DIMACS Technical Report 99-41
July 1999

Transformation of Regular Non-Dominated Coteries

by

Kazuhisa Makino[1] Tiko Kameda [2]

[1]Department of Systems and Human Science, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531, Japan. (makino@sys.es.osaka-u.ac.jp)
[2]School of Computing Science, Faculty of Applied Sciences, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada. (tiko@cs.sfu.ca)

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, Telcordia Technologies (formerly Bellcore) and NEC Research Institute.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

A coterie is a family of subsets such that every pair of subsets has at least one element in common but neither is a subset of the other. A coterie C is said to be non-dominated (ND) if there is no other coterie D such that, for $\forall Q \in C$, there exists $Q' \in D$ satisfying $Q' \subseteq Q$.

We introduce an operator σ, which transforms a ND coterie to another ND coterie. A regular coterie is a natural generalization of a "vote-assignable" coterie, which is used in some practical applications. We show that any "regular" ND coterie C can be transformed to any other regular ND coterie D by judiciously applying σ operations 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 "g-regular" function, as a generalization of availability. We show how to construct an optimum coterie C with respect to a g-regular function in $O(n^3|C|)$ time. We also discuss the structures of optimum coteries with respect to a g-regular function.

Keywords Coteries, Non-dominated coteries, Regular coteries, Availability, Mutual-exclusion, Positive self-dual Boolean functions, Regular self-dual Boolean functions, G-regular functions.

1 Introduction

A *coterie* C over 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 [26], selective dissemination of information [37], and distributed access control and signatures [28].

For example, to achieve mutual exclusion in a distributed system, let the elements in U represent the sites in the distributed system. A task 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 intersecting property of coterie C, it is guaranteed that at most one task can enter the critical section at any time.

A coterie C over U is said to *dominate* another coterie D ($\neq C$) over 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 *non-dominated* (ND) [18]. ND coteries are important in practical applications, since they have maximal "efficiency" in some sense [4, 18, 21].

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. 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, 35], where $\mathbb{R}^+$ is a nonnegative real number 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 (see the definition of threshold Boolean function in Section 2). Vote-assignable coteries are important and have been used in many practical problems, since they can be handled efficiently (see e.g., [18, 19, 35, 36]). 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 "essentially different" (i.e., non-equivalent 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 every $Q \in C$ and every pair (i, j) 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 the converse is not true in general. Regular Boolean functions were defined as a generalization of the threshold functions [27]. It is known that most regular coteries are vote-assignable [27], in particular, all regular ND coteries for $n \leq 9$ are vote-assignable.

Among the important problems regarding coteries are:

(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 (equivalently, construct an "optimal" positive self-dual function), and

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

Unfortunately, the complexity of question (i) is still unknown [8, 16, 22], although a result by M. Fredman and L. Khachiyan [17] suggests that it is unlikely that the problem is NP-hard. References [8, 16] give a

[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.

number of interesting equivalent problems, which arise in various fields of applications. However, it is known that, if we restrict ourselves to regular coteries, problem (i) is polynomially solvable [6, 31].

Although (i) is an interesting problem, we do not consider (i) in this paper. Instead, we focus on questions (ii) and (iii). As for (ii), let us consider the availability of a coterie. Assume that element i is *operational* with the probability p_i, while it is *failed* with the probability $1-p_i$, and the probabilities for different components are independent. Given operational probabilities p_i, $i \in U$, where we assume without loss of generality that $1 \geq p_1 \geq p_2 \geq \ldots \geq p_n \geq 0$, the availability of a coterie C is 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, 30, 34, 36]. It is known [1, 34] 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, 30]). Thus, we shall assume that

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

Moreover, it is 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 [34, 36] that the coterie C_{max}, given below, maximizes availability. First define the weight for $i \in U$ by

$$w^*(i) \quad = \quad \log_2(p_i/1-p_i),$$

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, [1, 34, 36] proposed algorithms to compute a vote assignment w from w^*, called *tie-breaking* (i.e., remove case (a)). An exponential algorithm is proposed in [36] to find the optimal tie-breaking rule, while [1, 34] presents 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 a maximum-availability coterie was known. In this paper, we present a polynomial time algorithm for it. More precisely, we define a "g-regular" function as a generalization of availability (see Section 6), and then show that, given a g-regular function 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 also mention that problem (iii) is mathematically interesting, giving us an insight into the structure of ND coteries (i.e., self-dual Boolean functions).

The generation of all ND coteries in a certain subclass of vote-assignable ND coteries was discussed in [27], which is used to give a lower bound on the number of all vote-assignable ND coteries. However, it is not polynomial and computes a proper subclass of vote-assignable ND coteries. H. Garcia-Molina and D. Barbara [18] proposed an algorithm to generate all ND coteries in a certain superclass of regular ND coteries. However, it is also not polynomial. J. C. Bioch and T. Ibaraki [9] later came up with a polynomial time algorithm to generate all ND coteries, and gave us a list containing all ND coteries of up to 7 elements, which are essentially different (i.e., non-equivalent under permutation). We remark here that their algorithm is not polynomial, if equivalent duplicates are to be deleted from the output. In fact, they made a list of all ND coteries over up to 7 elements by first running their algorithm and then selecting non-equivalent representatives from among them. In this paper, we present a polynomial algorithm to generate all *regular* ND coteries. Since no regular ND coteries C is equivalent to any other regular ND coterie $C' \, (\neq C)$ under permutation, our algorithm does not output ND coteries which are essentially the same. Although our algorithm outputs only regular ND coteries, it is practically useful, because all ND coteries of up to $n = 5$ elements are all regular (if we consider their representatives), and when n relatively is small, a large fraction of ND coteries are regular [27]. Moreover, if the objective function of problem (ii) is g-regular (e.g., the availability of a coterie), then we can restrict our attention to regular coteries.

After defining necessary terminologies in Section 2 (we use Boolean terminology, which is simpler than that of set theory), we discuss in Section 3 two operations, called ρ and σ, which transforms 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. These operations are used in [9, 18], where it is called *coterie transformation* in [18].

