Minor complexity of discrete functions

Slavcho Shtrakov
Department of Computer Science, South-West University, Blagoevgrad, Bulgaria

Received 4 June 2018
Revised 16 July 2018
Accepted 17 July 2018

Abstract

In this paper we study a class of complexity measures, induced by a new data structure for representing k-valued functions (operations), called minor decision diagram. When assigning values to some variables in a function the resulting functions are called subfunctions, and when identifying some variables the resulting functions are called minors. The sets of essential variables in subfunctions of f are called separable in f.

We examine the maximal separable subsets of variables and their conjugates, introduced in the paper, proving that each such set has at least one conjugate. The essential arity gap $gap(f)$ of the function f is the minimal number of essential variables in f which become fictive when identifying distinct essential variables in f. We also investigate separable sets of variables in functions with non-trivial arity gap. This allows us to solve several important algebraic, computational and combinatorial problems about the finite-valued functions.

Keywords Separable set, Subfunction, Identification minor, Minor decision diagram, Minor complexity
Paper type Original Article

1. Introduction

The complexity of finite operations is still one of the fundamental tasks in the theory of computation and besides classical methods like substitution or degree arguments a bunch of combinatorial, and algebraic techniques have been introduced to tackle this extremely difficult problem.

A *logic gate* is a physical device that realizes a Boolean function. A *logic circuit* is a direct acyclic graph in which all vertices except input vertices carry the labels of gates. When realizing n-variable k-valued functions the circuit is called the (k, n)-*circuit or Multi-Valued Logic circuit (MVL-circuit)*.

To move from logical circuits to MVL-circuits, researchers attempt to adapt CMOS (complementary metal oxide semiconductor), I^2L (integrated injection logic) and ECL (emitter-coupled logic) technologies to implement the many-valued and fuzzy logics gates. The MVL-circuits offer more potential opportunities for the improvement of present VLSI circuit

Declarations of interest: None.

An idea for a mesure of the minor complexity of functions was presented by M. Couceiro, E. Lehtonen and T. Waldhauser in [5,6], named parametrized arity gap. However, these results, the present paper and [16] are suitable basis for future investigation of the problems for minor complexities and reconstruction of functions from their MDDs.

Publishers note: The publisher wishes to inform readers that the article "Minor complexity of discrete functions" was originally published by the previous publisher of *Applied Computing and Informatics* and the pagination of this article has been subsequently changed. There has been no change to the content of the article. This change was necessary for the journal to transition from the previous publisher to the new one. The publisher sincerely apologises for any inconvenience caused. To access and cite this article, please use Shtrakov, S. (2021), "Minor complexity of discrete functions", *Applied Computing and Informatics*. Vol. 17 No. 1, pp. 108-130. The original publication date for this paper was 24/07/2018.

Applied Computing and Informatics
Vol. 17 No. 1, 2021
pp. 108-130
Emerald Publishing Limited
e-ISSN: 2210-8327
p-ISSN: 2634-1964
DOI 10.1016/j.aci.2018.07.003

designs. For instance, MVL-circuits are well-applied in memory technology as flash memory, dynamic RAM, and in algebraic circuits [4].

In this paper we investigate a method for reduction of finite valued functions, namely by their identification minors. This method is a basic model of computing with MVL-circuits corresponding to collapsing of some inputs in the circuits. We, also study the computational complexity of this method and classify the functions in finite algebras for small values of k and n under such complexity.

Computational complexity is examined in concrete and abstract terms. The concrete analysis is based on models that capture the exchange of space for time. It is also performed via the knowledge about circuit complexity of functions. The abstract analysis is done via complexity classes, the classification of data structures, functions etc. by the time and/or space they need.

There are two key methods for reduction (computing) of the k-valued functions which are realized by assigning constants or variables to their inputs. Then the resulting objects are: subfunctions or minors, respectively. These reductions are also naturally suited to complexity measures, which illustrate "difficulty" of computing as the number of subfunctions, separable sets, and minors of the functions.

Another topic in complexity theory is to classify finite functions by their complexity such that the functions are grouped into equivalence classes with same evaluations of the corresponding complexities. Each equivalence relation in the algebra P_k^n of k-valued functions determines a transformation group whose orbits are the equivalence classes (see [8,10,12]. Using the lattice of Restricted Affine Groups (RAG) in [15] we have obtained upper bounds of different combinatorial parameters of several natural equivalences in P_k^n for small values of k and n. In the present paper we follow this line to study assigning (not necessarily unique) variable names to some of the input variables in a function f. This method of computing consists of equalizing the values of several inputs of f.

Section 2 introduces the basic definitions and notation of separable sets, subfunctions, minors, arity gap, etc. An important result, namely if a function has non-trivial arity gap then all its sets of essential variables are separable, complements this section. Section 3 examines the ordered decision diagrams (ODD), minor decomposition trees (MDTs) and minor decision diagrams (MDDs) of k-valued functions. In Section 3.3 we treat the minor complexities of functions with their classifications by the transformation groups. Section 4 is an illustration of the results in the paper applied to the simplest case of Boolean functions. In the Appendix we provide a classification of all ternary Boolean functions with respect to the minor complexity.

2. Subfunctions and minors of functions

A *discrete function* f is defined as a mapping: $f : A \to B$ where the domain $A = \times_{i=1}^{n} A_i$ and the range B are non-empty finite or countable sets. Let $X = \{x_1, x_2, \ldots\}$ be a countable set of variables and let $X_n = \{x_1, x_2, \ldots, x_n\}$ denote the set of the first n variables in X. Let k be a natural number with $k \geqslant 2$. Let Z_k denote the set $Z_k = \{0, 1, \ldots, k-1\}$. The operations addition "$\oplus$" and product "." modulo k constitute Z_k as a ring. An *n-ary k-valued function (operation) on* Z_k is a mapping $f : Z_k^n \to Z_k$ for some natural number n, called *the arity* of f. P_k^n denotes the set of all n-ary k-valued functions and $P_k = \cup_{n=1}^{\infty} P_k^n$ is called *the algebra of k-valued logic*. It is well-known fact that there are k^{k^n} functions in P_k^n. For simplicity, let us assume that throughout the paper we shall consider k-valued functions, only.

For a given variable x and $\alpha \in Z_k$, x^α is defined as follows:

$$x^\alpha = \begin{cases} 1 & if \quad x = \alpha \\ 0 & if \quad x \neq \alpha. \end{cases}$$

The ring-sum expansion (RSE) of a function f is the sum modulo k of a constant and products of variables x_i or x_i^α, (for $\alpha, \alpha \in Z_k$) of f. For example, $1 \oplus x_1 x_2^2$ is a RSE of the function f in

the algebra P_3^2, with $f(1,2) = 2, f(2,2) = 0$ and $f = 1$, otherwise. Any k instances of the same product in the RSE can be eliminated since they sum to 0. Throughout the present paper, we shall use RSE-representation of functions.

Let $f \in P_k^n$ and let $var(f) = \{x_1, \ldots, x_n\}$ be the set of all variables, which occur in f. We say that the i-th variable $x_i \in var(f)$ is *essential* in f, or f *essentially depends* on x_i, if there exist values $a_1, \ldots, a_n, b \in Z_k$, such that

$$f(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n) \neq f(a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n).$$

The set of all essential variables in the function f is denoted $Ess(f)$ and $ess(f) = |Ess(f)|$. The variables from $var(f)$ which are not essential in $f \in P_k^n$ are called *inessential* or *fictive*.

