Median based calculus for lattice polynomials and monotone Boolean functions

Miguel Couceiro, Pierre Mercuriali, Romain Péchoux, Abdallah Saffidine

Median based calculus for lattice polynomials and monotone Boolean functions

Miguel Couceiro, Pierre Mercuriali, Romain Péchoux
LORIA (CNRS - inria Nancy G.E. - Univ. Lorraine) Vandœuvre-les-Nancy, F-54506, France
{miguel.couceiro, pierre.mercuriali, romain.pechoux}@loria.fr

Abdallah Saffidine*
CSE, The University of New South Wales, Sydney, Australia
abdallah.saffidine@gmail.com

*was supported by the Australian Research Council (project DE 150101351).

***Abstract*—In this document, we consider a *median*-based calculus for efficiently representing polynomial functions over distributive lattices. We extend an equational specification of median forms from the domain of Boolean functions to the domain of lattice polynomials. We show that it is *sound* and *complete*, and we illustrate its usefulness when simplifying median formulas algebraically. Furthermore, we propose a definition of *median normal forms* (MNF), that are thought of as minimal median formulas with respect to a structural ordering of expressions. We also investigate related complexity issues and show that the problem of deciding whether a formula is in MNF is in Σ_2^{P}. Moreover, we explore polynomial approximations of solutions to this problem through a sound *term rewriting system* extracted from the proposed equational specification.**

I. Introduction

Representing a function with various operators sometimes yields drastically different results in terms of the size of the formula used to represent it. In the case of Boolean functions, it has been shown that a median-based representation, that is, a representation based on the ternary median operator, yields asymptotically smaller formulas –in terms of number of connectives– than the classical DNF, CNF, and polynomial normal forms [7]. Moreover, algorithmic procedures to obtain such median representations were given in [8]. However, these procedures may not produce median formulas of the lowest possible complexity (size), and consequently procedures for simplifying median formulas are required, in analogy with well known resolution procedures for DNF and CNF expressions.

Since median expressions can be translated into the language of lattices and conversely, lattice polynomials can also be represented by median expressions. Thus, in addition to applications in logic and circuit design, these simplifications become useful when efficiently representing noteworthy classes of aggregation functions such as the Sugeno integral ([13]).

Motivated by these observations and following the work of [7], in this paper we investigate the use of the ternary median connective to efficiently represent lattice polynomials, which in the case of 2-element lattices reduce to monotone Boolean functions. In Section II we recall basic background on lattice functions and median algebras that will be used throughout this paper. In Section III we give a finite equational specification (System 1) that allows the simplification of median formulas while preserving logical equivalence, and we show that this equational specification is both *sound* and *complete* (Theorem 1). As an immediate consequence, it then follows that we may rewrite any formula into any other that is logically equivalent to it, and that median formulas can be simplified algebraically according to this equational specification. We also propose a structural and lexicographic ordering of formulas in Section IV, and introduce *median normal forms* (MNFs) as being median formulas that are minimal with respect to this ordering.

In Section V we consider two decision problems related to the task of finding an MNF of a given formula, and we show that both are at most *moderately intractable*, by which we mean that it is likely to be intractable, but not beyond Σ_2^{P}. We also explore a different approach to the potential intractability issue by providing a *term rewriting system* based on the equational specification (System 1). Even though the resulting system is not complete, it runs in polynomial time, thus highlighting a trade-off in complexity: either we sacrifice completeness but preserve tractability, or we keep completeness at the cost of high complexity.

II. Lattice polynomials and median formulas

A. Notation and Preliminaries

In this subsection we recall definitions and notations on lattice and lattice functions, while adopting the terminology of [9]. A *lattice* is an algebraic structure $\langle L, \wedge, \vee \rangle$ where L is a nonempty set, called *universe*, and where $\wedge$ and $\vee$ are two binary operations that satisfy the laws of commutativity, associativity, absorption, and idempotence; a lattice is said to be *distributive* if the two laws distribute over one another. With no danger of ambiguity, we will denote a lattice $\langle L, \wedge, \vee \rangle$ by its universe L.

