数理解析研究所講究録
1185 巻 2001 年 130-139

Transformations on Regular Non-Dominated Coteries and Their Application

Kazuhisa Makino[1] and Tiko Kameda[2]

[1] Division of Systems 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

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 non-dominated (ND) if there is no other coterie D under U such that, for $\forall Q\in C$, 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 g-*regular* functional, 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, where $n=|U|$. Finally, we discuss the structures of optimum coteries with respect to a g-regular functional.

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

1 Introduction

A *coterie* C under an underlying set $U=\{1,2,\ldots,n\}$ 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) [8, 11]. The concept of a coterie has applications in diverse areas (see e.g., [6, 8, 11, 15]).

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 intersection property, it is guaranteed that at most one task can enter the critical section at any time.

A coterie D is said to *dominate* another coterie C if, for $\forall Q\in C$, there exists a quorum $Q'\in D$ satisfying $Q'\subseteq Q$ [8]. A coterie C is *non-dominated* (ND) if no other coterie dominates it. ND coteries are important in practical applications, since they have maximal "efficiency" in some sense [3, 8].

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 [10] that C is a coterie (resp., ND coterie) if and only if f_C is *dual-minor* (resp., *self-dual*) [14]. Based on this characterization, Boolean algebra can be exploited to derive var-

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

ious properties of (ND) coteries.

A coterie C is said to be *vote-assignable* if there exist a *vote assignment* $w : U \mapsto \mathbf{R}^+$ and a *threshold* $t \in \mathbf{R}^+$ such that $w(S) \geq t$ if and only if $S \supseteq Q$ for some $Q \in C$ [8, 9, 18], where $\mathbf{R}^+$ 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 (see Section 2). The vote-assignable coteries are important and have been used in many practical applications, since they can be handled efficiently (see e.g., [8, 9, 18, 19]). 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 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. A coterie C is said to be *regular* if, for every $Q \in C$ and every pair $(i, j) \in U \times U$ with $i < j$, $i \notin Q$ and $j \in Q$, there exists quorum $Q' \in C$ such that $Q' \subseteq (Q \setminus \{j\}) \cup \{i\}$.[2] By definition, a vote-assignable coterie C is always regular, though the converse is not true in general. It is known that most regular coteries are vote-assignable [14], in particular, all regular ND coteries for $n \leq 9$ are vote-assignable.

Among the important problems regarding coteries are:

(i) construct "optimal" ND coteries according to a certain criterion, such as availability and load (equivalently, construct an "optimal" positive self-dual function), and

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

As for (i), let us consider the availability of a coterie. Assume that element i is *operational* with probability p_i, where the probabilities for different components are independent. Given the 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 clearly an important concept in practical applications, and it is desirable to construct a coterie with the maximum availability.

The availability of coteries has been studied extensively. It is known [1, 17] 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, 7, 16]). Thus, we shall assume that

$$p_1 \geq p_2 \geq \ldots \geq p_n \geq 1/2.$$

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 [17, 19] that the coterie C_{max}, given below, maximizes availability. First define the weight for $i \in U$ by

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

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 with $w^*(Q) > w^*(U)/2$, and Q does not contain any quorum of type (a).

Since this coterie C_{max} is vote-assignable, [1, 17, 19] proposed algorithms to compute a vote assignment w from w^*, called *tie-breaking*, in order to remove case (a). An exponential algorithm is proposed in [19] to find the "optimal" tie-breaking rule, while [1, 17] 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 a maximum-availability coterie 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 5), and then show that, given

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

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 (ii) is known to be useful to solve (i) [5, 8]. To solve (i), one might first enumerate all (or some) ND coteries efficiently, and select the best one under a certain criterion, which may not be easily computable. This procedure is useful when n is small, or when we have enough time to compute it.

The generation of all ND coteries in a certain subclass of vote-assignable ND coteries was discussed in [14], 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. H. Garcia-Molina and D. Barbara [8] 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 [5] later came up with a polynomial time algorithm to generate all ND coteries, and compiled a list containing all ND coteries under up to 7 elements, which are 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 compiled a list of all ND coteries under 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 coterie C is equivalent to any other regular ND coterie $C' (\neq C)$ under permutation, 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 up to $n = 5$ elements are all regular (if we consider their representatives), and when n is relatively is small, a large fraction of ND coteries are regular [14]. Moreover, if the objective function of problem (i) 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 discuss in Section 3 two operations, called ρ and σ, which transform a 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.

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 computing an optimal self-dual function with respect to a g-regular functional Φ and generating all regular self-dual functions, 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 learning theory, operations research and set theory. The results of this paper are relevant to all these areas.

Due to the space limitation, the proofs of some results are omitted (see [12, 13]).

