An algorithm for producing median formulas for Boolean functions

Miguel Couceiro*, Erkko Lehtonen*, Jean-Luc Marichal* and Tamás Waldhauser*†
*Faculty of Science, Technology and Communication, University of Luxembourg
6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg
Email: {miguel.couceiro, erkko.lehtonen, jean-luc.marichal, tamas.waldhauser}@uni.lu
†Bolyai Institute, University of Szeged
Aradi vértanúk tere 1, H-6720 Szeged, Hungary
Email: twaldha@math.u-szeged.hu

***Abstract*—We review various normal form representations of Boolean functions and outline a comparative study between them, which shows that the median normal form system provides representations that are more efficient than the classical DNF, CNF and Reed–Muller (polynomial) normal form representations. We present an algorithm for producing median normal form representations of Boolean functions.**

I. Preliminaries

Let A be an arbitrary nonempty set. An *operation* on A is a map $f\colon A^n \to A$, for some positive integer n, called the *arity* of f. Operations on $\mathbb{B} := \{0,1\}$ are called *Boolean functions.* The set of all n-ary operations is denoted by $\mathcal{O}_A^{(n)} := A^{A^n}$, and the set of all operations on A is denoted by $\mathcal{O}_A := \bigcup_{n\geq 1} \mathcal{O}_A^{(n)}$. For $1 \leq i \leq n$, the operation $(a_1,\dots,a_n) \mapsto a_i$ is called the i-th n-ary *projection* on A, and it is denoted by $x_i^{(n)}$, or when the arity is clear from the context, by x_i. We denote the set of all projections on A by $\mathcal{J}_A$. For a subset $\mathcal{C} \subseteq \mathcal{O}_A$, we set $\mathcal{C}^{(n)} := \mathcal{C} \cap \mathcal{O}_A^{(n)}$.

If $f \in \mathcal{O}_A^{(n)}$ and $g_1,\dots,g_n \in \mathcal{O}_A^{(m)}$, then the *composition* of f with $g_1,\dots,g_n$ is the operation $f(g_1,\dots,g_n) \in \mathcal{O}_A^{(m)}$ given by the rule

$$f(g_1,\dots,g_n)(\mathbf{a}) := f\big(g_1(\mathbf{a}),\dots,g_n(\mathbf{a})\big)$$

for all $\mathbf{a} \in A^m$.

Let $\mathcal{C},\mathcal{D} \subseteq \mathcal{O}_A$. The *composition* of $\mathcal{C}$ with $\mathcal{D}$, denoted by $\mathcal{C}\circ\mathcal{D}$, or simply by $\mathcal{C}\mathcal{D}$, is defined as

$$\mathcal{C}\circ\mathcal{D} := \{f(g_1,\dots,g_n) : f \in \mathcal{C}^{(n)},\ g_1,\dots,g_n \in \mathcal{D}^{(m)},\ n,m \geq 1\}.$$

We say that f is a *simple minor* of g if $f \in \{g\}\mathcal{J}_A$. The simple minor relation is a quasi-order (i.e., a reflexive and transitive relation) on $\mathcal{O}_A$. We say that f and g are *equivalent,* denoted $f \equiv g$, if each is a simple minor of the other. This is commonly described as follows: f and g are equivalent if there exists an operation that can be obtained from both f and g by repeated addition of inessential variables and permutation of variables. For $f_1,\dots,f_n \in \mathcal{O}_A$, we denote $[f_1,\dots,f_n] = \{f \in \mathcal{O}_A : f \equiv f_i \text{ for some } i = 1,\dots,n\}$.

A *clone* on A is a set $\mathcal{C} \subseteq \mathcal{O}_A$ that contains all projections on A and satisfies $\mathcal{C}\mathcal{C} \subseteq \mathcal{C}$ (or equivalently, $\mathcal{C}\mathcal{C} = \mathcal{C}$). In the case when A is finite, the set of all clones on A forms an algebraic lattice, where the lattice operations are the following: meet is the intersection, join is the smallest clone that contains the union. The greatest element is the clone $\mathcal{O}_A$ of all operations on A; the least element is the clone $\mathcal{J}_A$ of all projections on A. For sets A of cardinality at least 3, this lattice is uncountable, and its structure remains a topic of current research; see, e.g., [5], [8].

In the case when $|A| = 2$, the lattice of clones on A is countably infinite, and it was completely described by E. Post (see [12], or [14], [16], [18] for shorter recent proofs). The clones of Boolean functions and the lattice of clones on $\mathbb{B}$ are often called *Post classes* and the *Post lattice,* respectively. See the Appendix for a list of Post classes.

The set $\mathbb{B}^n$ is a Boolean (distributive and complemented) lattice of 2^n elements under the componentwise ordering of n-tuples. We will write $\mathbf{a} \leq \mathbf{b}$ to denote comparison in this lattice. The *complement* of $\mathbf{a} = (a_1,\dots,a_n)$ is defined as $\overline{\mathbf{a}} := (1-a_1,\dots,1-a_n)$. With no danger of ambiguity, we denote by $\mathbf{0}$ and $\mathbf{1}$ the constant tuples $(0,\dots,0)$ and $(1,\dots,1)$ of any length.

The set $\mathbb{B}^{\mathbb{B}^n}$ is a Boolean lattice of 2^{2^n} elements under the pointwise ordering of functions. Both $\mathbb{B}^n$ and $\mathbb{B}^{\mathbb{B}^n}$ are vector spaces over the two-element field $\mathrm{GF}(2) = \mathbb{B}$.

