Normal form systems generated by single connectives have mutually equivalent efficiency

Miguel Couceiro[1], Erkko Lehtonen[2], Pierre Mercuriali[1], Romain Péchoux[1], and Mathias Soeken[3]

1 **Université de Lorraine, CNRS, Inria, LORIA, F 54000 Nancy, France**
`{miguel.couceiro, pierre.mercuriali, romain.pechoux}@loria.fr`
2 **Technische Universität Dresden, Institut für Algebra, 01062 Dresden, Germany**
`Erkko.Lehtonen@tu-dresden.de`
3 **Integrated Systems Laboratory, EPFL, Lausanne, Switzerland**
`mathias.soeken@epfl.ch`

Abstract

In this paper we consider a Normal Form System (NFS) as being a factorization of the class of all Boolean functions into a composition of clones. This formalism includes classical normal forms such as DNF, CNF, ... We study the efficiency of NFSs that yield terms built using one or several connectives taken in a fixed order, and applied to literals and constants. Here, efficiency is measured by the minimal size of terms representing a function. Each clone is finitely generated but can have different sets of generators. We show that the choice of the generator used in a given NFS does not impact its efficiency, up to polynomial equivalence.

1998 ACM Subject Classification Theory of computation → Complexity theory and logic

Keywords and phrases Normal Form System, Clone theory, Efficiency

Digital Object Identifier 10.4230/LIPIcs.DICE.2016.1

1 Introduction

Normal Form Systems (NFS) induce sets of terms with a constrained structure: for example the Median Normal Form system (MNF) that corresponds to terms built using a connective m computing the ternary majority function MAJ (i.e. $MAJ(x, y, z) = 1$ iff $x + y + z \geq 2$) as basic connective. It was proven in [2] that the MNF is polynomially more efficient than the DNF, the CNF, the polynomial and the dual polynomial normal form systems.

In our context, polynomially more efficient means that for any Boolean function f, a term of minimal size representing f in one NFS can be converted into a term of minimal size representing f in the other NFS up to a polynomial overhead and that the converse does not hold.

A natural question to ask is then whether these results still hold for arbitrary clone generators. It is known that clones are finitely generated, but they can have different sets of generators. For instance, the clone SM of self-dual monotone Boolean functions is generated by any $2n + 1$-ary majority function, for $n \geq 1$. Are the NFSs corresponding to the 3-ary majority m and the 5-ary majority m_5 equivalently efficient? We show that they are by providing efficient conversion formulas. For instance, equations of the shape $\mathrm{m}(x, y, z) = \mathrm{m}_5(x, y, z, 0, 1)$ and $\mathrm{m}_5(x, y, z, t, u) = \mathrm{m}(\mathrm{m}(x, y, z), t, \mathrm{m}(\mathrm{m}(x, y, u), u, z))$ allow the efficient conversion from terms involving m_5 into terms involving m, and reciprocally.

More generally, it can be shown that the choice of any generator for SM does not impact the efficiency of the corresponding NFS. These results motivate the study of other efficient

9th International workshop on Developments in Implicit Computational complExity (DICE 2018).
Editor: Martin Avanzini; Article No. 1; pp. 1:1–1:5
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

NFSs, that is, NFSs that are equivalent to the MNF. These include, for instance, the Sheffer NF that is generated by the Sheffer function $x \uparrow y \equiv \neg(x \wedge y)$, one of the two generators of minimal arity of the clone Ω of all Boolean functions, the other being Peirce's arrow, $x \downarrow y \equiv \neg(x \vee y)$ (see, e.g., [9]).

Such efficient NFSs were studied and were shown to be equivalent for generators of minimal arity. A natural question is then whether these results still hold for arbitrary generators. In this paper, we answer positively to this question and show that it also holds for other non-efficient NFS (CNF, DNF, ...). The paper is organized as follows. After recalling the basic background on clones and NFSs in Section 2, we present in Section 3 the main result, stating the choice of generators does not impact the efficiency of NFSs, for efficient NFSs. In Section 3.2 we briefly discuss the remaining cases of NFSs, those generated by at least 2 non-trivial connectives.