2 Preliminaries

A *Boolean function* (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., $F(f)$). For any two functions f and g, we say that f *is covered by* g (written $f \leq g$) 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 $\overline{x}_i$. 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 $P(t) \cap N(t) = \emptyset$; E.g., $t_1 = x_1 x_4 \overline{x}_5 x_6$ is a term, while $t_2 = x_2 x_4 \overline{x}_2$ is not. 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 $v \neq w$. 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, $\min_{\geq} T(f)$ (resp., $\max_{\geq} F(f)$) denotes the set of all minimal true vectors (resp., 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 t that implies f but no proper 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 $t_v = x_1 x_3$. 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 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 $f = x_1x_2 + x_2x_3 + x_3x_1$ by a simplified form $f = 12+23+31$, by using only the subscripts of the literals. Coteries can be conveniently modeled by positive Boolean functions, based on the fact that $\min T(f)$ can represent a family of subsets, none of which includes the other [10].

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. It is easy to see that $(f+g)^d = f^d g^d$, $(fg)^d = f^d + g^d$, 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 $f = f^d$. For example, $f = 123$ is dual-minor since $f^d = 1+2+3$ satisfies $f \leq f^d$.

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 $\mathcal{C}_{SD}(n)$ (resp., $\mathcal{C}_{DMA}(n)$ and $\mathcal{C}_{DMI}(n)$) denote the class of all positive self-dual (resp., dual-major and 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's 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 v *supports* w. If $v \prec w$ or $v = w$, then we write $v \preceq w$ (or $w \succeq v$).

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 [14].

Lemma 1 ([14]) *A function f is regular if and only if f is* profile-monotone.

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 $v \succeq w$. 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 was originally introduced in conjunction with threshold functions (e.g., [14]), 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:

$$f(x) \;=\; 1 \text{ if and only if } \sum w_i x_i \geq t. \quad (2)$$

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 [14].

3 The operators ρ and σ

Let f be a positive function of n variables. Throughout this paper, we assume that f is *nontrivial* in the sense that $f \not\equiv 0, 1$ 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 $T(f)$ [5]. More precisely, while adding $\overline{v}$, all the vectors larger than $\overline{v}$ are also added to $T(f)$. Therefore,

$$T(\rho_v(f)) \;=\; (T(f) \setminus \{v\}) \cup T_{\geq}(\overline{v}), \quad (3)$$

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

$$\rho_v(f) \;=\; f_{\setminus v} + t_{\overline{v}} + t_v t^d_{\overline{v}}, \quad (4)$$

where $f_{\setminus v}$ denotes the function defined by all the prime implicants of f except t_v, and $t^d_{\overline{v}}$ 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

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

represents all the vectors larger than v. The expression (4) is not necessarily in MDNF, even if $f_{\setminus v}$ is represented by its MDNF, because some of the prime implicants in $t_{\overline{v}} + t_v t^d_{\overline{v}}$ may cover or may be covered by some prime implicants of $f_{\setminus v}$.

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

$$\sigma_{(v;I)}(f) \;=\; f_{\setminus v} + t_{\overline{v[I]}} + t_{v[I]} t^d_{\overline{v[I]}}, \quad (5)$$

where $v[I]$ denotes 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 [8].

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 $x_i \in V \setminus I$.

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 MDNF. Since $I \supseteq V_f$, we have

$$\sigma_{(v;I)}(f) \;=\; Exp_V(\rho_{v[I]}(Proj_I(f))). \quad (6)$$

Thus σ has properties similar to those 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 $\mathcal{C}_{DMI}(n)$.*

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

$$T(\rho_v(f)) \;=\; (T(f) \setminus \{v\}) \cup \{\overline{v}\}, \quad (7)$$

i.e., by interchanging v with $\overline{v}$ in $T(f)$ [5]. To see the effect of $\sigma_{(v;I)}$ on $T(f)$, where $V_f \subseteq I \subseteq V$, define

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

It is easy to see that

$$T(\sigma_{(v;I)}(f)) \;=\; (T(f) \setminus v[I]\underline{*}) \cup \overline{v}[I]\underline{*}. \quad (8)$$

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

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

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 on regular self-dual functions, and give a transformation algorithm between two regular self-dual functions f and g, which satisfies

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

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 regularity if v satisfies a certain condition. (We have already seen that ρ_v preserves self-duality.) Recall that $\rho_v(f)$ is specified by (7), and therefore, we concentrate on the vectors v and $\overline{v}$.

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 $\overline{v} \not\prec v$.*

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

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.*

Interestingly, the existence of v satisfying the condition in Lemma 3 is equivalent the relevance of x_n to f, as proved in the following Lemma 4.