For a Boolean function f, the *dual* of f is defined as $f^{\mathrm{d}}(\mathbf{a}) = \overline{f(\overline{\mathbf{a}})}$ for all $\mathbf{a}$. The *dual* of $\mathcal{C} \subseteq \mathcal{O}_{\mathbb{B}}$ is defined as $\mathcal{C}^{\mathrm{d}} = \{f^{\mathrm{d}} : f \in \mathcal{C}\}$. The dual of a clone is a clone, and it is well known that dualization gives the only nontrivial order-automorphism of the Post lattice.

We also denote by $\mathbf{0}$ and $\mathbf{1}$ the constant functions of any arity having value 0 and 1, respectively, everywhere. We denote the ternary majority function $x_1x_2 + x_1x_3 + x_2x_3$ by μ and the ternary triple sum $x_1 + x_2 + x_3$ by τ, where addition and multiplication are performed in $\mathrm{GF}(2)$.

It is well known that every Boolean function can be represented in disjunctive normal form (DNF) and in conjunctive normal form (CNF). This fact can be restated as

$$\mathcal{O}_{\mathbb{B}} = V_c \circ \Lambda_c \circ I^* = \Lambda_c \circ V_c \circ I^*.$$

It is also known that $M_c = V_c \circ \Lambda_c = \Lambda_c \circ V_c$, so the previous equalities can be written as $\mathcal{O}_{\mathbb{B}} = M_c \circ I^*$.

It is also known that every Boolean function is represented by a unique multilinear polynomial over $\mathrm{GF}(2)$, called the Reed–Muller or Zhegalkin polynomial representation (see [1], [11], [13], [17]). This fact can be restated as $\mathcal{O}_{\mathbb{B}} = L_c \circ \Lambda$. Allowing only constant-preserving linear functions is not really a restriction, because $\mathbf{0}$ can be substituted for a variable if necessary.

These facts led to a study of compositions of clones of Boolean functions [2]. If $\mathcal{C}$ and $\mathcal{D}$ are clones, then either $\mathcal{C} \circ \mathcal{D} = \mathcal{C} \vee \mathcal{D}$ or $\mathcal{C} \circ \mathcal{D}$ is not a clone. It is not always the case that the composition of two clones is a clone. For example, as one can straightforwardly verify, $I_0 \circ I^* = I^* \cup [\mathbf{0}]$, and this is not a clone. These facts were observed in [2], where all pairs of clones of Boolean functions were classified according to whether their composition is a clone. From this classification it followed that every clone can be factorized into "prime" clones, and in particular the clone $\mathcal{O}_{\mathbb{B}}$ can be factorized into minimal clones, i.e., clones that cover the least clone $\mathcal{J}_{\mathbb{B}}$ in the Post lattice. The latter result led to a formalization of the notion of normal form, which subsumes the classical DNF, CNF, Reed–Muller polynomial representations. In addition, we have the so-called median normal form, which was shown to provide, in a certain sense, more efficient representations than the said classical normal form systems. Still in [2], procedures were devised for converting DNF, CNF and Reed–Muller representations into median representations. However, no algorithmic approach to obtain median representations directly was proposed in [2].

Algorithms for producing median representations of Boolean functions have been considered in the literature; see, e.g., [10], [15]. These are based on a set of decomposition rules of the form

$$f = \mu(f_1, f_2, f_3),$$

to be applied recursively, and at each iteration of the algorithm, the applicable rule and the functions f_1, f_2, f_3 are specified.

In this paper, we provide a simple algorithm to construct a median representation of an arbitrary Boolean function directly from its operation table. Our algorithm applies the same decomposition rule at each iteration, namely the median decomposition formula (6). Formula (6) holds only for monotone functions; therefore our algorithm first makes a simple preprocessing step that guarantees that (6) can be applied at each iteration. In Section II, we recall the formalism of [2] and some results concerning the comparison of the various normal form systems, and the algorithm is then presented in Section III.

II. Class composition and normal forms

As mentioned, it was observed in [2] that the clone $\mathcal{O}_{\mathbb{B}}$ can be represented as a composition of minimal clones. Each of the seven minimal clones is generated by a single function (see, e.g., [8]). We refer to the minimum arity of such a generating function as the *arity* of the clone. For each of the minimal clones, there is a unique generating function of minimum arity. The minimal clones, their generating functions of minimum arity, and their arities are summarized in the following table.

clone	generator	arity
SM	μ	3
L_c	τ	3
Λ_c	$\wedge$	2
V_c	$\vee$	2
I^*	$\overline{x_1}$	1
I_0	$\mathbf{0}$	1
I_1	$\mathbf{1}$	1

We impose two simple and natural conditions on the factorization $\mathcal{C} = \mathcal{C}_1 \cdots \mathcal{C}_n$ of a clone $\mathcal{C}$ into minimal clones.

Condition 1. The factors occur in descending order of arity with no repetitions of factors.

Condition 2. For any factorization $\mathcal{C} = \mathcal{D}_1 \cdots \mathcal{D}_m$ of $\mathcal{C}$ into minimal clones satisfying Condition 1, there are no integers i, j, k, l with $0 \le i \le j \le n$, $0 \le k \le l \le m$, such that

$$\mathcal{D}_1 \cdots \mathcal{D}_k \subseteq \mathcal{C}_1 \cdots \mathcal{C}_i,$$
$$\mathcal{D}_{k+1} \cdots \mathcal{D}_l \subset \mathcal{C}_{i+1} \cdots \mathcal{C}_j,$$
$$\mathcal{D}_{l+1} \cdots \mathcal{D}_m \subseteq \mathcal{C}_{j+1} \cdots \mathcal{C}_n.$$

Note that Condition 2 implies in particular that no factor can be dropped off. We say that a factorization satisfying Condition 1 is *redundant,* if it does not satisfy Condition 2.

