since $\overline{w} \succeq \overline{v}$ by $w \preceq v$, $\varphi(\overline{w}) \ge \varphi(\overline{v})$ holds. Thus, we have

$$\varphi(w) \le \varphi(v) < \varphi(\overline{v}) \le \varphi(\overline{w}).$$

This means that this w satisfies $w \in \min_{\succeq} T(f)$ and $\varphi(w) < \varphi(\overline{w})$. ∎

THEOREM 6.2. *Let* $\Phi(f)$ *be a* g*-regular functional defined by* (24)*. Then there exists a* regular *self-dual function* f *which maximizes* $\Phi(f)$ *among all self-dual functions.*

Proof. Let f be a regular self-dual function that maximizes Φ among all *regular* self-dual functions. We claim that f in fact maximizes Φ among all self-dual functions.

If not, by Lemma 6.1, there exists a vector $v \in \min_{\succeq} T(f)$ such that $\varphi(v) < \varphi(\overline{v})$. Note that $v \not\succeq \overline{v}$ holds, since otherwise (i.e., $v \succeq \overline{v}$) $\varphi(v) \geq \varphi(\overline{v})$, a contradiction. Thus, it follows from Lemma 4.1 that $\rho_v(f)$ is regular and self-dual. Moreover, by (9), we have $\Phi(\rho_v(f)) > \Phi(f)$, which contradicts the assumption. □

However, there may be nonregular functions f that also maximize $\Phi(f)$. For example, let $p_1 = p_2 = 1/2$. Then $f_1 = x_1, f_2 = x_2, f_3 = \overline{x}_1$, and $f_4 = \overline{x}_2$ have the maximum availability, and, clearly, f_2, f_3, and f_4 are not regular. (f_3 and f_4 are not even positive.)

A pseudo-Boolean function φ is said to be *properly* g*-regular* if $\varphi(v) > \varphi(w)$ holds for all pairs of vectors v and w with $v \succ w$. The functional Φ defined by (24) is also said to be *properly* g*-regular* if φ is properly g-regular.

The next theorem is a weak uniqueness result for Theorem 6.2, showing that any optimal coterie is regular if $\Phi(f)$ is properly g-regular.

THEOREM 6.3. *Let $\Phi(f)$ be a* properly g*-regular functional defined by* (24)*. Then any function f that maximizes $\Phi(f)$ among all self-dual functions is regular.*

Proof. Assume that a nonregular function f maximizes $\Phi(f)$ among all self-dual functions. Since f is nonregular, there exists two vector v and w with $v \succ w$ such that $f(v) = 0$ and $f(w) = 1$. By Lemma 6.1, we have $\varphi(w) \geq \varphi(\overline{w})$. This, together with $v \succ w$ (and $\overline{w} \succ \overline{v}$), implies that $\varphi(v) > \varphi(\overline{v})$, which contradicts the assumption by Lemma 6.1. □

The above theorem directly implies the following corollary.

COROLLARY 6.4. *Let p_i, $i = 1, 2, \ldots, n$, be the operational probability of the ith element. If $p_1 > p_2 > \cdots > p_n > 1/2$, then any function f that maximizes the availability $A(f)$ among all self-dual functions is regular.*

Algorithm OPT-REG-SD

Input: A membership oracle of g-regular function φ.

Output: A regular self-dual function f that maximizes $\Phi(f)$ among all self-dual functions.

Step 0: Let $i := 1$ and $f := x_1$.

Step 1: While $\exists v \in \min_{\succeq} T(f)$ such that $v_i = 0$, $v[V_i] \not\succeq \overline{v}[V_i]$ and $\varphi(v') < \varphi(\overline{v'})$ for $v' = v + \sum_{j=i+1}^{n} e^{(j)}$, let $f := \sigma_{(v;V_i)}(f)$, where $V_i = \{1, 2, \ldots, i\}$.

Step 2: If $i = n$, output f and halt. Otherwise, $i := i + 1$ and return to Step 1.

Note that the set $\min_{\succeq} T(f)$ in the "while" statement of Step 1 is updated as a result of applying the σ transformation to f in Step 2.