Lemma 4 *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 $v_n = 1$.*

Lemma 3 deals with the case where x_n is relevant to f. To deal with the case where x_n is irrelevant to f, note that for any $i, j \in V$ such that $i < j$, if x_j is relevant to a regular function f then so is x_i. This 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 1 below generalizes Lemma 3 to the case where x_n may be irrelevant to f.

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

We now have the theoretical foundation for TRANS-REG-SD. By Lemma 3 and Corollary 1, 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 σ transformations 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. 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 $\min_{\succeq} T(f')$. For a vector v, let us introduce the notation, $T_{\succ}(v) = \{w \mid w \succ v\}$ and $T_{\prec}(v) = \{w \mid w \prec v\}$.

Lemma 5 *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*

$$\begin{aligned} \min T(\rho_v(f)) &= \min T(f) \setminus (\{v\} \cup \{\overline{v} + e^{(j)} \mid \max \\ &\qquad OFF(v) < j \leq n\}) \cup \{\overline{v}\} \qquad (9) \\ \min_{\succeq} T(\rho_v(f)) &= \min_{\succeq} T(f) \setminus (\{v\} \cup \min_{\succeq} T_{\succ}(\overline{v})) \cup \\ &\qquad \{\overline{v}\} \cup \{u \in \min_{\succeq} T_{\succ}(v) \mid u \not\succ z \text{ for all } z \in \\ &\qquad (\min_{\succeq} T(f) \setminus \{v\}) \cup \{\overline{v}\}\}. \qquad (10) \end{aligned}$$

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 $n > \max OFF(v)$, $\{\overline{v} + e^{(j)} \mid \max OFF(v) < j \leq n\}$ is non-empty, and (9) implies Lemma 6.

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*

$$|\min T(\rho_v(f))| \leq |\min T(f)| - 1 \quad (11)$$

$$\min T(\rho_v(f))_n \cup \{v, \overline{v} + e^{(n)}\} = \min T(f)_n, \quad (12)$$

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

We are now ready to describe the transformation algorithm. If we repeatedly apply ρ_v operations (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 3, 4 and 6, we have a regular self-dual function f', to which x_n is irrelevant. Note that f' may not be unique, i.e., it in general depends on the sequence of vectors $v \in \min_{\succeq} T(f)$ with $v_n = 1$ that are used in ρ_v.

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 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 $V_{f''} = \{1, 2, \ldots, j_2\}$ 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 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 (7)), 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 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 study some applications of algorithm TRANS-REG-SD.

5 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 $\mathbf{R}$. φ is said to be g-*regular* if it is profile-monotone, i.e., $\varphi(v) \geq \varphi(w)$ holds for all pairs of vectors v and w with $v \succ w$. Define a functional $\Phi()$ of Boolean functions f as follows:

$$\Phi(f) \;=\; \sum_{v \in T(f)} \varphi(v), \qquad (13)$$

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 functional of interest, we cite the *availability* $A(f)$ of a Boolean function f. Assume that 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 probability p_i. We also assume that the probabilities for different elements are independent. Then the *availability* of a Boolean function f is defined by

$$A(f) \;=\; \sum_{v \in T(f)} \Big(\prod_{i \in ON(v)} p_i \prod_{i \in OFF(v)} (1 - p_i) \Big) \qquad (14)$$

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. Availability has been studied extensively, especially, in the case where f represents a ND coterie (i.e., f is positive self-dual) [1, 4, 7, 16, 17, 19]. As commented in the Introduction, we can 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 maximize g-regular functional Φ among all self-dual functions.

Lemma 7 *Given a g-regular function φ, let Φ be a g-regular functional defined by (13). 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 $\varphi(v) \geq \varphi(\overline{v})$.*

Theorem 3 *Let $\Phi(f)$ be a g-regular functional defined by (13). 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 7, 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. (7), we have $\Phi(\rho_v(f)) > \Phi(f)$, which contradicts the assumption. □

However, there may be non-regular functions f that also maximize $\Phi(f)$.

Based on Theorem 3, the following algorithm computes an optimum regular self-dual function.

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 $v' = v + \sum_{j=i+1}^{n} e^{(j)}$, do $f := \sigma_{(v;V_i)}(f)$, where $V_i = \{1, 2, \ldots, i\}$.

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

Let f_i, $i = 1, 2, \ldots, n$, be the function f after the i-th iteration of Step 1 of 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 8 *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*

$$\varphi(v') \;\geq\; \varphi(\overline{v'}), \tag{15}$$

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

Lemma 9 *Let f_n be as defined above. Then f_n maximizes Φ among all self-dual functions.*

Therefore, OPT-REG-SD computes an optimum function $f\,(= f_n)$. Moreover, it requires polynomial time in n and the size of f. (Due to the space limitation, we omit the proof. See [12]).

Theorem 4 *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.*

6 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* in the sense that the i-th 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 $I \supseteq V_g$. Figure 1 shows the graph G_5. (Ignore the arrows on some edges).