A *descending irredundant factorization* of the clone $\mathcal{C}$ is a factorization of $\mathcal{C}$ into minimal clones that satisfies Conditions 1 and 2.

Theorem 1 ([2])**.** *The descending irredundant factorizations of $\mathcal{O}_{\mathbb{B}}$ are exactly the following:*

$$SM \circ I^* \circ I_0, \qquad SM \circ I^* \circ I_1,$$
$$L_c \circ \Lambda_c \circ I_0 \circ I_1, \qquad L_c \circ \Lambda_c \circ I_1 \circ I_0,$$
$$L_c \circ V_c \circ I_0 \circ I_1, \qquad L_c \circ V_c \circ I_1 \circ I_0,$$
$$\Lambda_c \circ V_c \circ I^*, \qquad V_c \circ \Lambda_c \circ I^*.$$

Replacing the sequence of unary clones by their composition in the eight descending irredundant factorizations of Theorem 1, we get the following five factorizations of $\mathcal{O}_{\mathbb{B}}$:

$$\mathcal{O}_{\mathbb{B}} = V_c \circ \Lambda_c \circ I^* \quad (1)$$
$$\mathcal{O}_{\mathbb{B}} = \Lambda_c \circ V_c \circ I^* \quad (2)$$
$$\mathcal{O}_{\mathbb{B}} = L_c \circ \Lambda_c \circ I \quad (3)$$
$$\mathcal{O}_{\mathbb{B}} = L_c \circ V_c \circ I \quad (4)$$
$$\mathcal{O}_{\mathbb{B}} = SM \circ \Omega(1) \quad (5)$$

Factorizations (1)–(5) express the fact that every Boolean function has certain normal form representations, namely (1), (2), (3) correspond to the classical DNF, CNF and Reed-Muller polynomial representations, respectively, whereas (4) and (5) correspond to other, less known normal form representations.

The factorization $\mathcal{O}_{\mathbb{B}} = L_c \circ V_c \circ I$ relates to $\mathcal{O}_{\mathbb{B}} = L_c \circ \Lambda_c \circ I$ in the same way as CNF relates to DNF. Essentially

the same as Factorization (4) is expressed by $\mathcal{O}_{\mathbb{B}} = L \circ V_c = L_0 \circ V_1$. These latter factorizations express the fact that every function can be represented as a sum of terms, where each term is a disjunction or $\mathbf{1}$. It is not difficult to prove that this representation is unique up to permutation and repetition of terms and permutation and repetition of variables within terms.

The factorization $\mathcal{O}_{\mathbb{B}} = SM \circ \Omega(1)$ expresses the fact that every function can be expressed by repeated applications of the ternary majority function μ to variables, negated variables, and constants.

We need to introduce some formalism to compare the efficiency of the normal form representations arising from these factorizations.

By a *normal form system* we mean a pair consisting of a sequence of clones $\mathcal{C}_i$ and a sequence of functions γ_i

$$((\mathcal{C}_1, \ldots, \mathcal{C}_k), (\gamma_1, \ldots, \gamma_{k-1}))$$

satisfying the following conditions:

- $\mathcal{O}_{\mathbb{B}} = \mathcal{C}_1 \cdots \mathcal{C}_{k-1}\mathcal{C}_k$,
- $\mathcal{C}_k$ contains only variables, negated variables, or constant functions,
- $\mathcal{C}_i$ is generated by $\gamma_i \notin \mathcal{C}_k$ $(1 \leq i \leq k-1)$,
- $\gamma_i \neq \gamma_j$ for $i \neq j$.

Observe that in this case, $((\mathcal{C}_1^{\mathrm{d}}, \ldots, \mathcal{C}_k^{\mathrm{d}}), (\gamma_1^{\mathrm{d}}, \ldots, \gamma_{k-1}^{\mathrm{d}}))$ also constitutes a normal form system called the *dual system* and the factorization $\mathcal{O}_{\mathbb{B}} = \mathcal{C}_1^{\mathrm{d}} \cdots \mathcal{C}_k^{\mathrm{d}}$ is called the *dual factorization*. Note that (1) and (3) are dual to (2) and (4), respectively, and that (5) is self-dual.

We define an *n-ary formula* of a normal form system $((\mathcal{C}_1, \ldots, \mathcal{C}_k), (\gamma_1, \ldots, \gamma_{k-1}))$ as a string over $\mathcal{C}_k^{(n)} \cup \{\gamma_1, \ldots, \gamma_{k-1}\}$ by the following recursion:

1) The elements of $\mathcal{C}_k^{(n)}$ are n-ary formulas.
2) If γ_i is m-ary and $a_1, \ldots, a_m$ are n-ary formulas and none of the a_i's starts with γ_j with $i > j$, then $\gamma_i a_1 \cdots a_m$ is an n-ary formula.

A *formula* of a normal form system is then an n-ary formula for some n, and the length of a formula Φ as a string of symbols is denoted by $|\Phi|$. Clearly every n-ary formula represents an n-ary function, and every n-ary function is represented by an n-ary formula.