Let x_i be an essential variable in f and let c be a constant from Z_k. The function $g = f(x_i = c)$ obtained from $f \in P_k^n$ by assigning the constant c to the variable x_i is called a *simple subfunction of* f (sometimes termed a *cofactor* or a *restriction*). When g is a simple subfunction of f we write $f \succ g$. The transitive closure of $\succ$ is denoted $\succeq$. $Sub(f) = \{g \mid f \succeq g\}$ is the set of all subfunctions of f and $sub(f) = |Sub(f)|$.

Let $f \succeq g$, $\overline{c} = (c_1, \ldots, c_m) \in Z_k^m$ and let $M = \{x_1, \ldots, x_m\} \subset X$ with $f = g_m \succ g_{m-1} \succ \ldots \succ g_1 \succ g, g = g_1(x_1 = c_1)$, and $g_i = g_{i+1}(x_{i+1} = c_{i+1})$ for $i = 1, \ldots, m-1$. Then we write $f \succeq_M^{\overline{c}} g$ or equivalently, $g = f(x_1 = c_1, \ldots, x_m = c_m)$. For brevity, sometimes we shall also use the notation $f \succeq_M g$ or $f \succeq g$.

We say that each subfunction g of f is a reduction to f via the *subfunction relationship*.

Definition 2.1. A non-empty set M of essential variables in the function f is called *separable* in f if there exists a subfunction $g, f \succeq g$ such that $M = Ess(g)$. $Sep(f)$ denotes the set of all the separable sets in f and $sep(f) = |Sep(f)|$.

The theory of separable sets (TSS) has been developed in the work of many mathematicians since the middle of the last century – K. Chimev [2], A. Salomaa [11], J. Denev, I. Gyudzhenov [7], Sl. Shtrakov [3] etc. TSS is important to avoid any redundancies when computing discrete functions and other structures as graphs [2], terms [14], etc.

Let x_i and x_j be two distinct essential variables in f. The function h is obtained from $f \in P_k^n$ by *identifying (collapsing) the variables* x_i *and* x_j, if

$$h(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n) = f(a_1, \ldots, a_{i-1}, a_j, a_{i+1}, \ldots, a_n),$$

for all $(a_1, \ldots, a_n) \in Z_k^n$.

Briefly, when h is obtained from f, by identifying the variable x_i with x_j, we write $h = f_{i \leftarrow j}$ and h is called *a simple identification minor of* f. Clearly, $ess(f_{i \leftarrow j}) < ess(f)$, because $x_i \notin Ess(f_{i \leftarrow j})$, but it has to be essential in f. When h is a simple identification minor of f we write $f \succ h$. The transitive closure of $\succ$ is denoted $\succeq$. $Mnr(f) = \{h \mid f \succeq h\}$ is the set of all distinct minors of f and $mnr(f) = |Mnr(f)|$. Let $h, f \succeq h$ be an identification minor of f. The natural number $r = ess(f) - ess(h), r \geqslant 1$ is called the *order* of the minor h of f.

We say that each minor h of f is a reduction to f via the *minor relationship*.

Let $Mnr_m(f)$ denote the set $Mnr_m(f) = \{g \mid g \in Mnr(f)\ \&\ ess(g) = m\}$ and let $mnr_m(f) = |Mnr_m(f)|$, for all $m, m < n$.

Let $f \in P_k^n$ be an n-ary k-valued function. The *essential arity gap* (shortly *arity gap* or *gap*) of f is defined as follows

$$gap(f) = ess(f) - \max_{h \in Mnr(f)} ess(h).$$

Let $2 \leqslant p \leqslant m$. We let $G_{p,k}^m$ denote the set of all k-valued functions which essentially depend on m variables whose arity gap is equal to p, i.e.

$$G^n_{p,k} = \{f \in P^n_k \mid ess(f) = m \ \& \ gap(f) = p\}.$$

We say that the arity gap of f is *non-trivial* if $gap(f) \geqslant 2$. It is natural to expect that the functions with "huge" gap, have to be more simple for realization by MVL-circuits and functional schemas when computing by identifying variables.

An upper bound of $gap(f)$ for Boolean functions is found in K. Chimev [2] and A. Salomaa [11], showing that $gap(f) \leqslant 2$. In [18] R. Willard also proved that if a function $f : A^n \to B$ depends on n variables and $k < n$, where $k = |A|$ then $gap(f) \leqslant 2$. It is clear that $gap(f) \leqslant n$. Thus in all cases $gap(f) \leqslant min(n, k)$.

A complete description of Boolean functions with non-trivial arity gap is presented in [13]. In [16] these results are extended including the functions of k-valued logic, $k \geqslant 2$. In [17], a special class of functions - namely the class of symmetric k-valued functions with non-trivial arity gap, is investigated.

Definition 2.2. Two functions g and h are called *equivalent (non-distinct as mappings)* (written $g \equiv h$) if g can be obtained from h by permutation of variables, introduction or deletion of inessential variables.

As mentioned earlier, there are two general ways for reduction of functions - by subfunctions or by minors. The complexities of these processes we call the *subfunction or minor complexities*, respectively.

An obvious difference between these concepts is the following: Each identification minor can be decomposed into subfunctions, but there are subfunctions which can not be decomposed into minors. For example, we have

$$f_{i \leftarrow j} = \bigoplus_{m=0}^{k-1} x_j^m . f(x_i = m, x_j = m)$$

for all $f, f \in P^n_k$, where $f \succ f(x_i = m, x_j = m)$ and $f \succ f_{i \leftarrow j}$.

Let $f = x_1 \oplus x_2 \oplus x_3$ be a Boolean function. It is easy to see that the subfunction $f(x_1 = 1) = x_2 \oplus x_3 \oplus 1$ can not be decomposed into any minors of f.

Roughly spoken, the complexity of functions, is a mapping (evaluation) $Val : P^n_k \to \mathbb{N}$ with $Val(x) = c$ for all $x \in X$ and for some natural number $c \in \mathbb{N}$, called the *initial value* of the complexity, and $Val(f) \geqslant c$ for all $f \in P^n_k$.

The concept of complexity of functions is based on the "difficulties" when computing several resulting objects as subfunctions, implementations, separable sets, values, superpositions, minors, etc.

As mentioned, the computational complexities $sub(f)$, $imp(f)$ and $sep(f)$ are used in [15] to classify the functions from the algebra P^n_k. These complexities are invariants under the action of the suitable transformation groups.

Many computations, constructions, processes, translations, mappings and so on, can be modeled as stepwise transformations of objects known as reduction systems. *Abstract Reduction Systems (ARS)* play an important role in various areas such as abstract data type specification, functional programming, automated deductions, etc. [9] The concepts and properties of ARS also apply to other rewrite systems such as string rewrite systems (Thue systems), tree rewrite systems, graph grammars, etc. For more detailed facts about ARS we refer to J.W. Klop and Roel de Vrijer [9]. An ARS in P^n_k is a structure $W = \langle P^n_k, \{\to_i\}_{i \in I}, \rangle$, where $\{\to_i\}_{i \in I}$ is a family of binary relations on P^n_k, called *reductions or rewrite relations*. For a reduction $\to_i$ the transitive and reflexive closure is denoted $\twoheadrightarrow_i$. A function $g \in P^n_k$ is a *normal form* if there is no $h \in P^n_k$ such that $g \to_i h$. In all different branches of rewriting two basic concepts occur, known as termination (guaranteeing the existence of normal forms) and confluence (securing the uniqueness of normal forms).

A reduction $\rightarrow_i$ has the *unique normal form property* (UN) if whenever $t, r \in P_k^n$ are normal forms obtained by applying the reductions $\rightarrow_i$ on a function $f \in P_k^n$ then t and r are equivalent (non-distinct as mappings).