In what follows, L will always denote an arbitrary bounded distributive lattice with least and greatest elements $\perp$ and $\top$, respectively. For $a, b \in L$, $a \leq b$ means that $a \wedge b = a$ or, equivalently, $a \vee b = b$. For any integer $n \geq 1$, we set $[n] = \{1, \ldots, n\}$. For an arbitrary nonempty set A and a lattice L, the set L^A of all functions from A to L also constitutes a lattice under the operations

$$(f \wedge g)(x) = f(x) \wedge g(x) \quad \text{and} \quad (f \vee g)(x) = f(x) \vee g(x)$$

for every $f, g \in L^A$. In particular, any lattice L induces a lattice structure on the Cartesian product $L^n, n \geq 1$, by defining $\wedge$ and $\vee$ componentwise, i.e.,

$$\begin{aligned}(a_1, \dots, a_n) \wedge (b_1, \dots, b_n) &= (a_1 \wedge b_1, \dots, a_n \wedge b_n),\\ (a_1, \dots, a_n) \vee (b_1, \dots, b_n) &= (a_1 \vee b_1, \dots, a_n \vee b_n).\end{aligned}$$

We denote the elements of L by lower case letters $a, b, c, \dots$ and the elements of $L^n, n > 1$, by bold face letters $\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$.

We now recall the notion of lattice polynomial function. The class of *lattice polynomial functions* (or simply, *polynomial functions*) from L^n to L is defined inductively as the set of functions represented by expressions constructed in the language of lattices: the projections $\mathbf{x} \mapsto x_i$, the constant functions $\mathbf{x} \mapsto c$, $c \in L$, and if f and g are polynomial functions, then so are $f \wedge g$ and $f \vee g$.

For instance, the ternary *median* m, i.e., the function given by:

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

is an example of such a polynomial function.

B. Equational specification

We make use of the notations of [12] and [3] to introduce equational specifications. An *equational specification*, (sometimes called *equational system* within this document) is a pair (Σ, E) of an *alphabet* or *signature* Σ and a set of equations E. The alphabet Σ consists of a countably infinite set of *variables* $x_1, x_2, \dots$ and a nonempty set of *function symbols* or *operator symbols*. In the current setting, this set contains m and constants.

The set of *terms* (or *expressions*) over Σ is denoted by $\mathrm{Ter}(\Sigma)$ and it is defined inductively as follows:

(1) every variable and constant in Σ is in $\mathrm{Ter}(\Sigma)$, and
(2) if f is an n-ary function symbol and $t_1, \dots, t_n$ are terms, then $f(t_1, \dots, t_n)$ is in $\mathrm{Ter}(\Sigma)$.

An *equation* is then an expression of the form $s = t$ where $s, t \in \mathrm{Ter}(\Sigma)$. The set of all equations is denoted by E.

The set of *median terms* (also referred to as *median expressions* or *median formulas* in [7]) $\mathbf{M}$ will denote the set of all formulas that are constructed using variables, constants and the median m. Even though [7] focuses on median normal expressions for Boolean functions, this notion adapts rather naturally to lattice polynomial functions.

Given a formula ϕ in $\mathbf{M}$, its *depth* is denoted by $d(\phi)$ and defined as follows:

(i) for every variable or constant a, $d(a) = 0$,
(ii) for every formula $\phi = \mathrm{m}(a, b, c) \in \mathbf{M}$,

$$d(\phi) = \max\{d(a), d(b), d(c)\} + 1.$$

The *size* $|\phi|$ of a median term ϕ is the number of medians in it.

Example 1. *Let* $\phi = \mathrm{m}(\mathrm{m}(x, x, y), \mathrm{m}(x, y, z), v)$. *Then* $d(\phi) = 2$ *and* $|\phi| = 3$.

Two median terms ϕ and ψ are said to be *equivalent*, denoted by $\phi \equiv \psi$, if they represent the same function.