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 σ operations to f at most $|\min T(f)| + |\min T(g)| - 2$ times. 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 function 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 [27], operations research [6, 11, 29, 31], clutters in set theory [7], minimal transversals in hypergraphs [16], and coherent systems of reliability theory [33]. The results of this paper are relevant to all these problems.

2 Definitions and Basic Properties

A *Boolean function*, or a *function* in short, is a mapping $f : \{0,1\}^n \mapsto \{0,1\}$, 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 $T(f) = \emptyset$ and $F(f) = \emptyset$ are respectively denoted by $f = \perp$ and $f = \top$. 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 respectively referred to as *positive* and *negative* literals. 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, $t_1 = x_1 x_4 \overline{x}_5 x_6$ is a term, while $t_2 = x_2 x_4 \overline{x}_2$ 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 that implies f but no subterm of t implies f. 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 $t_v = x_1 x_3$. 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 $v \geq w$. Here, for any two functions f and g, we write $f \leq g$ if $T(f) \subseteq T(g)$. 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 $t_{\overline{v}} = x_2 x_4$.

It is known that a positive function f is uniquely determined by $\min T(f)$ (hence the input length of a positive function f is $n|\min T(f)|$, if it is represented in this manner), 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 $f = x_1 x_2 + x_2 x_3 + x_3 x_1$ by a simplified form $f = 12 + 23 + 31$, by 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 $(f + g)^d = f^d g^d$, $(fg)^d = f^d + g^d$, $f \leq g$ 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:

$$f(v) \quad \leq \quad f(v + e^{(i)} - e^{(j)}), \tag{1}$$

where $e^{(k)}$ denotes the unit vector which has a 1 in its k-th 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. For a vector $v \in \{0,1\}^n$, let us introduce the notation

$$\mathit{prof}_v(k) = \sum_{j \leq k} v_j,$$

where $k = 1, 2, \ldots, n$. If $v, w \in \{0,1\}^n$, where $v \neq w$, satisfy $\mathit{prof}_v(k) \leq \mathit{prof}_w(k)$ for all k, then we write $v \prec w$ (or $w \succ v$), and we say that v *supports* w. 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 Fig. 1.

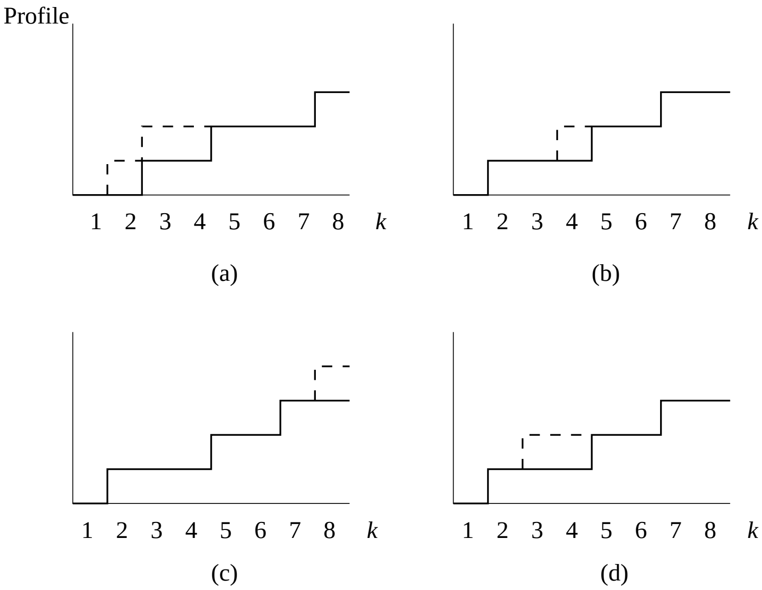

Figure 1: The profiles of different vectors: $\mathit{prof}_v(k)$ is shown in solid "curves."

In Fig. 1 (a), the profiles of $v = (00101001)$ and $w = (01100001)$ are depicted, in the solid curve and dashed curve, respectively. The dashed curve is not visible where it overlaps with the solid curve. It is easy to see that if v supports w, then the profile of w lies above the profile of v.

Note that $v \leq w$ implies $v \preceq w$. The following lemma is proved in [27].

Lemma 1 [27] *A function f is regular if and only if $f(v) \leq f(w)$ holds for all pairs of vectors v and w with $v \preceq w$.*

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 $\min T(f) \setminus \min_{\succeq} T(f)$ is supported by an element of $\min_{\succeq} T(f)$. It follows from Lemma 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, 27, 31], game theory [33], hypergraph theory [11] and learning theory [10, 24, 25]. The 2-monotonicity was originally introduced in conjunction with threshold functions (e.g., [27]), 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 non-negative 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 (1) 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 [27].

3 The operators ρ and σ and their properties

Let f be a positive function of n variables. Throughout this paper, we assume that f is *nontrivial* in the sense that $f \neq \perp, \top$ and $n \geq 1$. Given a vector $v \in \min T(f)$, the operation ρ_v 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,

$$T(\rho_v(f)) \quad = \quad (T(f) \setminus \{v\}) \cup T_{\geq}(\overline{v}), \tag{2}$$

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

$$\rho_v(f) \quad = \quad t_{\overline{v}} + t_v t_{\overline{v}}^d + f_{\setminus v}, \tag{3}$$

where $f_{\setminus v}$ denotes the function defined by all the prime implicants of f excluding t_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. The expression (3) is not necessarily in MDNF, even if $f_{\setminus v}$ is represented by its MDNF, as seen in Example 1, 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

$$\sigma_{(v;I)}(f) \quad = \quad t_{\overline{v[I]}} + t_{v[I]} t_{\overline{v[I]}}^d + f_{\setminus v}, \tag{4}$$

where $v[I]$ denote the *projection* of v on I; e.g., if $v = (1100)$, $I_1 = \{1, 2, 3\}$ and $I_2 = \{2, 3\}$, then $v[I_1] = (110)$ and $v[I_2] = (10)$. By definition, $\sigma_{(v;V)} = \rho_v$ holds. This operation $\sigma_{(v;I)}$ is implicitly used in [18].