Example 1. Consider the n-ary function $\overline{x_i}$ and a factorization $\mathcal{O}_{\mathbb{B}} = \mathcal{C}_1 \cdots \mathcal{C}_k$. If $\mathcal{C}_k$ contains negated variables, as in Factorizations (1), (2), (5), then the function $\overline{x_i}$ can be represented by the formula $\overline{x_i}$ of length 1. On the other hand, if one of the clones $\mathcal{C}_1, \ldots, \mathcal{C}_{k-1}$ is generated by a negated variable, i.e., it is the clone I^*, then the function $\overline{x_i}$ can be represented by the formula $\neg x_i$ of length 2, where $\neg$ denotes the unary function $0 \mapsto 1$, $1 \mapsto 0$. Finally, $\overline{x_i}$ can be represented by the formula $\tau\mathbf{01}x_i$ of length 4 corresponding to Factorizations (3) and (4) (τ generates L_c and $\mathbf{0}, \mathbf{1}, x_i \in I$).

Factorizations (1)–(5) together with the generators $\vee$, $\wedge$, τ, μ for the clones V_c, Λ_c, L_c, SM will be called *disjunctive, conjunctive, polynomial, dual polynomial,* and *median normal form systems,* denoted $\mathbf{D}$, $\mathbf{C}$, $\mathbf{P}$, $\mathbf{P}^{\mathrm{d}}$, $\mathbf{M}$, respectively, and the corresponding formulas will be called *disjunctive, conjunctive, polynomial, dual polynomial,* and *median formulas.*

Let A be a normal form system, and denote by F_A the set of formulas of A. For a function $f \in \mathcal{O}_{\mathbb{B}}$, we define the *$A$-complexity* of f, denoted $C_A(f)$, as

$$\min\{|\Phi| : \Phi \in F_A,\ \Phi \text{ represents } f\}.$$

Example 2. Let us determine the A-complexity of the ternary majority (median) function μ for the normal form systems defined above. It is well known that

$$\begin{aligned}\mu(x_1, x_2, x_3) &= (x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee (x_2 \wedge x_3)\\ &= (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_3)\\ &= x_1x_2 + x_1x_3 + x_2x_3\\ &= (x_1 \vee x_2) + (x_1 \vee x_3) + (x_2 \vee x_3) + 1.\end{aligned}$$

From these facts it is easy derive the corresponding formulas representing μ in each of the normal form systems.

$$\begin{aligned}\mathbf{M}&\colon \mu x_1x_2x_3,\\ \mathbf{D}&\colon \vee\vee\wedge x_1x_2\wedge x_1x_3\wedge x_2x_3,\\ \mathbf{C}&\colon \wedge\wedge\vee x_1x_2\vee x_1x_3\vee x_2x_3,\\ \mathbf{P}&\colon \tau\wedge x_1x_2\wedge x_1x_3\wedge x_2x_3,\\ \mathbf{P}^{\mathrm{d}}&\colon \tau\tau\vee x_1x_2\vee x_1x_3\vee x_2x_3\mathbf{01}.\end{aligned}$$

It is not difficult to see that these formulas are the shortest possible representations of μ in each of the normal form systems. Therefore,

$$\begin{aligned}C_{\mathbf{M}}(\mu) &= 4, \qquad C_{\mathbf{D}}(\mu) = 11, \qquad C_{\mathbf{C}}(\mu) = 11,\\ C_{\mathbf{P}}(\mu) &= 10, \qquad C_{\mathbf{P}^{\mathrm{d}}}(\mu) = 13.\end{aligned}$$

For normal form systems A and B, we say that A is *polynomially as efficient as* B, denoted $A \preceq B$, if there is a polynomial p with integer coefficients such that $C_A(f) \leq p(C_B(f))$ for all $f \in \mathcal{O}_{\mathbb{B}}$. The relation $\preceq$ is a quasi-order on any set of normal form systems. If neither $A \preceq B$ nor $B \preceq A$ holds, we say that A and B are *incomparable* or, to be more descriptive, that A and B *provide representations of incomparable complexity.* In the case of $A \preceq B$ but $B \not\preceq A$, we say that A is *polynomially more efficient than* B, or that A *provides a representation of lower complexity than* B.

Example 3. We illustrate the fact that $\mathbf{C}$, $\mathbf{D}$, $\mathbf{P}$, $\mathbf{P}^{\mathrm{d}}$ are pairwise incomparable. For detailed explanations, we refer the reader to [2].

Let n be an even positive integer, and let $f_n \colon \mathbb{B}^n \to \mathbb{B}$ be given by

$$(x_1 \vee x_2) \wedge \cdots \wedge (x_{2i-1} \vee x_{2i}) \wedge \cdots \wedge (x_{n-1} \vee x_n).$$

Then $C_{\mathbf{C}}(f_n) \leq 2n$ and $C_{\mathbf{D}}(f_n) \geq (n/2)2^{n/2}$. Thus, $\mathbf{D} \not\preceq \mathbf{C}$. Moreover, by considering the dual functions f_n^{d} (n even) and the dual systems, we conclude that $\mathbf{C} \not\preceq \mathbf{D}$.