2 Preliminaries

2.1 Boolean functions and clone theory

Let $\mathbb{B} = \{0,1\}$. We consider the usual linear order $0 < 1$. The set $\mathbb{B}^n$ is a Boolean (distributive and complemented) lattice of 2^n elements under the component-wise (strict) ordering of tuples $\leq$ (resp. $<$). Tuples can be viewed as words $(x_1, \ldots, x_n) = x_1 \cdots x_n$, and x^k denotes the word comprising of k copies of the letter x. The *complement* of a tuple $\mathbf{x} = x_1 \cdots x_n$ is defined as $\overline{\mathbf{x}} = \overline{x_1} \cdots \overline{x_n}$ with $\overline{x} = 1 - x$. Given a tuple $\mathbf{x} \in \mathbb{B}^n$, we define $\mathbf{x}\{b/i\} = x_1 \cdots x_{i-1}\ b\ x_{i+1} \cdots x_n$, with $b \in \mathbb{B}$ and $1 \leq i \leq n$.

A *Boolean function* is a map $f : \mathbb{B}^n \to \mathbb{B}$, for some positive integer n called the *arity* of f. A *class* of functions is a subset of $\bigcup_{n\geq 1} \mathbb{B}^{\mathbb{B}^n}$. For a fixed arity n, there are n different *projection maps* $(x_1, \ldots, x_n) \mapsto x_i, 1 \leq i \leq n$.

For a function $f : \mathbb{B}^n \to \mathbb{B}$, the *dual* of f is defined as $f^d(\mathbf{x}) := \overline{f(\overline{\mathbf{x}})}$.

Given $f : \mathbb{B}^n \to \mathbb{B}$, and $g_1, \ldots, g_n : \mathbb{B}^m \to \mathbb{B}$, the *composition* $f(g_1, \ldots, g_n)$ is the function of $\mathbb{B}^m \to \mathbb{B}$ defined by $f(g_1, \ldots, g_n)(\mathbf{x}) = f(g_1(\mathbf{x}), \ldots, g_n(\mathbf{x}))$, for all $\mathbf{x} \in \mathbb{B}^m$. This notion extends naturally to classes of functions $\mathcal{I}$ and $\mathcal{J}$. The *composition of* $\mathcal{I}$ *with* $\mathcal{J}$ is defined by $\mathcal{I} \circ \mathcal{J} := \{f(g_1, \ldots, g_n) \mid f \in \mathcal{I},\ g_1, \ldots, g_n \in \mathcal{J}\}$.

We recall the definition of *essential variables* (see, e.g., [3, 7, 8]). Given $f : \mathbb{B}^n \to \mathbb{B}$, the i^{th} argument of f is said to be *essential* in f, or that f *depends on* x_i, if there exists $\mathbf{x} \in \mathbb{B}^n$ such that $f(\mathbf{x}\{0/i\}) \neq f(\mathbf{x}\{1/i\})$. The number of essential variables in f is called the *essential arity of* f.

Two functions f and g are *equivalent* if each one can be obtained from the other by permutation of variables and addition or deletion of inessential variables (see, e.g., [3]). If two functions are equivalent, then they have the same essential arity.

A *clone* is a class $\mathcal{C}$ of Boolean functions that contains all projections and satisfies $\mathcal{C} \circ \mathcal{C} \subseteq \mathcal{C}$ (i.e., it is closed under composition). Clones of Boolean functions constitute an algebraic lattice (see [6]), where the meet is the intersection, the join of two clones is the smallest clone that contains their union, the largest clone is the clone of all Boolean functions $\Omega = \bigcup_{n\geq 1} \mathbb{B}^{\mathbb{B}^n}$, and the smallest clone is the clone of all projections I_c. These clones and the lattice are often called Post's classes and Post's lattice, respectively. For a more detailed presentation we refer to [4, 5, 2].

We now recall some clones of interest. For $x \in \mathbb{B}$, T_x denotes the clone of x-preserving functions: $T_x = \{f : \mathbb{B}^n \to \mathbb{B} \mid f(x, \ldots, x) = x\}$, and T_c denotes the clone of constant-preserving functions, i.e., $T_c = T_0 \cap T_1$. The clone $M = \{f \in \Omega \mid f(\mathbf{x}) \leq f(\mathbf{y})$ whenever $\mathbf{x} \leq \mathbf{y}\}$ is the clone of all monotone functions; and $M_x = M \cap T_x$ for $x \in \mathbb{B}$, $M_c = M \cap T_c$.