The computations on functions proposed in the present paper can be regarded as an ARS, namely: $W = \langle P_k^n, \{\succ, \succ\}\rangle$. Next, we show that $\succ$ completes the reduction process with unique normal form, whereas $\succ$ has not unique normal form property.

A reduction $\rightarrow$ is *terminating* (or *strongly normalizing* SN) if every reduction sequence $f \rightarrow f_1 \rightarrow f_2 \ldots$ eventually must terminate. A reduction $\rightarrow$ is *weakly confluent* (or *has weakly Church-Rosser property* WCR) if $f \rightarrow r$ and $f \rightarrow v$ imply that there is $w \in P_k^n$ such that $r \twoheadrightarrow w$ and $v \twoheadrightarrow w$.

Theorem 2.3.

(i) *The reduction $\succ$ is UN;*

(ii) *The reduction $\succ$ is SN, but it is not WCR.*

Proof. (i) (SN) If $f \succ g$ then $ess(f) > ess(g)$. Since the number of essential variables $ess(f_i)$ of the functions f_i in any reduction sequence $f \succ f_1 \succ \ldots \succ f_i \succ \ldots$ strongly decrease, it follows that the sequence eventually must terminate, i.e. the reduction is terminating.

(WCR) Let f be a function and $f \succ g$, and $f \succ h$. Let t and r be normal forms such that $g \succeq t$ and $h \succeq r$. Note that each normal form is a resulting minor obtained by collapsing all the essential variables in f. Hence, $ess(t) \leqslant 1$ and $ess(r) \leqslant 1$. Then we have $t = f(x_j, \ldots, x_j)$, for some $x_j \in Ess(f)$ and $r = f(x_i, \ldots, x_i)$, for some $x_i \in Ess(f)$, and hence,

$$t = f(x_j, \ldots, x_j) \equiv f(x_i, \ldots, x_i) = r.$$

Now, (i) follows from Newman's Lemma (Theorem 1.2.1. [9]), which states that WCR & SN $\Rightarrow$ UN.

(ii) Clearly, each value of a function f with $ess(f) > 0$ is an its subfunction normal form and each subfunction of f which is not a constant is not a normal form. Hence $\succ$ is SN. Every non-constant functions have at least two values (normal forms), which shows that $\succ$ is not WCR and UN. □

Thus, for each function $f(x_1, \ldots, x_n)$ that depends on all its variables, the function $f(x, \ldots, x)$ is the identification minor normal form of f.

An essential variable x_i in a function $f \in P_k^n$ is called a *strongly essential variable* in f if there is a constant c_i such that $Ess(f(x_i = c_i)) = Ess(f) \backslash \{x_i\}$. The set of all strongly essential variables in f is denoted $SEss(f)$.

The following lemma is independently proved by K. Chimev [2] and A. Salomaa [11] in different variations.

Lemma 2.4. [2] *Let f be a function. If $ess(f) > 1$ then f has at least two strongly essential variables, i.e. $ess(f) > 1 \Rightarrow |SEss(f)| > 1$.*

We are going to prove several results in TSS which will be used later to show relationship between arity gap and separable sets.

Lemma 2.5. *Let $N \in Sep(f)$. If there exist m constants $c_1, \ldots, c_m \in Z_k$ such that $N \cap Ess(g_i) = \emptyset$ where $g_i = f(x_i = c_i)$ for $1 \leqslant i \leqslant m$ then $M \cup N \in Sep(f)$ for all $M \neq \emptyset$, $M \subseteq \{x_1, \ldots, x_m\}$.*

Proof. It suffices to look only at the set $M = \{x_1, \ldots, x_m\}$. First, assume that $M \cap N = \emptyset$ and without loss of generality let us assume $N = \{x_{m+1}, \ldots, x_s\}, m < s \leqslant n$. Since $N \in Sep(f)$, there exists a vector of constants, say $\overline{d} = (d_{s+1}, \ldots, d_n) \in Z_k^{n-s}$ such that $N \subseteq Ess(g)$, where

$$g = f(x_{s+1} = d_{s+1}, \ \ldots, x_n = d_n).$$

Let us fix an arbitrary variable from N, say the variable $x_s \in N$. Then there exist $s-m-1$ constants $d_{m+1}, \ldots, d_{s-1} \in Z_k$ such that $x_s \in Ess(h)$ where

$$h = g(x_{m+1} = d_{m+1}, \ldots, x_{s-1} = d_{s-1}).$$

We have to prove that $M \subseteq Ess(h)$. Let us suppose the opposite, i.e. there is a variable, say $x_1 \in M$ which is inessential in h. Since $x_1 \in \{x_1, \ldots, x_m\}$, there is a value $c_1 \in Z_k$ such that $N \cap Ess(t) = \emptyset$ where $t = f(x_1 = c_1)$. Our supposition shows that $h = h(x_1 = c_1)$ and hence, $N \cap Ess(h) = \emptyset$, i.e. $x_s \notin Ess(h)$, which is a contradiction. Consequently, $M = Ess(h)$. Then $g \succeq h$ implies $M \subseteq Ess(g)$ and hence, $M \cup N = Ess(g)$ which establishes that $M \cup N \in Sep(f)$.

Second, let $M \cap N \neq \emptyset$. Then we can pick $P = M \backslash N$ and hence, $P \subseteq \{x_1, \ldots, x_m\}$, $P \cap N = \emptyset$, and $N \in Sep(f)$. As shown, above $P \cup N \in Sep(f)$ and $M \cup N \in Sep(f)$, as desired. □

Corollary 2.6. *Let x_i and x_j be two distinct essential variables in f. If there is a constant $c, c \in Z_k$ such that $f(x_i = c)$ does not essentially depend on x_j then $\{x_i, x_j\} \in Sep(f)$.*

Definition 2.7. Let M be an inseparable set in f. A subset M_1 of M is called *a maximal separable subset* of M in f, if M_1 is separable in f and for each M_2, $M_1 \underset{\neq}{\subseteq} M_2 \subseteq M$ it is held $M_2 \notin Sep(f)$.

The set of all maximal separable subsets of M in a function f is denoted by $Max(M, f)$.

Definition 2.8. Let M_1, $M_1 \in Max(M, f)$ be a maximal separble subset of the inseparable set M in f. The essential variable x_i in f is called an *essential conjugate* of the set M_1 in f if for each subfunction $g, f \succ m_2 g$, where $M_2 = M \backslash M_1$ we have $M_1 \in Sep(g)$ and $x_i \in Ess(g)$.

Example 2.9. Let f be the following function $f = x_1^0 x_2^1 \oplus x_2^0 x_3^1 x_4^2 (mod\, 3)$. It is easy to see that $M = \{x_1, x_3, x_4\} \notin Sep(f)$ and $Max(M, f) = \{\{x_1\}, \{x_3, x_4\}\}$. Clearly, x_2 is an essential conjugate of both $\{x_1\}$ and $\{x_3, x_4\}$ in f.

The next theorem was proven by K. Chimev, and it is an important step to achieve a series of results concerning identification minors of functions [2,3].

Theorem 2.10. [2] *Let $f \in P_k^n$, $\emptyset \neq M \notin Sep(f)$, $M_1 \in Max(M, f)$ and $M_2 = M \backslash M_1$. Then for each subfunction $g, f \succeq M_2 g$ of f, there exists a variable $x_i, x_i \in Ess(f) \backslash M$ such that $x_i \in Ess(g)$ and $M_1 \in Sep(g)$.*

Note that Theorem 2.10 does not provide the existence of at least one essential conjugate of any maximal separable subset of M. We are going to strengthen Theorem 2.10 in this direction. First, we shall prove the following lemma.