Example 1 Consider a positive function of $n = 7$ variables:

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

For this function, we have $V_f = \{x_1, x_2, x_3, x_4, x_5\}$. For $v = (1100000)$ and $w = (0111000)$, we show below how operations ρ and σ are applied:

$$\rho_v(f) \quad = \quad 34567 + 12(3 + 4 + 5 + 6 + 7) + 13 + 145 + 234 + 235$$

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

□

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.,

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

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 variables $x_i\in J\setminus V$ as irrelevant ones. By definition, f and its expansion can be represented by the same DNF.

Since $I\supseteq V_f$, we have

$$\sigma_{(v;I)}(f) \quad = \quad Exp_V(\rho_v(Proj_I(f))). \tag{5}$$

Thus σ shares "reasonable" properties of ρ.

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 $I\subseteq V_f$).

Theorem 1 *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]. Since $\sigma_{(v;I)}(f)=Exp_V(\rho_v(Proj_I(f)))$ by (5), and $I\supseteq V_f$, the theorem is also valid for σ. □

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

$$T(\rho_v(f)) \quad = \quad (T(f)\setminus\{v\})\cup\{\overline{v}\}, \tag{6}$$

i.e., interchanging v with $\overline{v}$ in $T(f)$.

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

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

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 ρ and σ operations for regular self-dual functions, and give a transformation algorithm between two regular self-dual functions f and g, which satisfies

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

4 Transformation of regular self-dual functions

Lemma 2 *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 $v \not\succeq \overline{v}$.*

Proof. Let us first consider the only-if part. Recall that $\rho_v(f)$ can be specified by (6). Thus we have $\rho_v(f)(v) = 0$ and $\rho_v(f)(\overline{v}) = 1$. Since $\rho_v(f)$ is regular, this implies $v \not\succeq \overline{v}$. Moreover, since $v \in \min T(f)$, if $v \notin \min_{\succeq} T(f)$, then there exists a vector $w \in \min_{\succeq} T(f)$ supporting v, i.e., $w \prec v$. (See the paragraph after Lemma 1.) Now we have $\rho_v(f)(v) = 0$ and $\rho_v(f)(w) = 1$, which contradicts the regularity of $\rho_v(f)$. Thus $v \in \min_{\succeq} T(f)$ holds.

To show the if part, let us assume that $\rho_v(f)$ is not regular (i.e., there are two vectors u and w satisfying $\rho_v(f)(u) = 0$, $\rho_v(f)(w) = 1$ and $u \succ w$), but $v \in \min_{\succeq} T(f)$ and $v \not\succeq \overline{v}$ are satisfied. $\rho_v(w) = 1$ implies either $w = \overline{v}$ or $f(w) = 1$ by (6). If $w = \overline{v}$, then $u \succ w$ $(= \overline{v})$ implies $\overline{u} \prec v$. Since $v \in \min_{\succeq} T(f)$, we have $f(\overline{u}) = 0$. Thus the self-duality of f implies $f(u) = 1$. Now, since $f(u) = 1$ and $\rho_v(f)(u) = 0$, $u = v$ holds by (6). However, this implies $v \succ \overline{v}$, a contradiction. On the other hand, if $f(w) = 1$, the regularity of f implies $f(u) = 1$. Since $\rho_v(f)(u) = 0$, the only possibility is $u = v$. We thus have $f(w) = 1$, $v \in \min_{\succeq} T(f)$ and $v \succ w$, a contradiction. □

Lemma 3 *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 2 we have only to show $v \not\succeq \overline{v}$. Assuming $v \succeq \overline{v}$, we derive a contradiction. It follows from $v \succeq \overline{v}$ and $v_n = 1$ that $v \succeq \overline{v} + e^{(n)}$. Moreover, $n \geq 2$ implies $v \succ \overline{v} + e^{(n)}$. We have $f(v - e^{(n)}) = 0$ from $v \in \min_{\succeq} T(f)$. Hence $f(\overline{v} + e^{(n)}) = 1$ follows from the self-duality of f. Since $v \succ \overline{v} + e^{(n)}$, this contradicts $v \in \min_{\succeq} T(f)$. □

We remark here that, for a regular function f and an index i, x_i is relevant if and only if $V_f \supseteq \{1, 2, \ldots, i\}$. Hence x_n is relevant if and only if $V_f = \{1, 2, \ldots, n\}$. Moreover, this is equivalent to the existence of v in Lemma 3.

Lemma 4 *For a regular function f, x_n is relevant 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.

On the other hand, 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 have variable x_n, and hence x_n is irrelevant. Assume that $w \notin \min_{\succeq} T(f)$. Then there exists $u \in \min_{\succeq} T(f)$ such that $u \prec w$. We shall show $u_n = 1$, implying the only-if part. Since $u \prec w$, $|ON(u)| \leq |ON(w)|$ holds. Moreover, if $u_n = 0$, $|ON(u)| \leq |ON(w)| - 1$ holds by $w_n = 1$. This implies $w - e^{(n)} \succeq u$. By the regularity of f, we have $f(w - e^{(n)}) \geq f(u) = 1$, which is a contradiction to $w \in \min T(f)$. □

For a vector v, let us introduce the notation, $T_{\succ}(v) = \{w \mid w \succ v\}$ and $T_{\prec}(v) = \{w \mid w \prec v\}$. To further discuss the properties of $\rho_v(f)$ in Lemma 3, we first show the following proposition.

Proposition 1 *$u \in T_{\succ}(v)$ 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, $1 \leq i \leq n$, and $prof_u(k) = prof_v(k)$ for all $k \neq i$.*

Proof. For simplicity, we present an informal "picture proof", using Fig. 1. In Fig. 1 (b) and (c), it is clear that the vector v whose profile is represented by the solid curve supports the vector u whose profile is represented by the dashed curve, and that u is a minimal vector with respect to $\prec$ in $T_{\succ}(v)$. The dashed curve in Fig. 1 (d) shows a non-minimal vector w with $|ON(w)| = |ON(v)|$, since w is supported by another vector in $T_{\succ}(v)$. Similarly, it is easy to see that any member of $T_{\succ}(v)$ violating the conditions of this proposition is supported by another member of $T_{\succ}(v)$. □

Formula (7) in the following corollary follows immediately from the above proposition. Formula (8) is dual to (7).