Example 6.5. Let us consider the availability of the 6-variable functions, when $p_1 = 9/10$, $p_2 = 6/7$, $p_3 = 4/5$, $p_4 = 7/10$, and $p_5 = 3/5$. Recall that $\Phi(f) = A(f)$ is given by (25). We apply algorithm OPT-REG-SD to this $\Phi(f)$.

Step 0: Let $i := 1$ and $f := x_1$ (thus, $\min_{\succeq} T(f) = \{(10000)\}$). Let $u = (10000)$.

First iteration of Steps 1-2: Since $u_1 = 1$, skip Step 1. Step 2 sets $i := 2$.

Second iteration of Steps 1-2: Since $u[V_2] = (10) \succeq \overline{u}[V_2] = (01)$ for the only vector u in $\min_{\succeq} T(f)$, skip Step 1. Step 2 sets $i := 3$.

Third iteration of Steps 1-2: Vector $u \in \min_{\succeq} T(f)$ satisfies $u_3 = 0$, $u[V_3] = (100) \not\succeq \overline{u}[V_3] = (011)$, but $\varphi(10011) = 9/10 \times 1/7 \times 1/5 \times 7/10 \times 3/5 = 189/17500 > \varphi(01100) = 1/10 \times 6/7 \times 4/5 \times 3/10 \times 2/5 = 144/17500$. (If we were to apply $\sigma_{(u;V_3)}$ to f, we would have $\Phi(\sigma_{(u;V_3)}(f) < \Phi(f)$.) Thus skip Step 1. Step 2 sets $i := 4$.

Fourth iteration of Steps 1-2: Vector u satisfies $u_4 = 0$, $u[V_4] = (1000) \not\succeq \overline{u}[V_4] = (0111)$. Moreover, we have $\varphi(10001) < \varphi(01110)$, since $\varphi(10001) = 9/10 \times 1/7 \times 1/5 \times 3/10 \times 3/5 = 81/17500$ and $\varphi(01110) = 1/10 \times 6/7 \times 4/5 \times 7/10 \times$

$2/5 = 336/17500$. Thus f is transformed to

$$f := \sigma_{(u;V_4)}(f) = 12 + 13 + 14 + 234.$$

For this new f, we have $\min_{\succeq} T(f) = \{u = (10010), v = (01110)\}$. Since $u_4 = v_4 = 1$, we skip Step 1. Step 2 sets $i := 5$.

Fifth iteration of Steps 1-2: u satisfies $u_5 = 0$, $u[V_5] = (10010) \not\succeq \overline{u}[V_5] = (01101)$. Moreover, we have $\varphi(10010) < \varphi(01101)$, since $\varphi(10010) = 9/10 \times 1/7 \times 1/5 \times 7/10 \times 2/5 = 126/17500$ and $\varphi(01101) = 1/10 \times 6/7 \times 4/5 \times 3/10 \times 3/5 = 216/17500$. Thus f is transformed to

$$\begin{aligned} f := \sigma_{(u;V_5)}(f) &= 12 + 13 + 14(2 + 3 + 5) + 234 + 235 \\ &= 12 + 13 + 145 + 234 + 235. \end{aligned} \tag{26}$$

As a result, we have $\min_{\succeq} T(f) = \{u = (10100), v = (10011), w = (01101)\}$. Since $v_5 = w_5 = 1$, we need to consider only $u = (10100)$. Since $u_5 = 0$, $u[V_5] = (10100) \not\succeq \overline{u}[V_5] = (01011)$, but $\varphi(10100) = 216/17500 > \varphi(01011) = 126/17500$, skip Step 1. Since $i = n = 5$, output function f given by (26) and halt.

As mentioned in the introduction, the weights $(\log_2 9, \log_2 6, 2, \log_2(7/3), \log_2(3/2))$ defined by (1) produce an optimal vote-assignable coterie, since tie-breaking is unnecessary here. However, some weights are irrational, and hence we cannot exactly compute the sum of the weights $w^*(S) = \sum_{i \in S} w^*(i)$.

Let f_i, $i = 1, 2, \ldots, n$, be the function f after the ith iteration of Step 1 of algorithm OPT-REG-SD has been completed. Then clearly $V_{f_i} \subseteq V_i \, (= \{1, 2, \ldots, i\})$ holds. Moreover, we have the following lemma.