$S = \{f \in \Omega \mid f^d(\mathbf{x}) = f(\mathbf{x})\}$ is the clone of all self-dual functions. $SM = S \cap M$ denotes the clone of self-dual monotone functions. A set $X \subseteq \{0,1\}^n$ is said to be *x-separating* if there is $i, 1 \leq i \leq n$, such that for every $x_1 \ldots x_n \in X$ we have $x_i = x$. A function f is said to be a *clique function* (resp. a *co-clique* function) if $f^{-1}(1)$ is 1-separating (resp. if $f^{-1}(0)$ is 0-separating). A function f is said to be a *clique (resp. co-clique) function of rank* $k \geq 2$ if every subset $X \subseteq f^{-1}(1)$ (resp. every subset $X \subseteq f^{-1}(0)$) of size at most k is 1-separating (resp. 0-separating). For $m \geq 2$, U_m and W_m denote the clones of clique and co-clique functions of rank m, respectively; U_∞ and W_∞ denote the clones of all clique and co-clique functions, respectively, i.e., $U_\infty = \bigcap_{k \geq 2} U_k$ and $W_\infty = \bigcap_{k \geq 2} W_k$; $M_c U_m = M_c \cap U_m$ and $M_c W_m = M_c \cap W_m$ for $m = 2, \ldots, \infty$.

Let F be a set of Boolean functions. The clone *generated* by F, denoted $\mathcal{C}(F)$, is the smallest clone that contains F and is defined as follows:

$$\bigcap \{\mathcal{C} \mid \mathcal{C} \text{ a clone, } F \subseteq \mathcal{C}\}.$$

For $f \in \Omega$, we write simply $\mathcal{C}(f)$ for $\mathcal{C}(\{f\})$.

2.2 Terms and normal form systems

We mostly adopt the terminology of [1]. A set Σ is a set of *function symbols*, also called *connectives*. Each $\alpha \in \Sigma$ is associated with a strictly positive integer n called the *arity* of α and denoted by $\mathrm{ar}(\alpha)$. Let $\top$ and $\bot$ be two constant symbols denoting the truth and false values. Let X be a countable set of variables. For a signature Σ such that $\Sigma \cap X = \emptyset$, the set of all *terms over* X, denoted by $T(\Sigma \cup \{\top, \bot\}, X)$, is recursively defined as follows:

- every variable in X is a term;
- the constants $\top$ and $\bot$ are terms;
- $\forall n > 0$, $\forall \alpha \in \Sigma$ such that $\mathrm{ar}(\alpha) = n$, $\forall t_1, \ldots, t_n \in T(\Sigma, X)$, $\alpha(t_1, \ldots, t_n)$ is a term.

In this paper we will use letters $s, t, s', t', \ldots$ to designate terms in $T(\Sigma \cup \{\top, \bot\}, X)$.

Terms can be put in correspondence with functions by *interpreting* them as Boolean functions. We denote by $[\alpha]$ the interpretation of a connective α. For instance $\mathrm{maj} = [\mathrm{m}]$. The constants $\top$ and $\bot$ are interpreted as 1 and 0, respectively. The *interpretation* $[t]$ of a term t is defined inductively on the structure of terms. The *interpretation* of a set of terms S is defined as $[S] = \{[t] : t \in S\}$.

Two terms t_1, t_2 are said to be *equivalent*, denoted by $t_1 \equiv t_2$, if $[t_1] = [t_2]$.

▶ Remark. As functions obtained from f by addition of inessential variables have essentially the same representation, with no danger of ambiguity we naturally extend this notion of functional representation by terms, to functions with inessential variables. For instance, $\top$ and $\bot$ will represent any constant 1 and any constant 0 function, respectively. The same applies to the ith projection of any arity.

Any subword of a term t that is itself a term is called a *subterm* of t. Given a sequence of distinct connectives $\alpha_1 \cdots \alpha_n$, we denote by $T(\alpha_1 \ldots \alpha_n)$ the set of all terms $t \in T(\{\alpha_1 \ldots \alpha_n\} \cup \{\top, \bot\}, X)$ such that no subterm of t of the form $\alpha_j(t_1, \ldots, t_n)$ ($j \in \{1, \ldots, n\}$) contains a connective α_i with $i < j$.

By abuse of notation, we denote by $\mathcal{C}(\{\alpha_1, \ldots, \alpha_k\})$ the smallest clone that contains $\{[\alpha_1], \ldots, [\alpha_k]\}$ for a fixed interpretation $[\ ]$. We will say that the clone $\mathcal{C}(\{\alpha_1, \ldots, \alpha_k\})$ is *generated by the set* $\{\alpha_1, \ldots, \alpha_k\}$. We also say that $\{\alpha_1, \ldots, \alpha_k\}$ is a generator set of $\mathcal{C}$.