To see that $\mathbf{D} \not\preceq \mathbf{P}, \mathbf{P}^{\mathrm{d}}$ and $\mathbf{C} \not\preceq \mathbf{P}, \mathbf{P}^{\mathrm{d}}$, consider, for each odd $n \geq 1$, the n-ary function

$$f_n = x_1 + \cdots + x_n.$$

Then $C_{\mathbf{P}}(f_n), C_{\mathbf{P}^{\mathrm{d}}}(f_n) \leq 2n$ but $C_{\mathbf{D}}(f_n), C_{\mathbf{C}}(f_n) \geq n2^{n-1}$.

To show that $\mathbf{P} \not\preceq \mathbf{D}, \mathbf{C}, \mathbf{P}^{\mathrm{d}}$, consider for each $n \geq 2$ the n-ary function

$$f_n = x_1 \vee \cdots \vee x_n.$$

Then $C_{\mathbf{D}}(f_n), C_{\mathbf{C}}(f_n), C_{\mathbf{P}^{\mathrm{d}}}(f_n) \leq 2n$ but $C_{\mathbf{P}}(f_n) \geq 2^n - 1$. Dually, it can be shown that $\mathbf{P}^{\mathrm{d}} \not\preceq \mathbf{D}, \mathbf{C}, \mathbf{P}$.

Example 4. Let us now illustrate the fact that $\mathbf{D}, \mathbf{C}, \mathbf{P}, \mathbf{P}^{\mathrm{d}} \not\preceq \mathbf{M}$. For each $k \geq 1$, let $n = 2k + 1$, and consider the n-ary self-dual monotone function f_k defined inductively as follows

$$f_1 = \mu(x_1, x_2, x_3)$$
$$f_k = \mu(f_{k-1}, x_{2k}, x_{2k+1}) \quad (k \geq 2).$$

Then $C_{\mathbf{M}}(f_k) \leq 3k + 1$, but $C_{\mathbf{D}}(f_k), C_{\mathbf{C}}(f_k), C_{\mathbf{P}}(f_k)$, $C_{\mathbf{P}^{\mathrm{d}}}(f_k) \geq 2^{k+1} - 1$.

Example 5. As shown in [2], we also have $\mathbf{M} \preceq \mathbf{D}, \mathbf{C}, \mathbf{P}, \mathbf{P}^{\mathrm{d}}$. To see that $\mathbf{M} \preceq \mathbf{D}$, just observe that

$$a \wedge b = \mu(0, a, b),$$
$$a \vee b = \mu(1, a, b).$$

Using these identities, one can easily convert every disjunctive formula Φ into a median formula Ψ representing the same function such that $|\Psi| \leq 2|\Phi|$. Thus, for every $f \in \mathcal{O}_{\mathbb{B}}$, we have that $C_{\mathbf{M}}(f) \leq 2C_{\mathbf{D}}(f)$; hence $\mathbf{M} \preceq \mathbf{D}$. Dually, $\mathbf{M} \preceq \mathbf{C}$.

In [2], a method was presented for converting a polynomial formula Φ into a median formula Ψ representing the same function such that $|\Psi| \leq 180|\Phi|^2$. Thus, for every $f \in \mathcal{O}_{\mathbb{B}}$, we have that $C_{\mathbf{M}}(f) \leq 180(C_{\mathbf{P}}(f))^2$; hence we also have $\mathbf{M} \preceq \mathbf{P}$. Dually, $\mathbf{M} \preceq \mathbf{P}^{\mathrm{d}}$.

Examples 3–5 illustrate the key points of the proof of the following theorem.

Theorem 2 ([2])**.** *The disjunctive, conjunctive, polynomial, and dual polynomial normal form systems provide representations of pairwise incomparable complexities. The median normal form system* $\mathbf{M}$ *provides representations of lower complexity than the other four normal form systems* $\mathbf{D}$*,* $\mathbf{C}$*,* $\mathbf{P}$*,* $\mathbf{P}^{\mathrm{d}}$*.*

III. The algorithm

The part of the proof of Theorem 2 in [2] that showed that $\mathbf{M} \preceq \mathbf{D}, \mathbf{C}, \mathbf{P}, \mathbf{P}^{\mathrm{d}}$ provided algorithms for translating DNF, CNF, Reed-Muller polynomial representations into median formulas. However, no hints were given how to construct median formulas directly from the operation table of a Boolean function. In the remainder of this paper, we will describe such an algorithm for producing median formulas directly.

A. The case of monotone functions

Let $(L; \wedge, \vee)$ be a bounded distributive lattice, and denote the least and the greatest elements of L by 0 and 1, respectively. The *median function* on L is the ternary operation $\mathrm{med}\colon L^3 \to L$ given by the rule

$$\begin{aligned}\mathrm{med}(x_1, x_2, x_3) &:= (x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee (x_2 \wedge x_3) \\ &= (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_3),\end{aligned}$$

for all $x_1, x_2, x_3 \in L$. Note that when $L = \mathbb{B}$, $\mathrm{med} = \mu$.

In [3], [4], [9], it was shown that lattice polynomial functions on a bounded distributive lattice $(L; \wedge, \vee)$ (i.e., members of the clone on L generated by $\wedge$, $\vee$ and all constants $c \in L$) can be axiomatized by the following property. A function $f\colon L^n \to L$ is said to be *median decomposable* if for every $i \in \{1, \ldots, n\}$, the following equality holds:

$$\begin{aligned}f(x_1, \ldots, x_n) = \\ \mathrm{med}(f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n), x_i, \\ f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)).\end{aligned} \quad (6)$$

Theorem 3 ([3], [4], [9])**.** *Let* $(L; \wedge, \vee)$ *be a bounded distributive lattice. A function* $f\colon L^n \to L$ *is a lattice polynomial function if and only if it is median decomposable.*

In the case $L = \mathbb{B}$, Tohma [15] provided a similar and somewhat more general decomposition with respect to a given variable, under the assumption that the function is monotone in that variable; see Theorem 1 in [15]. For monotone Boolean functions, (6) is a particular instance of Tohma's decomposition scheme. However, Tohma's decomposition scheme does not generalize as such to arbitrary bounded distributive lattices.