LEMMA 6.6. *Let f_i, $i = 1, 2, \ldots, n$, be as defined above. For each $i = 1, 2, \ldots, n$, all vectors $v \in \min_{\succeq} T(f_i)$ with $v[V_i] \not\succeq \overline{v}[V_i]$ satisfy*

$$\varphi(v') \geq \varphi(\overline{v'}), \tag{27}$$

where $v' = v + \sum_{j=i+1}^{n} e^{(j)}$.

Proof. If $i = 1$, then $f_1 = x_1$, and the lemma holds in this case, since $\min_{\succeq} T(f_1) = \{v = (100 \cdots 0)\}$ and $v[V_1] \succeq \overline{v}[V_1]$.

Assuming it holds for $i = k$, consider the case where $i = k + 1$. Let us consider the vector $v \in \min_{\succeq} T(f_{k+1})$ with $v[V_{k+1}] \not\succeq \overline{v}[V_{k+1}]$. If $v_{k+1} = 0$, then v satisfies $\varphi(v') \geq \varphi(\overline{v'})$, where $v' = v + \sum_{j=k+2}^{n} e^{(j)}$, since otherwise, v would have been removed from $T(f_{k+1})$ in Step 1 by applying the operation $\sigma_{(v;V_{k+1})}$ to f_{k+1}. This contradicts the definition of f_{k+1}.

If $v_{k+1} = 1$, on the other hand, there are two possibilities: (1) $f_k(v) = 0$ and (2) $f_k(v) = 1$. In case (1), since $v \in T(f_{k+1}) \setminus T(f_k)$, f was updated by $f := \sigma_{(\overline{v'};V_{k+1})}(f)$ in the $(k+1)$st iteration of Step 1. Therefore, we have $\varphi(v') \geq \varphi(\overline{v'})$.

In case (2), assuming $\varphi(v') < \varphi(\overline{v'})$, we derive a contradiction. $V_{f_k} \subseteq V_k$ implies $f_k(v - e^{(k+1)}) = 1$. Since $v[V_{k+1}] \not\succeq \overline{v}[V_{k+1}]$, we have $(v - e^{(k+1)})[V_k] \not\succeq \overline{(v - e^{(k+1)})}[V_k]$. Note that $(v - e^{(k+1)})' = v - e^{(k+1)} + \sum_{j=k+1}^{n} e^{(j)} = v'$. Thus, if $v - e^{(k+1)} \in \min_{\succeq} T(f_k)$, by assumption, f_k would have been updated by $f_k := \sigma_{(v - e^{(k+1)};V_k)}(f_k)$. This contradicts the definition of f_k. Now, since $v - e^{(k+1)} \notin \min_{\succeq} T(f_k)$, there exists a vector $w \in \min_{\succeq} T(f_k)$ such that $w \preceq v - e^{(k+1)}$. We can easily see that this w satisfies $w[V_k] \not\succeq \overline{w}[V_k]$ and $\varphi(w') < \varphi(\overline{w'})$, where $w' = w + \sum_{j=k+1}^{n} e^{(j)}$. This implies that f_k would have been updated by $f_k := \sigma_{(w;V_k)}(f_k)$, again a contradiction. ∎

Lemma 6.7. *Let f_n be as defined above. Then f_n maximizes Φ among all self-dual functions.*

Proof. Lemma 6.6 asserts (when $i = n$) that all vectors $v \in \min_{\succeq} T(f_n)$ with $v \not\succeq \overline{v}$ satisfy $\varphi(v) \geq \varphi(\overline{v})$. As for vectors $v \in \min_{\succeq} T(f_n)$ with $v \succeq \overline{v}$, the g-regularity of φ implies $\varphi(v) \geq \varphi(\overline{v})$. Together with Lemma 6.1, this completes the proof. □

Theorem 6.8. *Algorithm* OPT-REG-SD *correctly outputs a regular self-dual function f that maximizes Φ among all self-dual functions in $O(n^3|\min T(f)|)$ time.*

Proof. Since the algorithm's correctness follows from Lemma 6.7, we consider only its time complexity.