Lemma 2.11. *Let M be a non-empty inseparable set of essential variables in f, $L = Ess(f) \backslash M$ and let $M_1 \in Max(M, f)$. Then there exists a subfunction $g, f \succeq_L g$ such that $M_1 \neg \subseteq Ess(g)$.*

Proof. Without loss of generality let us assume that

$$M_1 = \{x_1, \ldots, x_m\},\ M = \{x_1, \ldots, x_{m+p}\} \text{ and } L = \{x_{m+p+1}, \ldots, x_n\}.$$

Indeed, suppose this were not the case. Then $M_1 \subseteq Ess(g)$ for each $\bar{c}$, $\bar{c} \in Z_k^{n-m-p}$. Since the variable x_{m+1} is essential in f, there is a vector of constants $\bar{b} = (b_{m+p+1}, \ldots, b_n) \in Z_k^{n-m-p}$, such that $x_{m+1} \in Ess(t)$, where

$$t = f(x_{m+p+1} = b_{m+p+1}, \ldots, x_n = b_n).$$

Let $\overline{a} = (a_{m+2}, \ldots, a_{m+p}) \in Z_k^{p-1}$ be a vector of constants from Z_k such that $x_{m+1} \in Ess(v)$, where

$$v = t(x_{m+2} = a_{m+2}, \ldots, x_{m+p} = a_{m+p}).$$

Theorem 2.10 implies $M_1 \subseteq Ess(v)$. Clearly, $Ess(v) \subseteq M_1 \cup \{x_{m+1}\}$. Hence $Ess(v) = M_1 \cup \{x_{m+1}\}$, $Ess(v) \subseteq Ess(f)$ and $M_1 \subseteq Ess(v) \subseteq M$ with $M_1 \neq Ess(v) \neq M$ which contradicts $M_1 \in Max(M, f)$. Consequently, there is a vector $\overline{c} \in Z_k^s$ of constants from Z_k such that $M_1 \neg \subseteq Ess(g)$ where $f \geq_L^c g$. □

The next theorem is a slight improvement of Theorem 2.10.

Theorem 2.12. *Let $f \in P_k^n, \emptyset \neq M \notin Sep(f)$ and let $M_1 \in Max(M, f)$. Then there exists at least one essential conjugate of M_1 in f.*

Proof. Without loss of generality let us assume
$Ess(f) = \{x_1, \ldots, x_n\}$, $M_1 = \{x_1, \ldots, x_m\}$ and $M = \{x_1, \ldots, x_{m+p}\}$.

According to Lemma 2.11 there exists a vector $\overline{c} = (c_{m+p+1}, \ldots, c_n) \in Z_k^{n-p}$ such that $M_1 \neg \subseteq Ess(g)$, where $g = f(x_{m+p+1} = c_{m+p+1}, \ldots, x_n = c_n)$.

Since M_1 is separable in f there exists a vector $\overline{b} = (b_{m+p+1}, \ldots, b_n) \in Z_k^{n-p}$ such that $M_1 \in Sep(h)$, where $h = f(x_{m+p+1} = b_{m+p+1}, \ldots, x_n = b_n)$.

Let s, $1 \leqslant s \leqslant n-m-p$ be the minimal natural number for which $M_1 \in Sept(t)$, where

$$t = f(x_{m+p+1} = c_{m+p+1}, \ldots, x_{m+p+s-1} = c_{m+p+s-1})$$

and $M_1 \notin Ess(u)$, where $u = t(x_{m+p+s} = c_{m+p+s})$. The number s must exist because $M_1 \neg \subseteq Ess(g)$ and $M_1 \in Sep(h)$.

First, let $s < n-m-p$. Then $M_1 \in Sep(t)$ implies that there exist constants $d_{m+p+s}, \ldots, d_n \in Z_k$, such that $M_1 \in Sep(t_1)$ and $M_1 \neg \subseteq Ess(t_2)$ where

$$t_1 = t(x_{m+p+s} = d_{m+p+s}, \ldots, x_n = d_n)$$

and

$$t_2 = u(x_{m+p+s+1} = d_{m+p+s+1}, \ldots, x_n = d_n).$$

Pick

$$v = t(x_{m+p+s+1} = d_{m+p+s+1}, \ldots, x_n = d_n).$$

Clearly, $M_1 \in Sep(v)$ and $x_{m+p+s} \in Ess(v)$. If $(M \backslash M_1) \cap Ess(v) = \emptyset$ then we are clearly done. Next, suppose with no loss of generality that

$$L = \{x_1, \ldots, x_m, x_{m+1}, \ldots, x_{m+r}\} \subseteq Ess(v)$$

with $1 \leqslant r \leqslant p$. Then L must be inseparable in v and $M_1 \in Max(L, v)$. Now, Theorem 2.10 shows that x_{m+p+s} is an essential conjugate of M_1 in v and f.

Second, let as assume $s = n-m-p$. Then we can pick $z = f(x_{m+p+1} = c_{m+p+1}, \ldots, x_{n-1} = c_{n-1})$ with $M_1 \in Sep(z)$ and $M_1 \neg \subseteq Ess(z(x_n = c_n))$. The rest of the proof that x_n is an essential conjugate of M_1 in z and f is left to the reader. □

The improvement of Theorem 2.10 consists in the fact that we might choose the variable x_i before the choice of the subfunction g, $f \geq M_2 g$.

A natural question to ask is there an "universal" essential conjugate $x_i \in Ess(f) \backslash M$ for all maximal separable subsets of M, i.e. is it possible to choose the variable in Theorem 2.12 before the choice of the set $M_1 \in Max(M, f)$? The next example shows that the answer is negative.

Example 2.13. Let $k = 2$ and $f = x_1x_4x_5^0 \oplus x_2x_4^0x_6 \oplus x_3x_5x_6^0$. Clearly $M = \{x_1, x_2, x_3\} \notin Sep(f)$ and $Max(M, f) = \{\{x_1\}, \{x_2\}, \{x_3\}\}$. Also, it is easy to verify that $x_6 \notin Ess(f(x_2 = 0, x_3 = 0))$, $x_5 \notin Ess(f(x_1 = 0, x_3 = 0))$ and $x_4 \notin Ess(f(x_1 = 0, x_2 = 0))$. The essential

conjugates of the maximal separable subsets are: $\{x_4, x_5\}$ of $\{x_1\}, \{x_4, x_6\}$ of $\{x_2\}$, and $\{x_5, x_6\}$ of $\{x_3\}$.

Let us turn our attention to the following:

1. Each simple minor obtained by collapsing pairs of variables belonging to distinct maximal separable subsets of M depends on possible maximal number of essential variables. Thus we have $ess(f_{2\leftarrow 1}) = ess(f_{3\leftarrow 1}) = ess(f_{3\leftarrow 2}) = 5$. For instance, $f_{2\leftarrow 1} = x_1x_4x_5^0 \oplus x_1x_4^0x_6 \oplus x_3x_5x_6^0$.
2. The simple minors obtained by pairs of essential conjugates essentially depend on four variables, for instance, $f_{5\leftarrow 4} = x_2x_4^0x_6 \oplus x_3x_5x_6^0$.

Next, we turn our attention to relationship between essential arity gap and separable sets in functions.

Theorem 2.14. Let $f \in p_k^n$. If $gap(f) \geqslant 2$ then each non-empty set of essential variables is separable in f.

Proof. Let M be an arbitrary non-empty set of essential variables in f. We prove that $M \in sep(f)$ by considering cases. The theorem is given to be true if $n \leqslant 2$. Next we assume $n > 2$.

Case 1: $gap(f) = 2$, $n \geqslant 3$ and $k = 2$.