It is well known that every monotone Boolean function is a lattice polynomial function on $(\mathbb{B}; \wedge, \vee)$. Thus Theorem 3 gives rise to Algorithm 1 to construct median normal forms of monotone functions.

Algorithm 1 MMNF – Median normal form for monotone Boolean functions

Input: a monotone Boolean function $f\colon \mathbb{B}^n \to \mathbb{B}$
Output: a median normal form representation of f

1: **if** $n \geq 2$ **then**
2: $\quad \alpha \leftarrow \mathrm{MMNF}(f(x_1, \ldots, x_{n-1}, 0))$
3: $\quad \beta \leftarrow \mathrm{MMNF}(f(x_1, \ldots, x_{n-1}, 1))$
4: $\quad$ **return** $\mu\alpha x_n\beta$
5: **else if** $f = \mathbf{0}$ **then**
6: $\quad$ **return** 0
7: **else if** $f = \mathbf{1}$ **then**
8: $\quad$ **return** 1
9: **else**
10: $\quad$ **return** x_1
11: **end if**

Remark 1. In Algorithm 1, on lines 2–3, $f(x_1, \ldots, x_{n-1}, c)$ ($c \in \mathbb{B}$) denotes the $(n-1)$-ary function obtained from f by substituting the constant c for x_n. The set of monotone Boolean functions is a clone; hence $f(x_1, \ldots, x_{n-1}, c)$ is monotone whenever f is monotone. Therefore the recursive calls on lines 2–3 are legitimate.

Remark 2. In fact, Algorithm 1, with obvious changes, can be used to produce a median representation for any polynomial function over an arbitrary bounded distributive lattice L.

B. The general case

To deal with the general case, when $f\colon \mathbb{B}^n \to \mathbb{B}$ is not necessarily monotone, we construct a monotone function

$g_f\colon \mathbb{B}^{2n} \to \mathbb{B}$ in such a way that f can be recovered from g_f by substituting negated variables for some variables. In this way, we can apply Algorithm 1 to obtain a median normal form representation of g_f, in which we can then perform the said substitutions of negated variables in order to obtain a median normal form representation of f.

Let $f\colon \mathbb{B}^n \to \mathbb{B}$ be an arbitrary Boolean function. Define $g_f\colon \mathbb{B}^{2n} \to \mathbb{B}$ as follows: for all $\mathbf{a} := (a_1, \ldots, a_{2n}) \in \mathbb{B}^{2n}$, let $\mathbf{b} := (a_1, \ldots, a_n)$, $\mathbf{c} := (a_{n+1}, \ldots, a_{2n})$, and let

$$g_f(\mathbf{a}) := \begin{cases} 0 & \text{if } w(\mathbf{a}) < n, \\ 1 & \text{if } w(\mathbf{a}) > n, \\ f(\mathbf{b}) & \text{if } \mathbf{b} = \overline{\mathbf{c}}, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Here, $w(\mathbf{a})$ denotes the *Hamming weight* of $\mathbf{a}$, i.e., the number of 1's in $\mathbf{a}$. It is easy to verify that g_f is monotone. Moreover, for all $\mathbf{b} \in \mathbb{B}^n$, $f(\mathbf{b}) = g_f(\mathbf{b}, \overline{\mathbf{b}})$; hence

$$f = g_f(x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}).$$

Remark 3. Observe that we could have defined g_f in any way in the "otherwise" case, i.e., on the tuples $\mathbf{a}$ with $w(\mathbf{a}) = n$ and $\mathbf{b} \neq \overline{\mathbf{c}}$, and it would still have our desired properties.

This construction leads to Algorithm 2.

Algorithm 2 GENMNF – Median normal form for Boolean functions

Input: a Boolean function $f\colon \mathbb{B}^n \to \mathbb{B}$
Output: a median normal form representation of f

```
if f is monotone then
    return MMNF(f)
else
    Construct g_f as in (7).
    w ← MMNF(g_f)
    for i = 1 to n do
        Replace each occurrence of x_{n+i} in w by x̄_i.
    end for
    return w
end if
```

Example 6. In order to illustrate the use of our algorithm, let us produce a median formula for the function $f\colon \mathbb{B}^2 \to \mathbb{B}$ defined by the following operation table.

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

Since f is not monotone, we first need to construct $g_f\colon \mathbb{B}^4 \to \mathbb{B}$; it is the 4-ary function whose true points are the following: 0110, 1001, 0111, 1011, 1101, 1110, 1111.