Let us assume that OPT-REG-SD generates the following sequence of functions: $f_m\,(= x_1) \to f_{m-1} \to \ldots \to f_0$, where f_0 is the output of algorithm OPT-REG-SD. Then there must be a sequence of transformations, $f_0 \to f_1 \to \ldots \to f_m$, which can be generated by algorithm TRANS-REG-SD. This means that $m \leq |\min T(f_0)|$.

As in algorithm GEN-REG-SD, we use binary trees B as the data structures of $\min T(f)$ and $\min_{\succeq} T(f)$ in the following way: For a vector v, define $\beta(v)$ by

$$\beta(v) = \min\left(\left\{ i \geq \max ON(v) \mid \varphi(v') < \varphi(\bar{v'}) \text{ for } v' = v + \sum_{j=i+1}^{n} e^{(j)} \right\} \cup \{n+1\}\right).$$

It is clear that $\varphi(v') < \varphi(\bar{v'})$ holds for $v' = v + \sum_{j=i+1}^{n} e^{(j)}$ with $i \geq \beta(v)$, if $\beta(v) \leq n$. Algorithm OPT-REG-SD then prepares $B(\min T(f))$ and $\{B(\min_{\succeq} T(f)_{(i,j)}) \mid i = 1, 2, \ldots, n+1, j = 1, 2, \ldots, n\}$, where

$$\min_{\succeq} T(f)_{(i,j)} = \{v \in \min_{\succeq} T(f) \mid i = \max\{\alpha(v), \beta(v)\},\ j = \max ON(v)\}.$$

As in the time complexity analysis of algorithm GEN-REG-SD, we can prove that each iteration of Step 1 in algorithm OPT-REG-SD can be executed in $O(n^3)$ time. Thus it requires $O(n^3|\min T(f)|)$ time in total. □

Acknowledgment. The authors thank the anonymous referees for their helpful and constructive comments which improved the presentation of this paper.

REFERENCES

[1] Y. Amir and A. Wool, *Optimal availability quorum systems: Theory and practice*, Inform. Process. Lett., 65 (1998), pp. 223–228.
[2] D. Avis and K. Fukuda, *A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra*, Discrete Comput. Geom., 8 (1992), pp. 295–313.
[3] D. Avis and K. Fukuda, *Reverse search for enumeration*, Discrete Appl. Math., 65 (1996), pp. 21–46.
[4] D. Barbara and H. Garcia-Molina, *The vulnerability of vote assignment*, ACM Trans. Computer Systems, 4 (1986), pp. 187–213.
[5] D. Barbara and H. Garcia-Molina, *The reliability of voting mechanisms*, IEEE Trans. Comput., 36 (1987), pp. 1197–1208.
[6] P. Bertolazzi and A. Sassano, *An $O(mn)$ time algorithm for regular set-covering problems*, Theoret. Comput. Sci., 54 (1987), pp. 237–247.
[7] L.J. Billera, *On the composition and decomposition of clutters*, J. Combinatorial Theory Ser. B, 11 (1971), pp. 234–245.
[8] J.C. Bioch and T. Ibaraki, *Complexity of identification and dualization of positive Boolean functions*, Inform. Comput., 123 (1995), pp. 50–63.
[9] J.C. Bioch and T. Ibaraki, *Generating and approximating positive non-dominated coteries*, IEEE Trans. Parallel Distrib. Systems, 6 (1995), pp. 905–914.
[10] E. Boros, P.L. Hammer, T. Ibaraki, and K. Kawakami, *Polynomial-time recognition of 2-monotonic positive Boolean functions given by an oracle*, SIAM J. Comput., 26 (1997), pp. 93–109.

[11] V. Chvátal and P.L. Hammer, *Aggregation of inequalities in integer programming*, Ann. Discrete Math., 1 (1977), pp. 145–162.

[12] Y. Crama, *Dualization of regular Boolean functions*, Discrete Appl. Math., 16 (1987), pp. 79–85.

[13] S.B. Davidson, *Replicated data and partition failures*, in Distributed Systems, S. Mullender, ed., Addison-Wesley, Reading, MA, 1989.