Example 2. *For instance, the median terms:*

$$\begin{aligned}\phi_1 &= \mathrm{m}(\mathrm{m}(\mathrm{m}(x, u, v), \mathrm{m}(x, y, v), \mathrm{m}(y, u, v)), x, v)\\ \phi_2 &= \mathrm{m}(\mathrm{m}(u, y, v), x, v)\end{aligned}$$

are equivalent. However, the median terms:

$$\phi_3 = \mathrm{m}(\mathrm{m}(x, y, z), u, v) \quad \textit{and} \quad \phi_4 = \mathrm{m}(x, \mathrm{m}(y, z, u), v)$$

are not equivalent. In other words, m *is not* associative *in the sense of [1], [10], [19].*

A *substitution* is a map σ from $\mathrm{Ter}(\Sigma)$ to $\mathrm{Ter}(\Sigma)$ that satisfies

$$\sigma(F(t_1, \dots, t_n)) = F(\sigma(t_1), \dots, \sigma(t_n))$$

for every n-ary function symbol (here $n \geq 0$).

Substitution together with the other rules recalled in Table I, give rise to the so-called *derivable equations*, i.e., equations obtained by applying a finite combination of these rules. If an equation $s = t$ is derivable from the equations in E, than we write $(\Sigma, E) \vdash s = t$ or $s \vdash_E t$.

$$\begin{array}{ll}
(\Sigma, E) \vdash t = t & \text{if } t \in \mathrm{Ter}(\Sigma)\\
(\Sigma, E) \vdash s = t & \text{if } s = t \in E\\
\dfrac{(\Sigma, E) \vdash s = t}{(\Sigma, E) \vdash t = s} & \\
\dfrac{(\Sigma, E) \vdash t_1 = t_2, (\Sigma, E) \vdash t_2 = t_3}{(\Sigma, E) \vdash t_1 = t_3} & \\
\dfrac{(\Sigma, E) \vdash s = t}{(\Sigma, E) \vdash \sigma(s) = \sigma(t)} & \text{for every substitution } \sigma\\
\dfrac{(\Sigma, E) \vdash s_1 = t_1, \dots, (\Sigma, E) \vdash s_n = t_n}{(\Sigma, E) \vdash F(s_1, \dots, s_n) = F(t_1, \dots, t_n)} & \text{for every } n\text{-ary } F \in \Sigma
\end{array}$$

Table I
EQUATIONAL INFERENCE SYSTEM.

C. Term Rewriting systems

A *Term Rewriting System (TRS)* is an equational specification with all its equations oriented. A pair (l, r) of terms in $\mathrm{Ter}(\Sigma)$, written as $l \longrightarrow r$, is a *reduction rule* if r is not a variable and all the variables in l are already contained in r.

Each term rewriting system yields a rewrite relation defined to be the closure by substitution and context of its reduction rules.

III. Axiomatization of lattice polynomials

In this section, we present an equational system for median calculus which is both sound and complete, and which we then use to manipulate median expressions.

First, recall that lattice polynomial functions $f\colon L^n \to L$ are exactly the solutions of the *median decomposition system* [17]:

$$f(\mathbf{x}) = \mathrm{m}(f(\mathbf{x}_k^{\perp}), x_k, f(\mathbf{x}_k^{\top})) \qquad (1)$$

for all $\mathbf{x} = (x_1, \dots, x_n)$, $k \in [n]$ and $c \in L$, and where

$$\mathbf{x}_k^c := (x_1, \dots, x_{k-1}, c, x_{k+1}, \dots, x_n).$$

A direct consequence of this result is that every polynomial function can be represented by a median term. Indeed, recursive applications of (1) on each $x_1, \dots, x_n$ of an n-ary polynomial function f produces a median formula representing f, see, e.g., [8].

However, this procedure may not produce optimal formulas (size wise), and this fact motivates the current study.