It is well known that every Boolean function can be represented in disjunctive normal form. This fact can be restated as $\Omega = \mathcal{C}(\vee) \circ \mathcal{C}(\wedge) \circ \mathcal{C}(\neg)$. This illustrates the fact that we can express Ω as a factorization into proper subclones, which was the basis for the notion of normal form systems proposed in [2]. We adapt this notion slightly to focus on terms instead of functions.

▶ **Definition 1** (Normal form systems). Let $\alpha_1, \ldots, \alpha_n$ be distinct connectives and $[\,]$ an interpretation. If $[T(\alpha_1 \cdots \alpha_n)] = \Omega$, then the couple $(T(\alpha_1 \cdots \alpha_n), [\,])$ is called a *normal form system* or *NFS* for short. We may refer to the sequence of connectives $\alpha_1 \cdots \alpha_n$ as the *generators* of the NFS. If there exists an $i \in \{1, \ldots, n\}$ such that $(T(\alpha_1 \cdots \alpha_{i-1}\alpha_{i+1} \cdots \alpha_n), [\,])$ is an NFS, then $(T(\alpha_1 \cdots \alpha_n), [\,])$ is said to be *redundant*, otherwise it is said to be *irredundant*.

For instance, the NFS $T(\wedge\,\mathrm{m}\,\neg)$ is redundant because $[T(\wedge\,\mathrm{m}\,\neg)] = [T(\mathrm{m}\,\neg)]$. In this paper we only consider irredundant NFSs.

▶ Remark. If $T(\alpha_1 \cdots \alpha_n)$ is an NFS, then $[T(\alpha_1 \cdots \alpha_n)] = \mathcal{C}(\alpha_1) \circ \cdots \circ \mathcal{C}(\alpha_n) \circ I = \Omega$, where I is the clone of constants and projections.

In this paper, $\wedge$, $\vee$, $\neg$, $\oplus$ are interpreted as the usual conjunction, disjunction, negation, and sum modulo 2. The connective m is interpreted as the ternary majority function maj. By fixing the standard interpretation of the usual connectives, we will use the shorthand notation $T(\alpha_1 \cdots \alpha_n)$ to denote the NFS $(T(\alpha_1 \cdots \alpha_n), [\,])$.

2.3 Efficiency of representations

We denote by $|t|_\alpha$ the number of occurrences of the connective α in the term t. The *size* of a term t is denoted by $|t|$, and it is defined as the number of all connectives in t; $|t| = \sum_{\alpha \in \Sigma} |t|_\alpha$. Note that $\Sigma \cap X = \emptyset$, hence variables are not counted. For instance, if $t = (x \vee y) \wedge (y \vee z) \wedge (z \vee x)$ then $|t| = |t|_\wedge + |t|_\vee = 2 + 3 = 5$.

▶ **Definition 2** ($\mathbf{A}$-complexity). Let $\mathbf{A} = T(\alpha_1 \cdots \alpha_k)$ be an NFS. For a function $f \in \Omega$ we define the $\mathbf{A}$-*complexity* of f, denoted $C_\mathbf{A}(f)$, by

$$C_\mathbf{A}(f) = \quad \min\{|t| : t \in T(\alpha_1 \cdots \alpha_k),\ [t] = f\}.$$

▶ **Example 3.** Consider the ternary majority function maj. Then $C_{T(\mathrm{m}\,\neg)}(\mathrm{maj}) = 1$ because $\mathrm{m}(x, y, z)$ is the smallest term in $T(\mathrm{m}\,\neg)$ whose interpretation is equal to maj. However, $C_{T(\wedge\vee\neg)}(\mathrm{maj}) = 5$ because $(x \vee y) \wedge (y \vee z) \wedge (z \vee x)$ is the smallest term in $T(\wedge \vee \neg)$ whose interpretation is equal to maj.

▶ **Definition 4** (Efficiency). For two NFSs $\mathbf{A}$ and $\mathbf{B}$, we say that $\mathbf{A}$ is *polynomially as efficient as* $\mathbf{B}$, denoted $\mathbf{A} \preceq \mathbf{B}$, if there is a polynomial P with non-negative integer coefficients such that $C_\mathbf{A}(f) \leq P(C_\mathbf{B}(f))$ for all $f \in \Omega$.