If $n = 3$ then Theorem 3.2 [13] implies that $f = x_3^{\alpha}(x_1^0x_2^1 \oplus x_1^1x_2^0) \oplus x_1^{\beta}x_2^{\beta}$ or $f = x_3^{\alpha}(x_1^0x_2^0 \oplus x_1^1x_2^1) \oplus x_3^{\neg(\alpha)}(x_1^0x_2^1 \oplus x_1^1x_2^0)$, where $\alpha, \beta \in \{0, 1\}$. Clearly, each set of essential variables in f is separable.

If $n = 4$ then according to Theorem 3.3 [13] we have $f = x_4^0.g(x_1, x_2, x_3) \oplus x_4^1.h(x_1, x_2, x_3)$, with $g = x_3^{\alpha}(x_1^0x_2^0 \oplus x_1^1x_2^1) \oplus x_3^{\neg(\alpha)}(x_1^0x_2^1 \oplus x_1^1x_2^0)$, and $h = x_3^{\neg(\alpha)}(x_1^0x_2^0 \oplus x_1^1x_2^1) \oplus x_3^{\alpha}(x_1^0x_2^1 \oplus x_1^1x_2^0)$, for some α, $\alpha \in \{0, 1\}$ (here $\neg(\alpha)$ means negation of α). Clearly, each set of essential variables in f is separable.

Let $n \geqslant 5$ From Theorem 3.4 in [13] it follows that

$$f = \bigoplus_{\alpha_1 \oplus \cdots \oplus \alpha_n = 1} x_1^{\alpha_1} \cdots x_n^{\alpha_n} \text{ or } f = \bigoplus_{\alpha_1 \oplus \cdots \oplus \alpha_n = 0} x_1^{\alpha_1} \cdots x_n^{\alpha_n}.$$

Thus we have $Ess(f) = X_n$. Suppose, with no loss of generality that $M = \{x_1, \cdots, x_m\}$, $m < n$ and

$$f = \bigoplus_{\alpha_1 \oplus \cdots \oplus \alpha_n = 1} x_1^{\alpha_1} \cdots x_n^{\alpha_n}.$$

Let $c_1, \cdots, c_n \in Z_k$ and $c_1 \oplus \cdots \oplus c_n = 1$. We can pick $g = f(x_{m+1} = c_{m+1}, \cdots, x_n = c_n)$ and $r = c_{m+1} \oplus \cdots \oplus c_n$. Assume without loss of generality that $r = 1$. Then we have

$$g = \bigoplus_{\alpha_1 \oplus \cdots \oplus \alpha_n = 1} x_1^{\alpha_1} \cdots x_m^{\alpha_m}.$$

It must be shown that $M = Ess(g)$. By symmetry, it is enough to show that $x_1 \in Ess(g)$. Let $c_2, \cdots, c_m \in Z_k$ with $c_2 \oplus \cdots \oplus c_m = 0$. Then we have

$$g(0, c_2, \cdots, c_m) = 1 \text{ and } g(1, c_2, \cdots, c_m) = 0,$$

which proves that $x_1 \in Ess(g)$.

Case 2: $gap(f) = 2, 2 < k < n$.

Theorem 2.1 [18] implies that f is a symetric function which essentially depends on all of its n variables. Theorem 4.1 [17] states that: If f is a symmetric function with non-trivial arity gap, then each set of essential variables in f is separable, which completes the proof of this case.

Case 3: $gap(f) = 2, n = 3$ and $k \geqslant 3$.

Lemma 5.1 [16] states that if $f \in G^3_{2,k}$ then $ess(f_{i \leftarrow j}) = 1$ for all $i, j \in \{1, 2, 3\}, i \neq j$, which shows that each subset of X_3 is separable in f.

Case 4: $gap(f) = 2, 4 \leqslant n \leqslant k$.

If f is a symmetric function then we are done because of Theorem 4.1 [17] and if f is not a symmetric function then according to Theorem 4.2 [16] there exist $n-2$ variables $y_{l_1}, \cdots, y_{l_{n-2}} \in X_n$ such that $f = h \oplus g$,where $Ess(h) = \{y_{l_1}, \cdots, y_{l_{n-2}}\}$ and $g \in G^n_{n,k}$. Moreover $g_{i \leftarrow j} = 0$ for all $1 \leqslant i, j \leqslant n$ with $i \neq j$. Now, the proof can be done as in **Case 6:**, for $p = 2$, given below.

Case 5: $gap(f) = n, 3 \leqslant n \leqslant k$.

From Theorem 3.1 [16] it follows that f is presented in the following form:

$$f = a_0 \left[\bigoplus_{\alpha \in Eq^n_k} \mathbf{x}^{\alpha} \right] \oplus \left[\bigoplus_{\beta \in Dis^n_k} a_r \mathbf{x}^{\beta} \right] \tag{1}$$

where
$Eq^s_k = \{\gamma \in Z^s_k \,|\, \exists i, j,\ 1 \leqslant i < j \leqslant s, \gamma_i = \gamma_j\}$,
$\gamma = \gamma_1 \cdots \gamma_s$, $\mathbf{x}^{\gamma} = x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_s^{\gamma_s}$ and $Dis^s_k = Z^s_k \backslash Eq^s_k$, for $s, s \geqslant 2$. Moreover, there exist at least two distinct numbers among $a_r \in Z_k$ for $r = 0, 1, \cdots, k^n - 1$.

It is easy to see that $Ess(f) = X_n$. We have to show that $M \in Sep(f)$. Without loss of generality let us assume that $M = \{x_1, \cdots, x_m\}, 1 \leqslant m \leqslant n$. If $m = n$ or $m = 1$ we are clearly done. Let $1 < m < n$ and let $r = \sum_{i=1}^{n} \beta_i k^{n-i}$ be a natural number such that $a_r \neq 0$. Then we have

$$g = f(x_{m+1} = \beta_{m+1}, \cdots, x_n = \beta_n) = a_0 \left[\bigoplus_{\delta \in Eq^m_k} \mathbf{x}^{\delta} \right] \oplus \left[\bigoplus_{\sigma \in Dis^m_k} a_j \mathbf{x}^{\sigma} \right],$$

where $j = \sum_{i=1}^{n} \sigma_i k^{n-i}$ with $\sigma_i = \beta_i$ for $i = m+1, \cdots, n$, and there are at least two distinct numbers among a_j. Clearly, g essentially depends on all of its variables, i.e. $Ess(g) = M$ and hence $M \in Sep(f)$.

Case 6 $gap(f) = p, 2 \leqslant p < n$, and $4 \leqslant n < k$.

According to Theorem 3.4 [16], there exist functions h and g, such that $f = h \oplus g$, where $g \in G^n_{n,k}$ and $ess(h) = n - p$. Without loss of generality, let us assume that $Ess(h) = \{x_1, \cdots, x_{n-p}\}$. Moreover, $g_{i \leftarrow j} = 0$ for all i and $j, 1 \leqslant j < i \leqslant n$.

Clearly, $Ess(f) = X_n$ and according to Theorem 3.1 [16] and the Eq. (1), given in Case 5, the function g can be represented as follows $g = u \oplus v$, where

$$u = \left[\bigoplus_{\beta \in Dis^n_k} a_r \mathbf{x}^{\beta} \right] \text{ and } v = a_0 \left[\bigoplus_{\alpha \in Eq^n_k} \mathbf{x}^{\alpha} \right]. \tag{2}$$

Let $x_i, x_j \in X_n,\ i > j$, be two arbitrary essential variables in g. Say $i = n$ and $j = n-1$, for simplicity. Then we have

$$u_{i \leftarrow j} = 0 \text{ and } v_{i \leftarrow j} = a_0 \left[\bigoplus_{\delta \in z^{n-2}_k} \widehat{\mathbf{x}}^{\delta} \right] = a_0. \tag{3}$$

Since $g_{i \leftarrow j} = 0$ for all i and $j, 1 \leqslant j < i \leqslant n$ we have $v_{i \leftarrow j} = a_0 = 0$ and hence $v = 0$, and $g = u$. Let M be a set of essential variables in f. Note that $M \in sep(g)$, according to Case 5 and if $M \cap Ess(h) = \emptyset$ then $M \in sep(f)$.