Corollary 1

$$\min_{\succeq} T_{\succ}(v) = \begin{cases} \{v+e^{(n)}\} \cup \{v+e^{(j)}-e^{(j+1)} \mid v_j=0, v_{j+1}=1, 1\le j\le n-1\} & \text{if } v_n=0 \\ \{v+e^{(j)}-e^{(j+1)} \mid v_j=0, v_{j+1}=1, 1\le j\le n-1\} & \text{if } v_n=1. \end{cases} \quad (7)$$

$$\max_{\succeq} T_{\prec}(v) = \begin{cases} \{v-e^{(n)}\} \cup \{v-e^{(j)}+e^{(j+1)} \mid v_j=0, v_{j+1}=1, 1\le j\le n-1\} & \text{if } v_n=1 \\ \{v-e^{(j)}+e^{(j+1)} \mid v_j=0, v_{j+1}=1, 1\le j\le n-1\} & \text{if } v_n=0. \end{cases} \quad (8)$$

Lemma 5 *Let f be a regular self-dual function of $n\,(\ge 2)$ variables, and let $v \in \min_{\succeq} T(f)$ with $v_n = 1$. Then we have*

$$\min T(\rho_v(f)) = \min T(f) \setminus (\{v\} \cup \{\overline{v}+e^{(j)} \mid \max OFF(v) < j \le n\}) \cup \{\overline{v}\} \quad (9)$$

$$\begin{aligned} \min_{\succeq} T(\rho_v(f)) = {} & \min_{\succeq} T(f) \setminus (\{v\} \cup \min_{\succeq} T_{\succ}(\overline{v})) \cup \{\overline{v}\} \\ & \cup \{u \in \min_{\succeq} T_{\succ}(v) \mid u \not\succ z \text{ for all } z \in (\min_{\succeq} T(f) \setminus \{v\}) \cup \{\overline{v}\}\}. \end{aligned} \quad (10)$$

Proof. By (6), we need to consider only the influence of (i) adding $\overline{v}$ to $T(f)$ and (ii) removing v from $T(f)$.

(9) (i): $\overline{v} \in \min T(\rho_v(f))$ clearly holds. 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 $j > \max OFF(v)$. 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 $w \ne \overline{v}$ 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 $w \prec \overline{v}$. Note that $f(\overline{v}) = 0$ by the self-duality of f. Thus it follows from Lemma 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.

(9) (ii): It is clear that $v \notin \min T(\rho_v(f))$. Note that a vector $u\ (\ne v) \in T(f)$ satisfying $w \not< u$ for any vector $w \in \min T(f) \setminus \{v\}$ may be in $\min T(\rho_v(f))$. Such a vector, if any, has the form $u = v+e^{(j)}$, $j \in OFF(v)$. 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, obviously, there exists a $w \in \min T(f)$ satisfying $w \le v+e^{(j)}-e^{(n)} < v+e^{(j)}$, a contradiction.

(10) (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. Consider now any vector $w \in \min_{\succeq} T(f)$ supported by $\overline{v}$, 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 $\min_{\succeq} T_{\succ}(\overline{v})$.

(10) (ii): As noted in (9) (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 (7), 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 (10) (i) above) and all vectors in $\min_{\succeq} T_{\succ}(\overline{v})$ are supported by $\overline{v}$. □

From the proof of Lemma 5 (case (9)(i)), we can see that $\overline{v}+e^{(n)} \in \min T(f)$. Since $v_n = 1$ implies $n > \max OFF(v)$, (9) implies the following lemma.

Lemma 6 *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*

$$\begin{aligned} |\min T(\rho_v(f))| &\leq |\min T(f)| - 1, & (11)\\ \min T(\rho_v(f))_n \cup \{v, \overline{v} + e^{(n)}\} &= \min T(f)_n, & (12) \end{aligned}$$

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

If we repeatedly apply ρ_v operations to a regular self-dual function f, until there is no vector $v \in \min_{\succeq} T(f)$ with $v_n = 1$, by Lemmas 3, 4 and 6, we have a regular self-dual function f', in which x_n is irrelevant. This, together with (12) implies that $|\min T(f)_n|$ is even. Note that f' is not unique, i.e., it depends on the sequence of $v \in \min_{\succeq} T(f)$ with $v_n = 1$ that is used in ρ_v. For example, consider a function $f = 12 + 13 + 145 + 234 + 235$ of 5 variables. For vectors $v = (10011)$ and $w = (01101)$, $\rho_v(f) = 12 + 13 + 23$ and $\rho_w(f) = 12 + 13 + 14 + 234$.

Now $V_{f'} = \{1, 2, \ldots, j_1\}$ 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 discussed 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 $V_{f''} = \{1, 2, \ldots, j_2\}$ with $j_2 < j_1$. By repeating this argument, we reach the 1-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, $f_2, \ldots,$ $f_m\,(= x_1)$.

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 (11), 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 $\rho_{\overline{v}}(\rho_v(f)) = f$, we can transform x_1 into any regular self-dual function g by repeatedly applying σ operations to x_1 at most $|\min T(g)| - 1$ times. Thus we have the following theorem.

Theorem 2 *Let f and g be 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" function Φ (for the definition of g-regularity see Section 6), as applications of this transformation.

5 Generation of all regular ND coteries

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 operator σ. The algorithm is *incrementally polynomial* [22] in the sense that the i-th function $f_i \in \mathcal{C}_{R\text{-}SD}(n)$ is output in polynomial time in n and $\sum_{j=0}^{i-1} |\min T(f_j)|$, for $i = 1, 2, \ldots, |\mathcal{C}_{R\text{-}SD}|$.

To visualize the algorithm, 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 $I \supseteq V_g$. Theorem 2 implies that G_n is connected. For example, Figure 2 shows the graph G_5.