If $\mathbf{A} \not\preceq \mathbf{B}$ and $\mathbf{B} \not\preceq \mathbf{A}$, then we say that $\mathbf{A}$ and $\mathbf{B}$ *provide representations of incomparable complexity* and we write $\mathbf{A}||\mathbf{B}$. If $\mathbf{A} \preceq \mathbf{B}$ but $\mathbf{B} \not\preceq \mathbf{A}$, we say that $\mathbf{A}$ is *polynomially more efficient than* $\mathbf{B}$ and we write $\mathbf{A} \prec \mathbf{B}$. If $\mathbf{A} \preceq \mathbf{B}$ and $\mathbf{B} \preceq \mathbf{A}$, we say that $\mathbf{A}$ and $\mathbf{B}$ are *equivalent* or *provide representations of equivalent complexity*, and we write $\mathbf{A} \sim \mathbf{B}$.

Remark that $\preceq$ is a preorder on the set of all NFSs, but not a total order in general (see, e.g., [2]). Moreover, $\sim$ is an equivalence relation.

3 Main result

3.1 Single connective

We can now present the main result of the paper that basically states that the choice of generator does not impact the efficiency of NFSs generated by a single connective.

▶ **Theorem 5.** *Let α, β be non-associative connectives that are generators of the clone $\mathcal{C}$. If $T(\alpha\neg)$ and $T(\beta\neg)$ are NFSs then $T(\alpha\neg) \sim T(\beta\neg)$.*

As this result holds for clones generated by a single non-associative connective, it holds in particular for any generator of the clone $\mathcal{C} \in \{SM, MU_k, M_cU_k, MW_k, M_cW_k, M, M_c, M_1, M_0, U_\infty, T_c, U_\infty, W_\infty, T_cW_\infty, T_0, T_1, T_c, S, S_c, \Omega\}$.

3.2 NFSs generated by more than one connective

We conclude with a brief discussion on NFSs generated by more than one connective. They are exactly those NFSs whose connectives are generators of the clones V, V_0, V_1, V_c of disjunctions, the clones Λ, Λ_0, Λ_1, Λ_c of conjunctions, and the clones L, LS, L_0, L_1, L_c of linear functions in Post's Lattice. The corresponding generators of minimal arity are the binary disjunction $\vee$, the binary conjunction $\wedge$, and the binary sum modulo 2, $\oplus$.

In [2], the authors proved that the disjunctive, conjunctive, polynomial and dual polynomial normal forms are pairwise incomparable and less efficient than the majority (or median). They had considered as their connectives the binary disjunction and conjunction, and the ternary sum. Do these results still hold with connectives of arbitrary arity? It is the case: in fact, all these connectives are interpreted as associative functions. At most, expressing an n-ary generator with a binary one, for the same clone, requires $n-1$ binary connectives. Reciprocally, expressing a binary generator with an n-ary one will require only a single n-ary generator (along with some constants). Furthermore, both conversions are linear.

References

1 Franz Baader and Tobias Nipkow. *Term rewriting and all that.* Cambridge university press, 1999.

2 Miguel Couceiro, Stephan Foldes, and Erkko Lehtonen. Composition of post classes and normal forms of Boolean functions. *Discrete Mathematics*, 306(24):3223–3243, 2006.

3 Miguel Couceiro, Erkko Lehtonen, and Tamás Waldhauser. Decompositions of functions based on arity gap. *Discrete Mathematics*, 312(2):238–247, 2012.

4 Klaus Denecke and Shelly L. Wismath. *Universal Algebra and Coalgebra.* World Scientific, 2009.

5 Dietlinde Lau. *Function Algebras on Finite Sets: Basic Course on Many-Valued Logic and Clone Theory.* Springer Science & Business Media, 2006.

6 Emil L. Post. *The Two-Valued Iterative Systems of Mathematical Logic*, volume 5, pages 1–122. Princeton, 1941.

7 Arto Salomaa. On essential variables of functions, especially in the algebra of logic. *Annales Academiæ Scientiarum Fennicæ*, Series A I 339:11, 1963.

8 Ross Willard. Essential arities of term operations in finite algebras. *Discrete Mathematics*, 149(1-3):239–259, 1996.

9 Eustachy Żyliński. Some remarks concerning the theory of deduction. *Fundamenta Mathematicae*, 7(1):203–209, 1925.