Example 3. *Consider the 5-ary majority operator* $\mathrm{m}_5 : \boldsymbol{x} \mapsto \mathrm{m}_5(x_1, x_2, x_3, x_4, x_5)$ *over Boolean variables. Using the median decomposition algorithm mentioned above we can construct a median formula representation of this function using its values on every point of* $\{\perp, \top\}^5$. *This representation, of size* 31*, is not minimal. There are smaller representations of size* 4*, as shown in [4] and [18].*

Let us now recall an axiomatisation of the algebraic structure $\langle L, \mathrm{m}, \perp, \top \rangle$, that is the set L with the ternary median function m and the 0-ary functions (constants) $\perp$ and $\top$, by the following equational system.

System 1.

$(M1)\, \mathrm{m}(x, y, z) = \mathrm{m}(x, z, y) = \mathrm{m}(z, x, y)$,

$(M2)\, \mathrm{m}(x, x, y) = x$,

$(M3)\, \mathrm{m}(\mathrm{m}(x, u, v), \mathrm{m}(y, u, v), z) = \mathrm{m}(\mathrm{m}(x, y, z), u, v)$,

$(M4)\, \mathrm{m}(\perp, \top, x) = x$,

for all x, y, z, u, v *in* L.

Equations $(M1), (M2), (M3)$ are known as *symmetry*, *majority*, *distributivity* laws, respectively.

Remark 1. *An axiomatisation of the Boolean algebra* $\langle \{0, 1\}, \wedge, \vee, \bar{\cdot}, 0, 1 \rangle$ *was used in [2] to simplify Boolean circuits. System 1 is an adaptation of these results to the case of lattice polynomials.*

In order to manipulate median formulas, we need to know whether the axiomatisation given by System 1 is both *sound* and *complete*. Soundness means that every equation $s = t$ that can be derived from System 1 is valid, i.e., that the formulas s and t are equivalent. Completeness means that every equation $s = t$ that is valid can be derived from the axioms of the system. For our purposes, having a sound and complete system is interesting in order to rewrite median terms, and hopefully simplify them. Soundness indeed ensures that whatever simplification we do by applying an equation to a median term will preserve logical equivalence, while completeness ensures that we can infer a median term from an equivalent one by using System 1.

Theorem 1. *The algebra* $\langle L, \mathrm{m}, \perp, \top \rangle$ *together with the axioms of System 1 is sound and complete.*

Sketch of Proof. Soundness is provable algebraically by deriving the axioms of the system using the properties of the lattice L. Completeness is an immediate consequence of the Birkhoff's Completeness Theorem for equational logic, see, e.g., [6], [12]. □

IV. Median normal forms

In this section, we propose a structural description of median formulas and introduce the notion of *median normal forms (MNF)* that, as we shall see, correspond to median formulas that are "minimal" with respect to the lexicographical ordering of their structural description.

A. Orders

A binary relation $\preceq$ on a set S is a *quasi-order*, or *preorder*, if it is reflexive and transitive. A quasi-order is said to be a *partial order* if it is antisymmetric. If $\preceq$ is a partial order, the structure $\langle S, \preceq \rangle$ is called a *partially ordered set* (or *poset*). A quasi-ordered set is *well-founded* if it satisfies the descending chain condition, i.e., there exists no infinite decreasing sequence $\cdots < t_2 < t_1$ of elements of $\mathcal{T}$. If for every pair (a, b), either $a \leq b$ or $b \leq a$, then $(\mathcal{T}, \leq)$ is said to be *totally ordered.* A well-founded and totally ordered set is said to be a *well-ordered set.*

Let n be a positive integer, and T_n the set of all ordered n-tuples over S. Let $T = \bigcup_{n \leq 1} T_n$. The *lexicographical extension* on T denoted by $\preceq_{lex}$, is defined by: $(x_1, \dots, x_m) \preceq_{lex} (y_1, \dots, y_n)$ if

- $m \leq n$ and for all $k \in [m]$, $\quad x_k = y_k$[1], or
- there is $k \in [\min(m, n)]$ such that for all$j \in [k-1]$, we have $x_j = y_j$ and $x_k \prec y_k$ (i.e., $x_k \preceq y_k, x_k \neq y_k$)

Example 4. *The* lexicographic order *defined on a finite set of words is well-founded. On the other hand, for* $S = \{a, b\}$ *with* $a \preceq b$*, the lexicographical extension on the infinite product* S^* *is not a well-order:*

$$\ldots \preceq aaab \preceq aab \preceq ab \preceq b.$$

We now define a way to partially describe the structure of median terms that induces a well-order on the set of median formulas. We then show that for every polynomial function, there exists a set of minimal representations with respect to this order, which we call *median normal forms (MNF)*. As it is the case for DNF and CNF representations, this representation is unique modulo some properties like commutativity or associativity. However, the general structure of these minimal representations still eludes us.