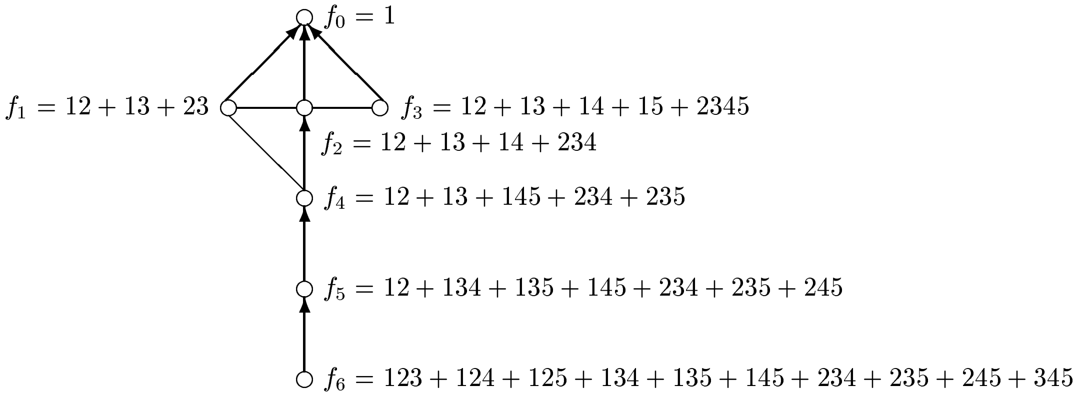

Figure 2: The graph G_5.

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 operation σ 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 at each operation the lexicographically smallest vector $v^* \in \min_{\succeq} T(g)$ such that $v^*_{\max V_g} = 1$.[3] Let μ be such an operation, i.e.,

$$\mu(g) \quad = \quad \sigma_{(v^*;V_g)}(g). \tag{13}$$

In this way, we define a directed spanning tree of G_n, $RT = (\mathcal{C}_{R\text{-}SD}(n), A_{RT})$, such that (g, f) is a directed arc in A_{RT} if and only if $\mu(g) = f$. Clearly, this RT is an in-tree rooted at $f_0 = x_1$. In Figure 2, A_{RT} is indicated by the bold arcs.

Our algorithm traverses RT from f_0 in the depth-first manner, outputting f whenever it first visits f. This type of enumeration algorithms is called *reverse search* in [2, 3] and have been applied to many other enumeration problems such as extreme points of a convex polyhedron, arrangements of hyperplanes, triangulations of a polygon, matroid bases and so on.

In order to traverse RT efficiently, we need the following lemmas. Lemma 7 is essentially a restatement of Lemma 5.

Lemma 7 *Let $f, g \in \mathcal{C}_{R\text{-}SD}(n)$ satisfy $f = \mu(g) = \sigma_{(v^*;V_g)}(g)$, where v^* is the lexicographically smallest vector in $\min_{\succeq} T(g)$ such that $v^*_{\max V_g} = 1$. Then we have*

$$\begin{aligned}
\min T(f) &= \min T(g) \setminus (\{v^*\} \ \cup \ \{w + e^{(j)} \mid \max(OFF(v^*) \cap V_g) < j \leq \max V_g\}) \ \cup \ \{w\}, && (14)\\
\min T(g) &= (\min T(f) \setminus \{w\}) \ \cup \ \{v^*\} \ \cup \ \{w + e^{(j)} \mid \max(ON(w) \cap V_g) < j \leq \max V_g\}, && (15)\\
\min_{\succeq} T(f) &= \min_{\succeq} T(g) \setminus (\{v^*\} \cup \min_{\succeq} T_{\succ}(w; V_g)) \ \cup \ \{w\} \\
&\quad \cup \{u \in \min_{\succeq} T_{\succ}(v^*; V_g) \mid u \not\succ z \ \textit{for all} \ z \in (\min_{\succeq} T(g) \setminus \{v^*\}) \cup \{w\}\}, && (16)\\
\min_{\succeq} T(g) &= \min_{\succeq} T(f) \setminus (\{w\} \cup \min_{\succeq} T_{\succ}(v^*; V_g)) \cup \{v^*\} \\
&\quad \cup \{u \in \min_{\succeq} T_{\succ}(w; V_g) \mid u \not\succ z \ \textit{for all} \ z \in (\min_{\succeq} T(f) \setminus \{w\}) \cup \{v^*\}\}, && (17)
\end{aligned}$$

where w is the vector satisfying $ON(w) = OFF(v^*) \cap V_g$, and for a vector v and an index set $I \subseteq V$, $T_{\succ}(v; I) = \{u \mid u \succ v, ON(u) \subseteq I\}$.

[3] For example, the lexicographic order of all vectors in $\{0, 1\}^3$ is (000), (001), (010), (011), (100), (101), (110), (111).

Proof. Follows from Lemma 5. □

Lemma 8 *Let $f, g \in \mathcal{C}_{R\text{-}SD}(n)$ satisfy $g = \sigma_{(w;V_g)}(f)$ for $w \in \min_{\succeq} T(f)$ (with $w[V_g] \not\succeq \overline{w}[V_g]$) and $V_g \supseteq V_f$. Then $f = \mu(g)\,(= \sigma_{(v^*;V_g)}(g))$ if and only if* (a) *$w_{\max V_g} = 0$,* (b) *$w_1 = 1$, and* (c) *the vector v^* (defined by $ON(v^*) = OFF(w) \cap V_g$) is lexicographically the smallest among all vectors $u \in \min_{\succeq} T(f)$ with $u_{\max V_g} = 1$.*

Proof. Let us first consider the only-if part. We clearly have $w_{\max V_g} = 0$ (i.e., (a) holds). Thus, it follows from (17) and (7) that

$$\min_{\succeq} T(g)_{x_{\max V_g}=1} = \begin{cases} (\min_{\succeq} T(f)_{x_{\max V_g}=1} \setminus \{v^* + e^{(j)} - e^{(j+1)} \mid v_j^* = 0, v_{j+1}^* = 1, 1 \le j \le \max V_g - 1\}) \\ \cup \{v^*\} \cup \{w + e^{(\max V_g)}\} \text{ if } w + e^{(\max V_g)} \not\succ u \text{ for all } u \in (\min_{\succeq} T(f) \setminus \{w\}) \cup \{v^*\} \\ (\min_{\succeq} T(f)_{x_{\max V_g}=1} \setminus \{v^* + e^{(j)} - e^{(j+1)} \mid v_j^* = 0, v_{j+1}^* = 1, 1 \le j \le \max V_g - 1\}) \\ \cup \{v^*\} \quad \text{otherwise.} \end{cases} \quad (18)$$

where $S_{x_i=1}$ denotes the set $\{v \in S \mid v_i = 1\}$. Now, since $f = \mu(g) = \sigma_{(v^*;V_g)}(g)$, v^* is lexicographically smaller than any vector in $\min_{\succeq} T(g)_{x_{\max V_g}=1}$. Moreover, since v^* is clearly lexicographically smaller than $u = v^* + e^{(j)} - e^{(j+1)}$ with $v_j^* = 0$ and $v_{j+1}^* = 1$, (18) implies (c).