We have to prove that M is separable in f in each other case. We argue by induction on n-the number of essential variables in f and g.

Let $n = 4$. This is our basis of induction.

First, let $|M| = 2$ and $P = 2$. Clearly, if $M \subseteq Ess(h)$ then (2) and (3) show that $M \in Sep(f)$. Next, let us assume that $M = \{x_1, x_3\}$ and $Ess(h) = \{x_1, x_2\}$. Let $c_2, c_4 \in Z_k$ be two constants, such that $Ess(t_1) = \{x_1, x_3\}$, where $t_1 = g(x_2 = c_2, x_4 = c_4)$. Clearly, $x_3 \in Ess(f_1)$, where $f_1 = f(x_2 = c_2, x_4 = c_4)$. Let $h_1 = h(x_2 = c_2)$. If $x_1 \notin Ess(h_1)$ then $x_1 \in Ess(f_1)$ and obviously, $M \in sep(f)$. If $x_1 \in Ess(h_1)$ then $f_1 = h_1(x_1) \oplus t_1(x_1, x_3)$. According to (2) and (3) there is a constant $c_3 = Z_k$, such that $Ess(t_1(x_3 = c_3)) = \emptyset$. Hence $x_1 \in Ess(f_1(x_3 = c_3))$ and $M \in sep(f)$, again.

Second, let $|M| = 3$ and $P = 2$, and $Ess(h) = \{x_1, x_2\}$.

Let $x_1 \notin M$. Then there is a constant $c_1 \in Z_k$ such that $x_2 \in Ess(h_2)$, where $h_2 = h(x_1 = c_1)$. Thus, (2) implies that $\{x_2, x_3, x_4\} = Ess(f_2)$, where $f_2 = f(x_2 = c_1)$ and $M \in sep(f)$, again.

Let $x_4 \notin M$. Then there is a constant $d_4 \in Z_k$ such that $x_3 \in Ess(t_2)$, where $t_2 = g(x_4 = d_4)$. Clearly, $x_3 \in Ess(f_3)$, where $f_3 = f(x_4 = d_4)$. According to (2) and (3), we have $f_3 = (x_3 = d_4) = h(x_1, x_2) \oplus a_0$, which shows that $\{x_1, x_2, x_3\} = Ess(f_3)$ and hence $M \in sep(f)$.

One can argue similarly if $P = 3$ and $n = 4$.

Let us assume that for some natural number $l, l \geqslant 4$, if $n < l, 2 \leqslant p, l < k$ and $f \in G^n_{p,k}$, then each set of essential variables in f is separable.

Let us pick $n = l$. According to Lemma 2.4 there is a strongly essential variable x_i, $1 \leqslant i \leqslant l$ in g, and let $c_i \in Z_k$ be a constant such that $X_l \backslash \{x_i\} = Ess(g(x_i = c_i))$. Without loss of generality, let us assume that $i = l$ and $c_i = k - 1$. Using (2), it is easy to verify that

$$t_3 = g(x_l = k - 1) = \left[\bigoplus_{\beta \in Dis^{l-1}_{k-1}} b_r \tilde{\mathbf{x}}^{\beta} \right], \tag{4}$$

where the coefficients b_r linearly depend on $a_0, \ldots, a_{k^l-1}$ and $\tilde{\mathbf{x}}^{\beta} = x_1^{\beta_1} \ldots x_{l-1}^{\beta_{l-1}}$.

By $p \geqslant 2$ it follows that we may reorder the variables in h such that $Ess(h) = \{x_1, \ldots x_{l-p}\}$ with $l - p < l - 1$.

Then we can pick $f_4 = f(x_l = k - 1) = h \oplus t_3$. It must be shown that $Ess(f_4) = X_{l-1}$. Since $p \geqslant 2$ it follows that $N = Ess(f_4) \backslash Ess(h) \neq \emptyset$. Next, using (2) one can show that $N \in Sep(t_3)$ and $N \in Sep(f_4)$. According to (3) we have

$$N \cap Ess(f_4(x_i = k - 1)) - \emptyset,$$

for all $i = 1, \ldots, l - p$. Now, Lemma 2.5 implies $N \cup \{x_1, \ldots, x_{l-p}\} \in Sep(f_4)$. Hence $Ess(f_4) = x_{l-1}$. According to (4) it follows that $f_4 \in G^{l-1}_{p,k-1}$.

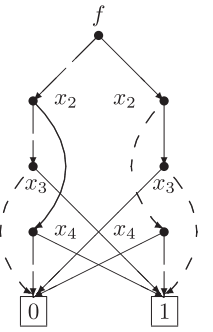
Therefore the inductive assumption may be applied to f_4, yielding $M \in Sep(f_4)$, and hence $M \in sep(f)$. □

3. Decision diagrams of functions

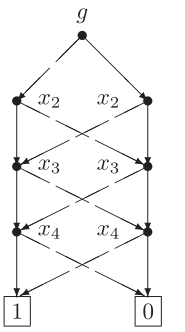
3.1 Ordered decision diagrams

Intuitively, it seems that a function f has the maximal complexity under the subfunction reduction if all its sets of essential variables are separable, because the variables from separable sets remain essential after assigning constants to other variables (see [15]). For example, when assigning Boolean constants to some variables of a Boolean function, then a natural complexity measure is the size of its Binary Decision Diagrams (BDDs), which also depend on the variable ordering (see [1]). Each path from the root (function node) to a terminal node (leaf) of BDD is called an *implementation* of f. The subfunction complexities $imp(f), sub(f)$ and $sep(f)$ of all implementations, subfunctions, and separable sets, obtained under all $n!$ variable orderings of n-ary Boolean functions for $n, n \leqslant 5$, are studied and calculated in [15].

Example 3.1. Let $f = x_1 x_4^0 \oplus x_1^0 x_3 \oplus x_2 x_3 \oplus x_2 x_4 (mod\, 2)$ and $g = x_1 \oplus x_2 \oplus x_3 \oplus x_4$ $(mod\, 2)$ be two Boolean functions. Figure 1 presents their BDDs under the natural



A) $f = x_1x_4^0 \oplus x_1^0x_3 \oplus x_2x_3 \oplus x_2x_4$



B) $g = x_1 \oplus x_2 \oplus x_3 \oplus x_4$

Figure 1.
Binary Desicion Diagrams.

variable ordering $x_1; x_2; x_3; x_4$. All sets of essential variables in g are separable, whereas the sets (x_1, x_2) and (x_3, x_4) are inseparable in f. Clearly, f has non-trivial essential arity gap and $f \in G_{2,2}^4$. Note that the implementations (longest paths) in Figure 1 A) consist of three edges, but in Figure 1 B) of four edges, which shows that the BDD of the function g is extremely complex with respect to the number of its subfunctions and separable sets, whereas the BDD of f is simpler.

3.2 Minor decision diagrams

Next we introduce a new graph-based presentation of the k-valued functions, namely by the minor decision diagrams.