[1] I.e., $(x_1, \dots, x_m)$ is a *prefix* of $(y_1, \dots, y_n)$.

Definition 1. *Let ϕ be a median term of depth d. Let $n_0, \dots, n_d$ be nonnegative integers, such that for all $i \in \{0\} \cup [d]$, n_i is the number of medians at depth $\leq i$. The* structural representation *of ϕ is the tuple*

$$S_\phi = (n_d, \dots, n_0).$$

Let $\leq_S$ be the ordering of median formulas defined by:

$$\phi_1 \leq_S \phi_2 \quad \textit{if} \quad S_{\phi_1} \leq_{lex} S_{\phi_2}.$$

Remark 2. *Note that S_ϕ is a decreasing sequence and that $n_d = |\phi|$. Also, the order $\leq_S$ prioritizes the size of the formula over its depth. For instance, consider the following equivalent formulas*

$$\begin{aligned} \phi_1 &= \mathrm{m}(x_1, x_2, \mathrm{m}(x_3, x_4, \mathrm{m}(x_5, x_6, x_7))) \\ \phi_2 &= \mathrm{m}(\mathrm{m}(x_1, x_2, x_3), \mathrm{m}(x_1, x_2, x_4), \mathrm{m}(x_5, x_6, x_7)). \end{aligned}$$

Clearly, $|\phi_1| = 3 < 4 = |\phi_2|$ while $d(\phi_1) = 3 > 2 = d(\phi_1)$. Looking at their structural representation, we have

$$S_{\phi_1} = (3, 2, 1) \quad \textit{whereas} \quad S_{\phi_1} = (4, 3),$$

and hence $\phi_1 \leq_S \phi_2$.

We now give a definition of a median normal form as a minimal median representation.

Definition 2. *We say that a median term ϕ is a* median normal form (MNF) *if for every median term $\phi' \equiv \phi$, we have*

$$\phi \leq_S \phi'.$$

Example 5. *The formula $\phi = \mathrm{m}(\mathrm{m}(x, x, y), y, z)$ is not a median normal form since $\phi' = \mathrm{m}(x, y, z)$ is an equivalent formula, and $S_{\phi'} = (1) \leq_{lex} (2, 1) = S_\phi$.*

Remark 3. *As it has been defined, the structural order cannot account for permutations of variables. For instance, the formulas $\mathrm{m}(x, y, z)$ and $\mathrm{m}(x, z, y)$ have the same structural tuple (1), but they are also equivalent, and both are median normal forms. Thus, a formula does not have a single median normal form, but rather a* set *of median normal forms: the formula ϕ from Example 5 has for set of normal forms $\{\mathrm{m}(x, y, z), \mathrm{m}(x, z, y), \mathrm{m}(y, z, x)\}$.*

V. Considerations on complexity

In this section we address the question of finding median normal forms. To this end, we formalize two decision problems that express the tasks of finding median formulas of smaller structural representation. We show that both problems are at most moderately intractable, and we propose a term rewriting system as a tool for approximating solutions to them.

A. Structurally smaller formulas

Although, the definition of the median normal form is expressed simply, a procedure to convert an input formula into an equivalent formula in median normal form is likely intractable. Indeed, we will show that the mere task of checking if a given formula is in MNF seems to be expensive. We formalize this decision problem in Definition 3.

Definition 3. *Consider the decision problem* SMALLMED*:*
Input: a median term ϕ and a decreasing sequence S
Output: succeeds if there exists a formula ψ whose structural representation is strictly smaller than S. Fails if none exists.