Let us finally consider (b). If $w + e^{(\max V_g)} \in \min_{\succeq} T(g)$, then $w_1 = 1$ holds by (18), since v^* is lexicographically smaller than $w + e^{(\max V_g)}$. On the other hand, if $w + e^{(\max V_g)} \notin \min_{\succeq} T(g)$, there exists a vector $u \in \min_{\succeq} T(g)$ such that $u \prec w + e^{(\max V_g)}$ and $u_{\max V_g} = 1$ (the proof is similar to that of Lemma 4). If $w_1 = 0$, then we have $u_1 = 0$ (by $u \prec w + e^{(\max V_g)}$) and $v_1^* = 1$ (by $ON(v^*) = OFF(w) \cap V_g$). However, this implies that u is lexicographically smaller than v^*, contradicting $\mu(g) = \sigma_{(v^*;V_g)}(g)$. Therefore, (b) must hold.

The if part is implied by (18). □

Note that, if $V_g \ne V_f$ (i.e., $V_g \supset V_f$), Lemma 8 implies that $f = \mu(g)$ if and only if $w_1 = 1$. Thus, for an index set $I \supset V_f$, every vector $w \in \min_{\succeq} T(f)$, satisfying $w_1 = 1$ and $w[I] \not\succeq \overline{w}[I]$, always 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 $B(S)$. For example, Figure 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 $S := S \setminus \{v\}$) in $O(n)$ time, respectively. Moreover, since the right-most (resp., left-most) 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 $V_i = \{1, 2, \ldots, i\}$. Define $\alpha : \{0,1\}^n \to \mathbb{Z}^+$ by

$$\alpha(v) \quad = \quad \min(\{\, i \mid v[V_i] \not\succeq \overline{v}[V_i],\ ON(v) \subseteq V_i \subseteq V\} \cup \{n+1\}). \quad (19)$$

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

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

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

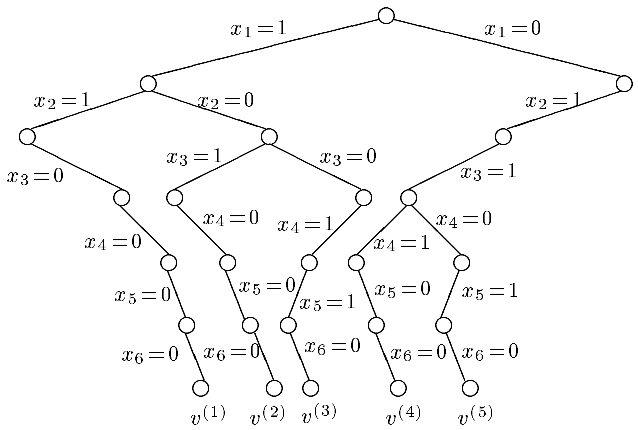

Figure 3: A binary tree $B(\min T(f))$, where $f = 12 + 13 + 145 + 234 + 235$ (of 6 variables).

During the depth-first search, at f, we first set up $I := V_f$ as the index set, in which the σ operator is applied. 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 $\min_{\succeq} T(f)_{\max I}$, according to the ordering $(i,j) = (1,1), \ldots, (1, \max I - 1), (2,1), \ldots, (2, \max I - 1), \ldots, (\max I, 1), \ldots, (\max I, \max I - 1)$. If there exists such a vector $w^{(i,j)}$, we go to $g = \sigma_{(w^{(i,j)};I)}(f)$. Since $w^{(i,j)}$ satisfies conditions (a), (b) and (c) of Lemma 8, 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 $w^{(i,j)}$ ($i = 1, 2, \ldots, \max I$, $j = 1, 2, \ldots, \max I - 1$) 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 8, then we go to $g' = \sigma_{(w';I)}(f)$; otherwise, check if $w^{(i,j)}$ satisfies (b) and (c), according to the ordering $(i,j) = (i^*, j^*+1), \ldots, (i^*, \max I - 1), (i^*+1, 1), \ldots, (i^*+1, \max I - 1), \ldots, (\max I, 1), \ldots, (\max I, \max I - 1)$.

This procedure has the advantage that there is no need to maintain the data of the entire search tree RT, but only the information about the current function is sufficient. Our algorithm can be written 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 $w = (11 \cdots 1)$, 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 applied 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 applied σ operation with $I \supset V_f$ previously, and hence we already applied 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 8, (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 $\min_{\succeq} T(f)_{\max I}$, according to the ordering $(i,j) = (1,1), \ldots, (1, \max I - 1), (2,1), \ldots, (2, \max I - 1), \ldots, (\max I, 1), \ldots, (\max I, \max I - 1)$. /* 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 $\max I < n$:[4] 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 8, 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 which are same in 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 which are same in Step 2. □

In case (ii), we check if there exist $I'' \supset I$ and $w'' \in \min_{\succeq} T(f)$ 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 after Lemma 8. According to the orderings on w' and w'', Algorithm GEN-REG-SD traverses RT depth-first, i.e., each arc in RT is traversed only twice, downward and upward.