Theorem 2 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 undirected. Let $f_0 = x_1$ be the designated function in $\mathcal{C}_{R\text{-}SD}(n)$, and consider the problem of transforming an arbitrary

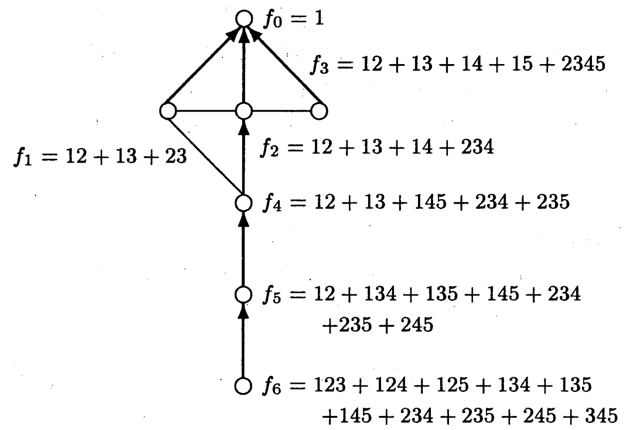

Figure 1: The graph G_5.

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

$$\mu(g) \;=\; \sigma_{(\tilde{v};V_g)}(g). \qquad (16)$$

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 1, A_{RT} is indicated by the thick arcs.

Our algorithm will traverse RT from f_0 in a depth-first manner, outputting each regular function f when it first visits f. This type of enumeration is called *reverse search* in [2]. When RT is traversed from f_0, for each arc $(g, f) \in A_{RT}$, the end node f is visited first, i.e., before g. Unfortunately, at f we cannot distinguish between the arcs in A_{RT} and 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 (16) computes f given g, not the other way around. In Lemma 10 below, we find the "inverse" of (16) in the sense that we find the conditions on the choice of $w \in \min_{\succeq} T(f)$ such that $g = \sigma_{(w;V_g)}(f)$.

For a vector $v \in \{0,1\}^n$ and $I \subseteq V$, let $v[I]\underline{0}$ denote the vector u defined by $ON(u) = OFF(v) \cap I$, i.e., $u[I] = v[I]$ and the remaining components of u, if any, are all set to 0's.

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

Lemma 10 *Let $f \in \mathcal{C}_{R\text{-}SD}(n)$ and $g = \sigma_{(w;V_g)}(f)$ for $w \in \min_{\succeq} T(f)$ such that $\overline{w}[V_g] \not\succeq w[V_g]$ and $V_g \supseteq V_f$. Then $f = \mu(g)\,(= \sigma_{(\tilde{v};V_g)}(g))$ 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 $u \in \min_{\succeq} T(f)$ with $u_{\max V_g} = 1$.*

Note that, if $V_g \neq V_f$ (i.e., $V_g \supset V_f$), Lemma 10 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 (a). Thus, for an index set $I \supset V_f$, every vector $w \in \min_{\succeq} T(f)$ which satisfies $w_1 = 1$ and $\overline{w}[I] \not\succeq w[I]$ always produces $g = \sigma_{(w;I)}(f)$ such that $f = \mu(g)$.

Let

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

Although the details are omitted due to the space limitation, we have the following results [12].

Theorem 5 *All functions in $\mathcal{C}_{R\text{-}SD}(n)$ can be generated in incrementally polynomial time. 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 reiterate here that regular functions are all *representatives* of a permutation class, i.e., no regular function f is equivalent to another regular function $g\,(\neq f)$ under permutation. Therefore, our algorithm generates non-equivalent functions. Let us remark that the algorithms in [5, 8] are not polynomial, if we try to output only non-equivalent 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 non-regular self-dual functions for $n \geq 6$, even if we consider the representatives. Moreover, it is known [14] that all regular self-dual functions for $n \leq 9$ are threshold functions.

7 Conclusion

We have introduced a new operator σ, which is similar to the ρ operator used by Bioch and Ibaraki [5]. 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, Reverse search for enumeration, *Discrete Applied Mathematics*, 65 (1996) 21-46.

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

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

[5] J. C. Bioch and T. Ibaraki, Generating and approximating positive non-dominated coteries, *IEEE Transactions on Parallel and Distributed Systems*, 6 (1995) 905-914.

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

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

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

[9] 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.

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

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

[12] K. Makino and T. Kameda, Transformation of Regular Non-Dominated Coteries, DIMACS Technical Reports 99-41, 1999.

[13] K. Makino and T. Kameda, Efficient generation of all regular non-dominated coteries, *Proc. Nineteenth ACM Symposium on Principles of Distributed Computing* (*PODC 2000*), Portland, July 16-19, pp. 279-288, 2000.

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

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

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

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

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

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