The minor decomposition tree (MDT) of a function, consists of the node, labelled f – called the *function* node and nodes labelled with minor names, called the *internal (non-terminal)* nodes, and the rectangular nodes (leaves of the tree) called the *terminal* nodes. The terminal nodes are labelled with the same name of a function (atomic minor) from P_k^1 (according to Theorem 2.3). The terminal and non-terminal nodes in the MDT for a function f, essentially depending on n variables, are disposed into maximum $n-1$ layers of the tree. The i-th layer consists of names of all the distinct minors of order i, for $i = 1, \ldots, n-1$. Figure 2 presents the MDT of the function $f = x_1x_4^0 \oplus x_1^0x_3 \oplus x_2x_3 \oplus x_2x_4$, given in Example 3.1.

We introduce the *minor decision diagrams* (MDDs) for k-valued functions constructed by reducing their *minor decomposition trees* (MDTs). Let f be a k-valued function. The *minor decision diagram (MDD)* of f is obtained from the corresponding MDT by *reductions* of its nodes and edges applying of the following rules, starting from the MDT and continuing until neither rule can be applied:

Reduction rules

- If two edges have equivalent (as mappings) labels of their nodes they are merged.
- If two nodes have equivalent labels, they are merged.

Example 3.2. Let us build the MDDs of the functions from Example 3.1, namely $f = x_1x_4^0 \oplus x_1^0x_3 \oplus x_2x_3 \oplus x_2x_4 (mod\, 2)$ and $g = x_1 \oplus x_2 \oplus x_3 \oplus x_4 (mod\, 2)$ using the reduction rules and their MDT's.

Figure 3 A) shows the MDD of the function f, and Figure 3 B) presents the MDD of g. The identification minors of f and g are:

$f_{2\leftarrow 1} = f_{4\leftarrow 3} = x_1 \oplus x_3$,
$f_{2\leftarrow 1} \equiv f_{4\leftarrow 3} \equiv [f_{3\leftarrow 1}]_{4\leftarrow 2} \equiv [f_{4\leftarrow 2}]_{2\leftarrow 1}$
$f_{4\leftarrow 2} = x_1 x_2^0 \oplus x_1^0 x_3 \oplus x_2 x_3^0$.
$g_{i\leftarrow j} \equiv g_{2\leftarrow 1} = x_3 \oplus x_4$,
$[g_{2\leftarrow 1}]_{4\leftarrow 3} = 0$.

$f_{3\leftarrow 1} = x_1 x_4^0 \oplus x_1 x_2 \oplus x_2 x_4$,
$f_{3\leftarrow 1} \equiv f_{4\leftarrow 1} \equiv f_{3\leftarrow 2}$

for $1 \le j < i \le 4$ and,

Each edge $e = (v_1, v_2)$ in the diagram is supplied with a label $l(v_1, v_2)$, (written as bold in Figure 3A), which presents the number of the merged edges of the MDT, connecting the nodes v_1 and v_2 in MDT.

If two nodes in MDT are connected with unique edge then this edge is presented in MDD without label, for brevity. For example, such pairs are $(f, f_{4\leftarrow 2})$, $(f_{3\leftarrow 1}, f_{2\leftarrow 1})$ and $(f_{2\leftarrow 1}, 0)$.

The label of the edge $(f, f_{3\leftarrow 1})$ is **3** because there are three identification minors, namely $f_{3\leftarrow 1}$, $f_{4\leftarrow 1}$ and $f_{3\leftarrow 2}$ of f which are equivalent to $f_{3\leftarrow 1}$ (see Figure 2).

In a similar way we count the labels of the edges in Figure 3 B).

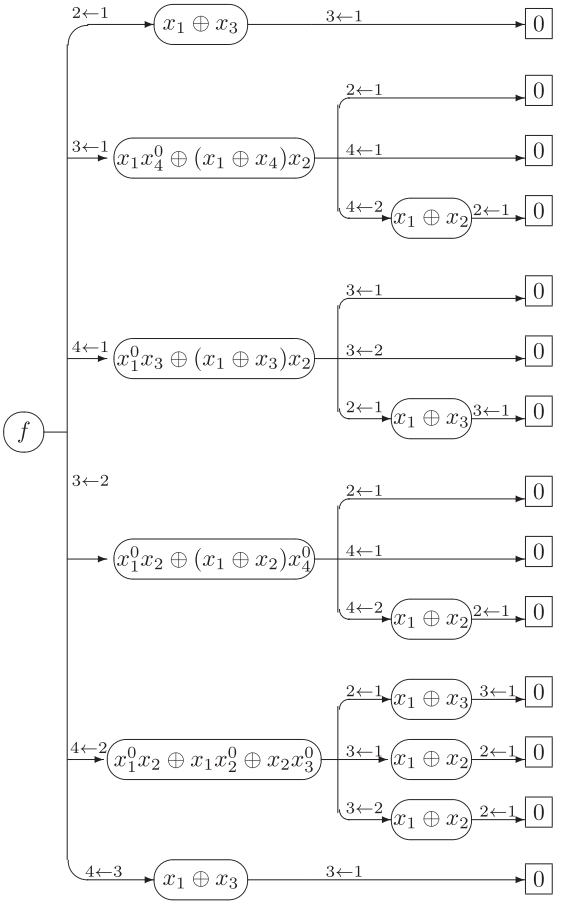


Figure 2.
MDT of $f = x_1 x_4^0 \oplus x_1^0 x_3 \oplus x_2 x_3 \oplus x_2 x_4 \ (mod\, 2)$.

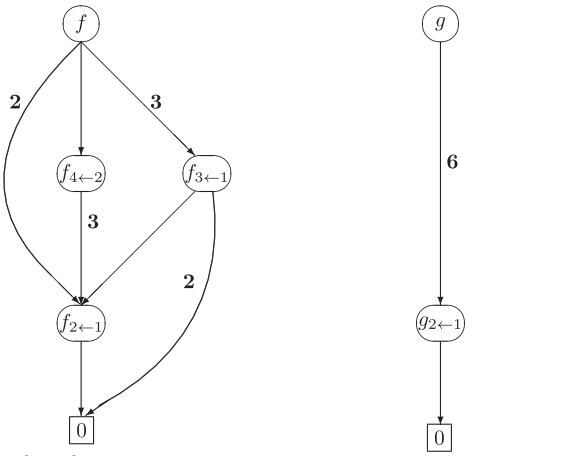


Figure 3.
Minor decision diagrams.

So, the MDD of f is an acyclic directed graph, with unique function node and according to Theorem 2.3, with unique terminal node. Clearly, the MDD and MDT are uniquely determined by the function f.

3.3 Complexity and equivalence relations with respect to minor reduction

Many of the problems in the applications of the k-valued logic are compounded because of the large number of the functions, namely k^{k^n}. Techniques which involve enumeration of functions can only be used if k and n are trivially small. A common way for extending the scope of such enumerative methods is to classify the functions into equivalence classes by some natural equivalence relation.

Let S_A denote the symmetric group of all permutations of the non-empty set A, and let S_m denote the group $S_{\{1,\ldots,m\}}$ for a natural number m, $m \geqslant 1$.

A *transformation* $\psi : P_k^n \to P_k^n$ is an n-tuple of k-valued functions $\psi = (g_1, \ldots, g_n)$, $g_1 \in P_k^n, i = 1, \ldots, n$, acting on a function $f = f(x_1 \ldots x_n) \in P_k^n$ as follows $\psi(f) = f(g_1, \ldots, g_n)$. Then the composition of two transformations ψ and $\phi = (h_1 \ldots h_n)$ is defined as follows

$$\psi\phi = (h_1(g_1 \ldots g_n), \ldots, h_n(g_1 \ldots g_n)).$$

The set of all transformations of P_k^n is the *universal monoid* Ω_k^n with unity - the identical transformation $\epsilon = (x_1 \ldots x_n)$. When taking only invertible transformations we obtain the *universal group* C_k^n which is isomorphic to the symmetric group $S_{z_k^n}$. The groups consisting of invertible transformations of P_k^n are called *transformation groups* (sometimes termed *permutation groups*).