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

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

Lemma 9 *Given $B(\min T(f))$, $B(\min_{\succeq} T(f)_{(i,j)})$ ($i = 1, 2, \ldots, n+1$, $j = 1, 2, \ldots, n$) and $B(\min_{\succeq} T(f)_j)$ ($j = 1, 2, \ldots, n$), each iteration of Steps 1 – 4 in Algorithm* GEN-REG-SD *computes either w' or w'' (or concludes that no such vectors exists) in $O(n^3)$ time.*

Proof. Since we can check if a given vector u satisfies (b) and (c) 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 10 *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 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, 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 only show the first statement in this Lemma, since the second one is similar to the first one. It directly follows from (14) in Lemma 7 that we can compute $B(\min T(f))$ in $O(n^2)$ time.

As for the rest, it is sufficient 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 (16) and (7). 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 (8), 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 [32]. This completes the proof. □

By Lemmas 9 and 10, 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 function in $\mathcal{C}_{R\text{-}SD}(n)$, i.e., $O(nM_{sum})$ time, where

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

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

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

Thus we have the following theorem.

Theorem 3 *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 2 *All functions in $\mathcal{C}_{R\text{-}SD}(n)$ can be scanned in $O(n^3|\mathcal{C}_{R\text{-}SD}(n)|)$ time.*

We mention here that regular functions are all *representatives*, i.e., there is no regular function f that is equivalent to the other regular function $g \, (\neq f)$ under permutation, where we say that f is *equivalent to g under permutation*, if permuting variables of f produces g. Therefore, our algorithm generates non-equivalent functions. Let us remark that the algorithms [18, 9] are not polynomial, if we try to output only non-equivalent functions.

It is known that ND coteries of up to $n = 5$ variables are all threshold functions (and hence regular, if we consider the representatives), but there are many non-regular ND coteries for $n \geq 6$, even if we consider the representatives. Moreover, it is known [27] that all regular ND coteries for $n \leq 9$ are threshold functions.

6 Optimum self-dual functions for regular functions Φ

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

$$\Phi(f) \quad = \quad \sum_{v \in T(f)} \varphi(v), \tag{20}$$

where φ is a pseudo Boolean function. Φ is also said to be g-*regular* if φ is g-regular. As an example of a g-regular pseudo Boolean function of interest, we cite the *availability* of a Boolean function, defined as follows. Each element $i \in V$ has the operational probability p_i $(0 \leq p_i \leq 1)$, i.e., the i-th element is *operational* with the probability p_i, while it is *failed* with the probability $1 - p_i$. We assume that the probabilities for different elements are independent. Then the *availability* $A(f)$ of a Boolean function f is defined by

$$A(f) \quad = \quad \sum_{v \in T(f)} \Big(\prod_{i \in ON(v)} p_i \prod_{i \in OFF(v)} (1 - p_i) \Big). \tag{21}$$

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 [33], especially, in the case where f represents ND coteries (i.e., f is positive self-dual) [1, 5, 15, 30, 34, 36]. It is known [1, 34] 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 $p_j \geq p_i$ for all i. Thus, we can assume that $p_i \geq 1/2$ holds for all i. Moreover, we assume without loss of generality that

$$p_1 \;\geq\; p_2 \;\geq\; \ldots \;\geq\; p_n \,(\geq\; 1/2).$$

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 maximizes g-regular function Φ among all self-dual functions.

Lemma 11 *Given a* g*-regular function* φ*, let* Φ *be a* g*-regular function defined by* (20)*. 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* $\varphi(v) \geq \varphi(\overline{v})$.

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

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, $\varphi(w) \leq \varphi(v)$ holds. Moreover, since $\overline{w} \succeq \overline{v}$ by $w \preceq v$, $\varphi(\overline{w}) \geq \varphi(\overline{v})$ holds. Thus, we have

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

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

Theorem 4 *Let $\Phi(f)$ be a* g*-regular function defined by* (20)*. 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 11, 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 2 that $\rho_v(f)$ is regular and self-dual. Moreover, by Eq. (6), we have $\Phi(\rho_v(f)) > \Phi(f)$, which contradicts the assumption. □

However, there may be non-regular 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 $\varphi(v) > \varphi(w)$ always holds for all pairs of vectors v and w with $v \succ w$. Φ defined by Eq. (20) is also said to be *properly* g*-regular* if φ is properly g-regular.

Theorem 5 *Let $\Phi(f)$ be a* properly g*-regular function defined by Eq.* (20)*. Then any function f that maximizes $\Phi(f)$ among all self-dual functions is regular.*

Proof. Assume that non-regular function f maximizes $\Phi(f)$ among all self-dual functions. Since f is non-regular, there exists two vector v and w with $v \succ w$ such that $f(v) = 0$ and $f(w) = 1$. By Lemma 11, 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 11. □

The above theorem directly implies the following corollary.

Corollary 3 *Let p_i, $i = 1, 2, \ldots, n$, be the operational probability of the i-th element. If $p_1 > p_2 > \ldots > p_n > 1/2$, 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 2 is updated as a result of applying the σ transformation to f in Step 2.

Example 2 Let us consider the availability of the 6-variable functions, where $p_1 = 9/10, p_2 = 6/7, p_3 = 4/5$, $p_4 = 7/10$ and $p_5 = 3/5$. Recall that $\Phi(f) = A(f)$ is given by Eq. (21). 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

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

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

$$\begin{aligned} f := \sigma_{(u;V_5)}(f) &= 12 + 13 + 14(2+3+5) + 234 + 235 \\ &= 12 + 13 + 145 + 234 + 235. \end{aligned} \tag{22}$$

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 (22) and halt. □

Let f_i, $i = 1, 2, \ldots, n$, be the function f after the i-th 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 12 *Let f_i, $i = 1, 2, \ldots, n$, be as defined above. For all $i = 1, 2, \ldots, n$, all vectors $v \in \min_{\succeq} T(f_i)$ with $v[V_i] \not\succeq \overline{v}[V_i]$ satisfy*

$$\varphi(v') \quad \geq \quad \varphi(\overline{v'}), \tag{23}$$

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

Proof. If $i = 1$, $f_1 = x_1$, and the lemma holds in this case, since $\min_{\succeq} T(f_1) = \{v = (100\cdots0)\}$ 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}.