Then we apply Algorithm 1 to g_f. Figure 1 presents the parsing tree of the median formula constructed by the algorithm. Each internal node and some leaves carry an extra label, namely, g_f or a function obtained from g_f by substituting constants for variables as produced by the algorithm when making a recursive call. Finally, we substitute $\overline{x_1}$ for x_3 and $\overline{x_2}$ for x_4, and we obtain the following median formula for f:

$$\mu\mu\mu 0 x_2 0 \overline{x_1} \mu 0 x_2 1 \overline{x_2} \mu\mu x_1 x_2 x_1 \overline{x_1} \mu x_1 x_2 1. \tag{8}$$

IV. Concluding remarks and future work

In Section II, we saw that the median normal form system provides representations of lower complexity than the classical disjunctive, conjunctive, polynomial and dual polynomial systems. However, the algorithms we presented here may not produce median formulas of the lowest possible complexity. This fact asks for ways of simplifying median formulas, in analogy with known resolution procedures for DNF and CNF. For instance, using the identities

$$\mu(x, x, y) = x, \quad \mu(x, \overline{x}, y) = y,$$

(8) simplifies to

$$\mu\mu 0 \overline{x_1} x_2 \overline{x_2} \mu x_1 x_2 1.$$

In fact, (6) constitutes a simplification rule in itself. For example, Algorithm 1 applied to each function f_k of Example 4 produces an unoptimal representation, which becomes optimal under the simplification rule given by (6). The development of these resolution procedures constitutes a topic of ongoing research.

Acknowledgment

The fourth author acknowledges that the present project is supported by the National Research Fund, Luxembourg, and cofunded under the Marie Curie Actions of the European Commission (FP7-COFUND), and supported by the Hungarian National Foundation for Scientific Research under grant no. K77409.

The authors would like to thank the anonymous reviewers for their constructive remarks which helped improve this manuscript.

Appendix – Post classes

We make use of notations and terminology appearing in [6] and [7].

- $\mathcal{O}_{\mathbb{B}}$: the clone of all Boolean functions;
- T_0: the clone of 0-preserving functions, i.e.,
$$T_0 = \{f \in \mathcal{O}_{\mathbb{B}} : f(0, \ldots, 0) = 0\};$$
- T_1: the clone of 1-preserving functions, i.e.,
$$T_1 = \{f \in \mathcal{O}_{\mathbb{B}} : f(1, \ldots, 1) = 1\};$$
- $T_c = T_0 \cap T_1$;
- M: the clone of monotone functions, i.e.,
$$M = \{f \in \mathcal{O}_{\mathbb{B}} : f(\mathbf{a}) \leq f(\mathbf{b}) \text{ whenever } \mathbf{a} \leq \mathbf{b}\};$$
- $M_0 = M \cap T_0$, $M_1 = M \cap T_1$, $M_c = M \cap T_c$;
- S: the clone of self-dual functions, i.e.,
$$S = \{f \in \mathcal{O}_{\mathbb{B}} : f^{\mathrm{d}} = f\};$$

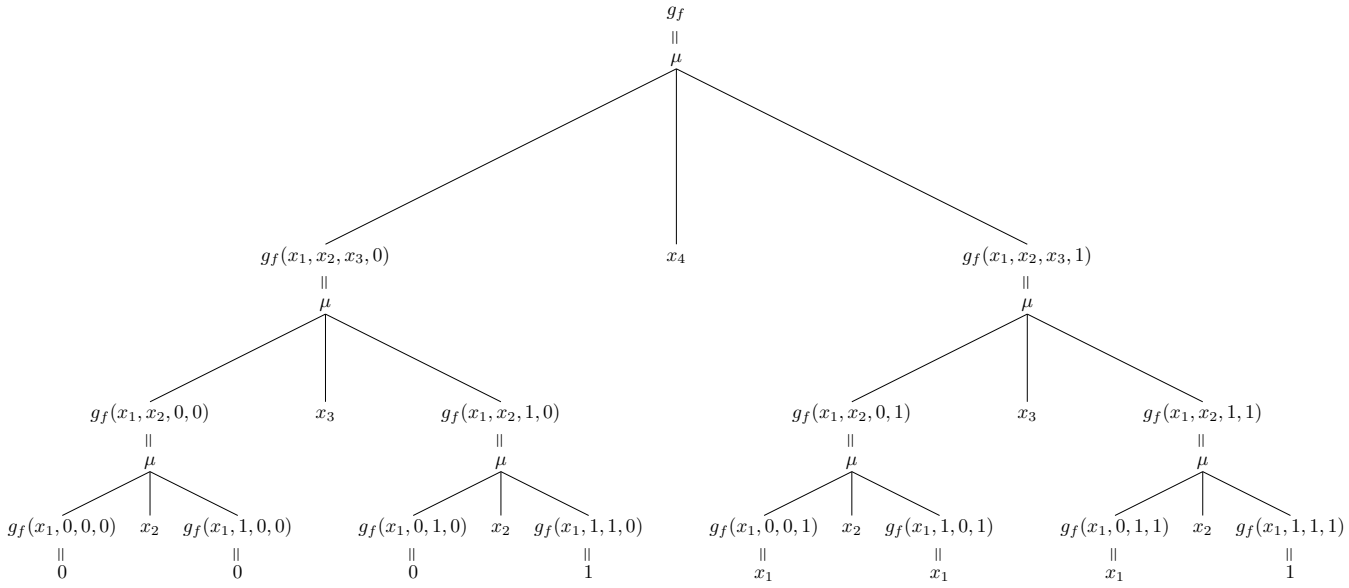

Fig. 1. The parsing tree of a median formula of g_f, for the f given in Example 6.

- $S_c = S \cap T_c$, $SM = S \cap M$;
- L: the clone of all linear functions, i.e.,

$$L = \{f \in \mathcal{O}_{\mathbb{B}} : f = c_0 + c_1x_1 + \cdots + c_nx_n \text{ for some } n \geq 0 \text{ and } c_0, \ldots, c_n \in \mathbb{B}\};$$

- $L_0 = L \cap T_0$, $L_1 = L \cap T_1$, $LS = L \cap S$, $L_c = L \cap T_c$.