Before proving that SMALLMED is intractable, we give a Σ_2^{P} upper-bound on its hardness. Recall that the Σ_2^{P} complexity class is on the second level of the polynomial hierarchy, between NP and PSPACE [20].

Theorem 2. SMALLMED *is in the class Σ_2^{P}.*

Proof. A convenient characterization of Σ_2^{P} is that it contains decision problems such that the accepting instances can be expressed as a set of words $\{x : \exists c_1 \forall c_2 F(x, c_1, c_2)\}$, where c_1 and c_2 are certificates whose lengths are polynomial in $|x|$ and F is computable in polynomial time. Consider Algorithm 1 which solves SMALLMED. The size of the first certificate ψ, is indeed polynomial in the size of the input, $|\phi|$, because its structural representation is bounded by that of ϕ. The size of the second certificate σ is also polynomial in the input. Therefore, Algorithm 1 ensures SMALLMED is in Σ_2^{P}. □

Algorithm 1 Finding a smaller equivalent median form.

```
Input: A formula $\phi$, a decreasing sequence $S$.
Existentially guess a formula $\psi$ such that $S(\psi) <_S S$
    Universally guess an assignment $\sigma$
        Ensure that $\sigma(\phi) = \sigma(\psi)$
        If so, return SUCCESS
If none exist, then FAIL
```

Note that Theorem 2 simply provides an asymptotic upper bound on the complexity of SMALLMED, but a corresponding lower-bound still eludes us.

Definition 3 assumed that the desired formula size was given as part of the input. If instead we assume a constant target size, then it is possible to obtain a better complexity bound.

Definition 4. *For any fixed decreasing sequence S, we define the decision problem* $\mathrm{SMALLMED}_S$ *as:*
Input: a median term ϕ
Output: succeeds if there is a formula ψ whose structural representation is smaller than S. Fails if none exists.

Theorem 3. *For any decreasing sequence S,* $\mathrm{SMALLMED}_S$ *is in the class co-NP.*

Proof. Let s be the first element of the sequence S, n be the number of variables occurring in ϕ, and V_ϕ be the set of variables occurring in ϕ. Any formula ψ of structural representation smaller than S, has no more than s medians. Hence, such a formula ψ cannot involve more than $N = 2s+1$ variables. If ϕ is equivalent to a formula of structural representation smaller than S, then at most N variables among the ones that occur in ϕ are relevant.

For every subset of variables $V \subset V_\phi$ of size at most N, there is a constant number of formulas smaller than S with variables drawn from V. For each such formula ψ, universally guess a variable assignment σ for the variables occurring in ϕ and check if ϕ and ψ agree on σ. If so, ψ is equivalent to ϕ and we can succeed. If no formula triggers a success, we can end the algorithm and fail.

The number of ways to specify the set V is bounded by n^N. So the total number of universal guesses is bounded by $O(n^N)$. Since N is constant, we conclude that we can determine if ϕ admits an equivalent formula of size smaller than S with a polynomial number of universal guesses. Therefore, SMALLMED$_S$ is in co-NP. □

B. Determining the MNF of a formula

Algorithm 1 does not find an MNF for the input formula directly, but it can be used as a subroutine to an algorithm that does. Let ϕ be an input formula and let S_ϕ be its structural representation, then a kind of binary search allows to identify the smallest structural representation such that Algorithm 1 succeeds. The output of the final call to Algorithm 1 is then an MNF of ϕ.

A binary search performs a number of comparison calls logarithmic in the size of the ordered domain. In our case, the domain is the set of sequences lexicographically smaller than S_ϕ, and its cardinality is at most exponential in the size of ϕ. Therefore, the binary search performs a number of calls to Algorithm 1 that is polynomial in the input formula.

C. A term rewriting approach

A naive implementation of Algorithm 1 amounts to an exhaustive search for equivalent formulas among structurally smaller ones. A possible alternative, would be to search for a structurally smaller formula among equivalent ones.