[14] S.B. Davidson, H. Garcia-Molina, and D. Skeen, Consistency in partitioned networks, ACM Comput. Surveys, 17 (1985), pp. 341–370.

[15] K. Diks, E. Kranakis, K. Krizanc, B. Mans, and A. Pelc, *Optimal coteries and voting schemes*, Inform. Process. Lett., 51 (1994), pp. 1–6.

[16] T. Eiter and G. Gottlob, *Identifying the minimal transversals of a hypergraph and related problems*, SIAM J. Comput., 24 (1995), pp. 1278–1304.

[17] M.L. Fredman and L. Khachiyan, *On the complexity of dualization of monotone disjunctive normal forms*, J. Algorithms, 21 (1996), pp. 618–628.

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

[19] D.K. Gifford, *Weighted voting for replicated data*, in Proceedings of the 7th Symposium on Operating System Principles, Pacific Grove, CA, ACM, 1979, pp. 150–162.

[20] T. Ibaraki and T. Kameda, *A theory of coteries: Mutual exclusion in distributed systems*, IEEE Trans. Parallel Distrib. Systems, 4 (1993), pp. 779–794.

[21] T. Ibaraki, H. Nagamochi, and T. Kameda, *Optimal coteries for rings and related networks*, in Proceedings of the 12th International Conference on Distributed Computing Systems, Yokohama, Japan, 1992, pp. 650–656.

[22] D.S. Johnson, M. Yannakakis, and C.H. Papadimitriou, *On generating all maximal independent sets*, Inform. Process. Lett., 27 (1988), pp. 119–123.

[23] L. Lamport, *Time, clocks, and the ordering of events in a distributed system*, Comm. ACM, 21 (1978), pp. 558–565.

[24] K. Makino and T. Ibaraki, *The maximum latency and identification of positive Boolean functions*, SIAM J. Comput., 26 (1997), pp. 1363–1383.

[25] K. Makino and T. Ibaraki, *A fast and simple algorithm for identifying* 2*-monotonic positive Boolean functions*, J. Algorithms, 26 (1998), pp. 291–305.

[26] K. Makino and T. Kameda, *Efficient generation of all regular non-dominated coteries*, in Proceedings of the Nineteenth ACM Symposium on Principles of Distributed Computing (PODC 2000), Portland, OR, 2000, pp. 279–288.

[27] S.J. Mullender and P.M.B. Vitányi, *Distributed match-making*, Algorithmica, 3 (1988), pp. 367–391.

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

[29] M. Naor and A. Wool, *The load, capacity and availability of quorum systems*, SIAM J. Comput., 27 (1998), pp. 423–447.

[30] M. Naor and A. Wool, *Access control and signatures via quorum secret sharing*, IEEE Trans. Parallel Distrib. Systems, 9 (1998), pp. 909–922.

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

[32] U.N. Peled and B. Simeone, *An* $O(nm)$*-time algorithm for computing the dual of a regular Boolean function*, Discrete Appl. Math., 49 (1994), pp. 309–323.

[33] D. Peleg and A. Wool, *The availability of quorum systems*, Inform. and Comput., 123 (1995), pp. 210–223.

[34] J.S. Provan and M.O. Ball, *Efficient recognition of matroids and* 2*-monotonic systems*, in Applications of Discrete Mathematics, R. Ringeisen and F. Roberts, eds., SIAM, Philadelphia, 1988, pp. 122–134.

[35] K.G. Ramamurthy, *Coherent Structures and Simple Games*, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1990.

[36] M. Spasojevic and P. Berman, *Voting as the optimal static pessimistic scheme for managing replicated data*, IEEE Trans. Parallel Distrib. Systems, 5 (1994), pp. 64–73.

[37] R.H. Thomas, *A majority consensus approach to concurrency control*, ACM Trans. Database Systems, 4 (1979), pp. 180–209.

[38] Z. Tong and R.Y. Kain, *Vote assignments in weighted voting mechanisms*, in Proceedings of the 7th Symposium on Reliable Distributed Systems, 1988, pp. 138–143.

[39] T.W. Yan and H. Garcia-Molina, *Distributed selective dissemination of information*, in Proceedings of the 3rd International Conference on Parallel and Distributed Information Systems, 1994, pp. 89–98.