On the other hand, if $v_{k+1} = 1$, 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 13 *Let f_n be defined as above. Then f_n maximizes Φ among all self-dual functions.*

Proof. Lemma 12 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 11, this completes the proof. □

Theorem 6 *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 13, we only consider its time complexity.

Let us assume that OPT-REG-SD generates the following sequence of functions: $f_m\ (= x_1) \rightarrow f_{m-1} \rightarrow \ldots \rightarrow f_0$, where f_0 is the output of Algorithm OPT-REG-SD. Then there must be a sequence of transformations, $f_0 \rightarrow f_1 \rightarrow \ldots \rightarrow f_m$, 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

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

It is clear that $\varphi(v') < \varphi(\overline{v'})$ 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 $\bigcup_{\substack{i=1,2,\ldots,n+1 \\ j=1,2,\ldots,n}} B(\min_{\succeq} T(f)_{(i,j)})$, where

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

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. □

7 Conclusion

We have introduced a new operator σ, which is similar to the ρ operator used by Bioch and Ibaraki [8]. It transforms a non-dominated (ND) coterie to another ND coterie. We showed that any regular ND coterie can be transformed to any other regular ND by a sequence of σ operations, and tried to find the shortest such sequence. As another application of the σ operation, we presented an incrementally-polynomial-time algorithm for generating all regular ND coteries, and showed that we can construct an optimum coterie C with respect to a "g-regular" function in polynomial time.

The challenging problem of deciding whether a given coterie is ND is still open.

References

[1] Y. Amir and A. Wool, Optimal availability quorum systems: theory and practice, *Information Processing Letters* 65 (1998) 223-228.

[2] D. Avis and K. Fukuda, A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra, *Discrete and Computational Geometry*, 8 (1992) 295-313.

[3] D. Avis and K. Fukuda, Reverse search for enumeration, *Discrete Applied Mathematics* 65 (1996) 21-46.

[4] D. Barbara and H. Garcia-Molina, The vulnerability of vote assignment, *ACM Trans. on Computer Systems*, 4 (1986) 187-213.

[5] D. Barbara and H. Garcia-Molina, The reliability of voting mechanisms, *IEEE Trans. on Computers*, 36 (1987) 1197-1208.

[6] P. Bertolazzi and A. Sassano, An $O(mn)$ time algorithm for regular set-covering problems, *Theoretical Computer Science*, 54 (1987) 237-247.

[7] L.J. Billera, On the composition and decomposition of clutters, *J. of Combinatorial Theory*, 11 (1971) 234-245.

[8] J.C. Bioch and T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, *Information and Computation*, 123 (1995) 50-63.

[9] J. C. Bioch and T. Ibaraki, Generating and approximating positive non-dominated coteries, *IEEE Transactions on Parallel and Distributed Systems*, 6 (1995) 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. Computing*, 26 (1997) 93-109.

[11] V. Chvátal and P.L. Hammer, Aggregation of inequalities in integer programming, *Annals of Discrete Mathematics*, 1 (1977) 145-162.

[12] Y. Crama, Dualization of regular Boolean functions, *Discrete Applied Mathematics*, 16 (1987) 79-85.

[13] S.B. Davidson, Replicated data and partition failures, in: S. Mullender (ed.), *Distributed Systems*, Ch.13, Addison Wesley, 1989.

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

[15] K. Diks, E. Kranakis, K. Krizanc, B. Mans adn A. Pelc, Optimal coteries and voting schemes, *Information Processing Letters*, 51 (1994) 1-6.

[16] T. Eiter and G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, *SIAM Journal on Computing*, 24 (1995) 1278-1304.

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

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

[19] D.K. Gifford, Weighted voting for replicated data, *Proc. the 7th Symposium on Operating System Princiles*, Pacific Grove, Calif., Dec. 1979, 150-162, ACM, New York.

[20] T. Ibaraki and T. Kameda, Theory of coteries: Mutual exclusion in distributed systems, *IEEE Trans. on Parallel and Distributed Syst.*, 4 (1993) 779-794.

[21] T. Ibaraki, H. Nagamochi and T. Kameda, Optimal coteries for rings and related networks, *Proc. 12th International Conference on Distributed Computing Systems*, Yokohama, June 1992, 650-656.

[22] D.S. Johnson, M. Yannakakis and C.H. Papadimitriou, On generating all maximal independent sets, *Information Processing Letters*, 27 (1988) 119-123.

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

[24] K. Makino and T. Ibaraki, The maximum latency and identification of positive Boolean functions, *SIAM Journal on Computing*, 26 (1997) 1363-1383.

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

[26] S.J. Mullender and P.M.B. Vitányi, Distirbuted match-making, *Algorithmica*, 3 (1988) 367-391.

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

[28] M. Naor and A. Wool, Access control and signatures via quorum secret sharing, *IEEE Trans. Parallel and Distributed Syst.*, 9 (1998) 909-922.

[29] U.N. Peled and B. Simeone, Polynomial-time algorithm for regular set-covering and threshold synthesis, *Discrete Applied Mathematics*, 12 (1985) 57-69.

[30] D. Peleg and A. Wool, The availability of quorum systems, *Information and Computation* 123 (1995) 210-223.

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

[32] 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.), 122-134, SIAM, Philadelphia, 1988.

[33] K. G. Ramamurthy, *Coherent Structures and Simple Games*, Kluwer Academic Publishers, 1990.

[34] M. Spasojevic and P. Berman, Voting as the optimal static pessimistic scheme for managing replicated data, *IEEE Trans. Parallel and Distributed Syst.* 5 (1994) 64-73.

[35] R.H. Thomas, A majority consensus approach to concurrency control, *ACM Trans. Database Syst.*, 4 (1979) 180-209.

[36] Z. Tong and R.Y. Kain, Vote assignments in weighted voting mechanisms, *Proc. 7th Symp. on Reliable Distributed Systems*, Oct. 1988, 138-143.

[37] T.W. Yan and H. Garcia-Molina, Distirbuted selective dissemination of information, *Proc. 3rd Intn'l. Conf. Par. Dist. Info. Syst.*, 1994, 89-98.