Recall from Theorem 1 that System 1 is sound and complete. In other words, for any two formulas ϕ and ψ, the formulas are equivalent if and only if there exists a rewriting of ϕ into ψ by means of a sequence of equations from System 1. This idea is implemented in Algorithm 2.

Algorithm 2 Bringing a formula closer to a median normal form.

```
Input: A formula $\phi$, a decreasing sequence $S$.
Output: A formula $\psi$ with structural representation $S_\psi < S$,
    diverges if none exists.
Set $\psi \leftarrow \phi$
While $\psi \geq_S \phi$ do
    Existentially guess a rewriting from System 1: $\psi = \psi'$
        Update $\psi \leftarrow \psi'$
return $\psi$
```

This non-deterministic algorithm uses very little space, namely its space complexity is linear in the size of the input, and solves SMALLMED. It therefore constitutes a proof that SMALLMED is in the class NPSPACE. From Savitch's theorem (see, e.g., [20]) we know that NPSPACE = PSPACE, but Σ_2^{P} is contained in PSPACE so Theorem 2 is stronger.

We do not know whether Algorithm 2 is guaranteed to terminate in polynomial time, in the best case, when a structurally smaller formula exists. Indeed, equational reasoning to transform a formula into some smaller one may conceivably require an exponential number of rewriting steps. On the contrary, were Algorithm 2 to always terminate in polynomial time, it would constitute a proof of SMALLMED being in NP.

A solution to the problem of finding a derivation without unbounded searches is to orient the equations of the system using the order on the size of the formula: from bigger to smaller. As a result, applying any rule (except commutativity) to a formula will simplify it with regard to $\leq_S$: every derivation will be a simplification.

Let us then consider the following "term rewriting system" extracted from System 1 by orienting its equations according to the decreasing structural ordering.

System 2.

$$(R1)\, \mathrm{m}(x, y, z) = \mathrm{m}(x, z, y) = \mathrm{m}(z, x, y),$$
$$(R2)\, \mathrm{m}(x, x, y) \longrightarrow x,$$
$$(R3)\, \mathrm{m}(\mathrm{m}(x, u, v), \mathrm{m}(y, u, v), z) \longrightarrow \mathrm{m}(\mathrm{m}(x, y, z), u, v),$$
$$(R4)\, \mathrm{m}(\bot, \top, x) \longrightarrow x,$$

for all variables x, y, z, u, v in L.

The rewrite rule $(R1)$, which is in fact $(M1)$, is kept the same, without orientation. Such systems are sometimes called *rewriting systems modulo commutativity* ([3]). A similar situation takes place when dealing with systems that involve commutative binary operations like $\vee$ or $\wedge$, and in such cases commutativity and associativity may be kept as equational rules. As explained in [21], in the case of median terms, commutativity $(M1)$ can be oriented by defining a total order on terms. Some effects of this orientation are the sorting of the terms by the application of the now-oriented commutativity rules, as well as the occasional blockage of derivation proofs (in these cases we thus lose completeness of the system).

By orientating the rules, we can more easily test a derivation between formulas.

Lemma 1. *Let ϕ and ψ be median formulas. If it exists, the derivation between ϕ and ψ is polynomial in the size of ϕ.*

Sketch of Proof. Every rewrite rule from System 2 save from the permutation $(R1)$ removes a median m from any formula it is applied to. If such a derivation exists, necessarily ϕ has more medians than ψ. The derivation between ϕ and ψ thus contains n steps, with n being the difference between the number of medians in ϕ and the number of medians in ψ. □

However, we may not be able to rewrite a formula into another equivalent one using System 2 (e.g., it is not possible to rewrite x into $\mathrm{m}(\bot, \top, x)$), much less into a normal form.

Example 6. *Consider $\phi = \mathrm{m}(\mathrm{m}(\mathrm{m}(x, y, z), u, v), y, z)$, which has for canonical form $\phi' = \mathrm{m}(\mathrm{m}(x, u, v), y, z)$. It is not possible to simplify ϕ into ϕ' using System 2, because it is not possible to apply any rule from System 2 to ϕ.*

Now, even though System 2 is no longer complete, the following result shows that it remains sound.