Let $a \in \mathbb{B}$. A set $A \subseteq \mathbb{B}^n$ is said to be *a-separating* if there is i, $1 \leq i \leq n$, such that for every $(a_1, \ldots, a_n) \in A$ we have $a_i = a$. A function f is said to be *a-separating* if $f^{-1}(a)$ is a-separating. The function f is said to be *a-separating of rank $k \geq 2$* if every subset $A \subseteq f^{-1}(a)$ of size at most k is a-separating.

- U_m: the clone of 1-separating functions of rank $m \geq 2$;
- $U_\infty = \bigcap_{k\geq 2} U_k$;
- $T_cU_m = T_c \cap U_m$, $MU_m = M \cap U_m$, $M_cU_m = M_c \cap U_m$ for $m = 2, \ldots, \infty$;
- W_m: the clone of 0-separating functions of rank $m \geq 2$;
- $W_\infty = \bigcap_{k\geq 2} W_k$;
- $T_cW_m = T_c \cap W_m$, $MW_m = M \cap W_m$, $M_cW_m = M_c \cap W_m$ for $m = 2, \ldots, \infty$;
- Λ: the clone of all conjunctions and constants, i.e.,

$$\Lambda = [x_1 \wedge \cdots \wedge x_n : n \geq 1] \cup [\mathbf{0}] \cup [\mathbf{1}];$$

- $\Lambda_0 = \Lambda \cap T_0$, $\Lambda_1 = \Lambda \cap T_1$, $\Lambda_c = \Lambda \cap T_c$;
- V: the clone of all disjunctions and constants, i.e.,

$$V = [x_1 \vee \cdots \vee x_n : n \geq 1] \cup [\mathbf{0}] \cup [\mathbf{1}];$$

- $V_0 = V \cap T_0$, $V_1 = V \cap T_1$, $V_c = V \cap T_c$;
- $\Omega(1)$: the clone of projections, negations, and constants;
- I^*: the clone of projections and negations;
- I: the clone of projections and constants;
- $I_0 = I \cap T_0$, $I_1 = I \cap T_1$;
- I_c: the clone of projections.

REFERENCES

[1] F. M. Brown, *Boolean Reasoning: The Logic of Boolean Equations,* 2nd ed., Dover, Mineola, NY, 2003.

[2] M. Couceiro, S. Foldes and E. Lehtonen, Composition of Post classes and normal forms of Boolean functions, *Discrete Math.* **306** (2006) 3223–3243.

[3] M. Couceiro and J.-L. Marichal, Polynomial functions over bounded distributive lattices, arXiv:0901.4888.

[4] M. Couceiro and J.-L. Marichal, Characterizations of discrete Sugeno integrals as polynomial functions over distributive lattices, *Fuzzy Sets and Systems* **161** (2010) 694–707.

[5] K. Denecke and S. L. Wismath, *Universal Algebra and Applications in Theoretical Computer Science,* Chapman & Hall/CRC, Boca Raton, 2002.

[6] S. Foldes and G. R. Pogosyan, Post classes characterized by functional terms, *Discrete Appl. Math.* **142** (2004) 35–51.

[7] S. W. Jablonski, G. P. Gawrilow and W. B. Kudrjawzew, *Boolesche Funktionen und Postsche Klassen,* Vieweg, Braunschweig, 1970.

[8] D. Lau, *Function Algebras on Finite Sets,* Springer-Verlag, Berlin, Heidelberg, 2006.

[9] J.-L. Marichal, Weighted lattice polynomials, *Discrete Math.* **309** (2009) 814–820.

[10] F. Miyata, Realization of arbitrary logical functions using majority elements, *IEEE Trans. on Electronic Computers* **EC-12** (1963) 183–191.

[11] D. E. Muller, Application of Boolean algebra to switching circuit design and to error correction, *IRE Trans. Electron. Comput.* **3**(3) (1954) 6–12.

[12] E. L. Post, *The Two-Valued Iterative Systems of Mathematical Logic,* Annals of Mathematical Studies, vol. 5, Princeton University Press, Princeton, 1941.

[13] I. S. Reed, A class of multiple-error-correcting codes and the decoding scheme, *IRE Trans. Inf. Theory* **4**(4) (1954) 38–49.

[14] M. Reschke and K. Denecke, Ein neuer Beweis für die Ergebnisse von E. L. Post über abgeschlossene Klassen Boolescher Funktionen, *Elektronische Informationsverarbeitung und Kybernetik* **25**(7) (1989) 361–380.

[15] Y. Tohma, Decompositions of logical functions using majority decision elements, *IEEE Trans. on Electronic Computers* **EC-13** (1964) 698–705.

[16] A. B. Ugol'nikov, On closed Post classes, *Izv. Vyssh. Uchebn. Zaved. Mat.* **7**(314) (1988) 79–88 (in Russian); translated in *Sov. Math.* **32**(7) (1988) 131–142.

[17] I. I. Zhegalkin, On the calculation of propositions in symbolic logic, *Mat. Sb.* **34** (1927) 9–28 (in Russian).

[18] I. E. Zverovich, Characterizations of closed classes of Boolean functions in terms of forbidden subfunctions and Post classes, *Discrete Appl. Math.* **149** (2005) 200–218.