Proposition 1. *System 2 is sound but not complete.*

Sketch of Proof. Soundness follows from the fact that System 1 is sound (Theorem 1). Incompleteness follows from Example 6. □

VI. Conclusion and future work

In this paper, we discussed median-based formalism to efficiently represent monotone Boolean functions as well as polynomial functions over distributive lattices. This was achieved by proposing so-called median normal forms that were defined as median expressions that are minimal with respect to a structural ordering of formulas.

We also formalized the task of finding median formulas of smaller structural representation and investigated its computational complexity. This task turns out to be at most moderately intractable. In fact, we showed that the corresponding decision problem falls into Σ_2^{P} or $\mathsf{co\text{-}NP}$ according to whether the structural representation is given as part of the input.

However, the question of determining corresponding complexity lower bounds remains open. This and other complexity questions concerning decision problems that appear naturally in this median-based formalism are to be investigated in forthcoming collaborations.

References

[1] J. Aczél. The associativity equation re-revisited. In G. Erikson and Y. Zhai, editors, *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, pages 195-203. American Institute of Physics, Melville-New York, 2004.

[2] L. Amarú, P.-E. Gaillardon and G. De Micheli. Majority-inverter graph: A novel data-structure and algorithms for efficient logic optimization. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6. ACM, 2014.

[3] F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge university press, 1999.

[4] H.-J. Bandelt and G. C. Meletiou. The algebra of majority consensus. *Algebra Universalis*, 29(4):546–555, 1992.

[5] G. Birkhoff. *Lattice theory*, volume 25. American Mathematical Society New York, 1948.

[6] S. Burris and H. P. Sankappanavar. *A Course in universal algebra*, volume 78 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1981.

[7] M. Couceiro, S. Foldes and E. Lehtonen. Composition of post classes and normal forms of boolean functions. *Discrete Mathematics*, 306(24):3223–3243, 2006.

[8] M. Couceiro, E. Lehtonen, J.-L. Marichal and T. Waldhauser. An algorithm for producing median normal form representations for Boolean functions. *Proceedings of the Reed–Muller Workshop 2011*, 49–54, 2011.

[9] M. Couceiro and J.-L. Marichal. Characterizations of discrete sugeno integrals as polynomial functions over distributive lattices. *Fuzzy Sets and Systems*, 161(5):694–707, 2010.

[10] M. Couceiro and J.-L. Marichal. Aczélian n-ary semigroups, *Semigroup Forum*, 85 :1 (2012) 81–90.

[11] Y. Crama and P. L. Hammer. *Boolean functions: Theory, algorithms, and applications.* Cambridge University Press, 2011.

[12] J. W. Klop Term rewriting systems. *Handbook of Logic in Computer Science*, 2:1–116, 1992.

[13] M. Grabisch, J.-L. Marichal, R. Mesiar and E. Pap. *Aggregation functions.* Cambridge University Press, Cambridge, UK, 2009.

[14] A. Imre, G. Csaba, L. Ji, A. Orlov, G.H. Bernstein and W. Porod. Majority logic gate for magnetic quantum-dot cellular automata. *Science*, 311(5758):205–208, 2006.

[15] D. E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In *Automation of Reasoning*, pages 342–376. Springer, 1983.

[16] D. E. Knuth. *The art of computer programming: Fundamental algorithms*, volume 1. Reading, Massachusetts: Addison-Wesley, 1997.

[17] J.-L. Marichal. Weighted lattice polynomials. *Discrete Mathematics*, 309(4):814–820, 2009.

[18] H.S. Miller and R.O. Winder. Majority-logic synthesis by geometric methods. *IRE Transactions on Electronic Computers* , (1):89–90, 1962.

[19] E. L. Post. Polyadic groups, *Transactions of the American Mathematical Society*, 48:208–350, 1940.

[20] J. E. Savage. *Models of computation: Exploring the power of computing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1997.

[21] R. Veroff and W. McCune. First-order proof of a median algebra problem. http://www.cs.unm.edu/~veroff/MEDIAN_ALGEBRA/, September 2006.