Let $\simeq$, $\simeq \subseteq P_k^n \times P_k^n$ be an equivalence relation on the algebra P_k^n. Since P_k^n is a finite algebra of k-valued functions, the equivalence relation $\simeq$ makes a partition of the algebra in a finite number equivalence classes.

A mapping $\varphi : P_k^n \to P_k^n$ is called a *transformation preserving* $\simeq$ if $f \simeq \varphi(f)$ for all $f \in P_k^n$. Taking only invertible transformations which preserve $\simeq$, we get the group $G_\simeq$ of all transformations preserving $\simeq$. The *orbits* (also called $G_\simeq$*-types*) of this group are denoted by $P_1, \ldots, Pr$.

Our aim is to classify functions from P_k^n into equivalence classes by $\simeq$. Thus we have to calculate the number r of $G_\simeq$-types, to count the number of functions in different equivalence classes, i.e. compute the cardinalities of the sets $P_1, \ldots, Pr$ and to create a list of functions belonging to different $G_\simeq$-types.

Let $f \in P_k^n$ and let $nof(f)$ denote the normal form obtained by applying the reduction $\succ$ on f. According to Theorem 2.3, the normal form $nof(f)$ is unique and $nof(f) \in P_k^1$. Thus, our first natural equivalence is defined as follows:

Definition 3.3. Let f and g be two functions from P_k^n. We say that f and g are *nof*-equivalent (written $f \simeq_{nof} g$) if $nof(f) = nof(g)$.

The transformation group induced by *nof*-equivalence is denoted NF_k^n. The transformations in NF_k^n preserve $\simeq_{nof}$, i.e. $nof(g) = nof(\psi(g))$ for all $g \in P_k^n$ and $\psi \in NF_k^n$. Since the atomic minors (labels of terminal nodes in MDD) depend on at most one essential variable, it follows that the number of the orbits of NF_k^n is equal to $\left|P_k^1\right| = k^k$. These transformations involve permuting variables, only (see Theorem 3.9, below).

By analogy with the ordered decision diagrams [1,15], we define several equivalence relations in P_k^n, which allow us to classify the functions by the complexity of their MDDs.

The "scalability" of the diagram is an important measure of the computational complexity of the function. We are going to formalize this problem and establish a method for classification of functions by the minor complexities.

First, the number $mnr(f)$ of all the minors of a function f is a complexity measure, which can be used to evaluate the MDD of f. Namely, it counts the size (number of terminal and non-terminal nodes) of the MDD. M. Couceiro, E. Lehtonen and T. Waldhauser have studied similar evaluation, named "parametrized arity gap" in [5,6], which characterizes the sequential identification minors of a function.

Second, we are going to classify functions in finite algebras under the complexity measures which count the number of minors and the number of ways to obtain these minors.

Definition 3.4. Let $f \in P_k^n$ be a k-valued function. Its *cmr-complexity* $cmr(f)$ is defined as follows:

(i) $cmr(f) = 1$ if $ess(f) \leqslant 1$;

(ii) $cmr(f) = 2$ if $ess(f) = 2$;

(iii) $cmr(f) = \sum_{j<i,\, x_i, x_j \in Ess(f)} cmr(f_{i \leftarrow j})$ if $ess(f) \geqslant 3$.

The minors $f_{i \leftarrow j}$ with $i < j$ are excluded because $f_{i \leftarrow j} \equiv f_{j \leftarrow i}$. The minor complexity *cmr* can be inductively calculated using the MDDs of the functions as it is shown in Example 3.5, given below. We start to assign *cmr*-complexity equals to 1 for the terminal node, which is labeled by the minor of "0" of highest order according to (i) of Definition 3.4. Next, we inductively calculate the *cmr*-complexity of the minors of f with lower order, applying (ii) and (iii) of Definition 3.4.

Example 3.5. Let us count the *cmr*-complexity of the function f from Example 3.1 , using the identification minors of f obtained in Example 3.2 and MDD of f, given in Figure 3 A). There is two simple minors ($f_{2 \leftarrow 1}$ and $f_{4 \leftarrow 3}$) of order 2 and four simple minors of order 1. Thus we have $cmr(f_{2 \leftarrow 1}) = cmr(f_{4 \leftarrow 3}) = 2$, $cmr(f_{3 \leftarrow 1}) = cmr(f_{4 \leftarrow 1}) = cmr(f_{3 \leftarrow 2}) = 1 * 2 + 2 * 1 = 4$ and $cmr(f_{4 \leftarrow 2}) = 3 * 2 = 6$. According to Definition 3.4 we have $cmr(f) = 2 * 2 + 1 * 6 + 3 * 4 = 22$.

In a similar way from the MDD of g in Figure 3 B) we obtain $cmr(g) = 6 * 2 = 12$.

Definition 3.6. Let f and g be two functions with $ess(f) = n$, $n \geqslant 0$. We say that f and g are *cmr*-equivalent (written $f \simeq_{cmr} g$) iff:

(i) $n \leqslant 1 \Rightarrow ess(f) = ess(g)$;

(ii) $n \geqslant 2 \Rightarrow$ there exists a bijection $\sigma : Ess(f) \to Ess(g)$, such that $f_{i \leftarrow j} \simeq_{cmr} g_{r \leftarrow s}$, where $x_r = \sigma(x_i)$ and $x_s = \sigma(x_j)$, for all j, i, with $x_i, x_j \in Ess(f)$, $j < i$.

Let CM_k^n denote the transformation group preserving the equivalence $\simeq_{cmr}$, i.e. $\psi \in CM_k^n$ if and only if $\psi(f) = g \Rightarrow f \simeq_{cmr} g$.

The *nof*-equivalence is independent on the *cmr*-complexity of functions, defined by reduction via minors. For example, the functions $f = 0$ and $g = x_1^0 x_2 \oplus x_1 x_2 x_3^0 (mod\, 2)$ are *nof*-equivalent, but they are not *cmr*-equivalent.

Next we define another equivalence based on the number of minors (size of MDD) in a function.

Definition 3.7. Let f and g be two functions from P_k^n. We say that f and g are *mnr-equivalent* (written $f \simeq_{mnr} g$) if $mnr_m(f) = mnr_m(g)$ for all m, $0 \leqslant m \leqslant ess(f) - 1$.

Clearly, if $ess(f) \leqslant 1$ then $Mnr(f) = \emptyset$. Hence, if $ess(f) = ess(g) \leqslant 1$ then $f \simeq_{mnr} g$. MN_k^n denotes the transformation group which preserves the equivalence $\simeq_{mnr}$.

Note that $f \simeq_{mnr} g$ or $f \simeq_{nof} g$ do not imply $ess(f) = ess(g)$, which can be seen by the following functions: $f = x_1^0 x_2^1 (mod\, 3)$ and $g = x_1^0 x_2^1 x_3^2 (mod\, 3)$. Clearly, $f \simeq_{mnr} g$ and $f \simeq_{nof} g$, but $ess(f) = 2$, and $ess(g) = 3$.

Theorem 3.8.

(i) $f \simeq_{cmr} g \Rightarrow cmr(f) = cmr(g)$;

(ii) $f \simeq_{cmr} g \Rightarrow f \simeq_{mnr} g$.

Proof. We argue by induction on the number $n = ess(f)$.

If $ess(f) \leqslant 2$ (basis for induction) then we are clearly done. Assume that (i) and (ii) are satisfied when $n < s$ for some natural number s, $s > 2$. Let $n = s$ and $f \simeq_{cmr} g$. Then our inductive assumption implies

$$cmr(f) = \sum_{j<i} cmr(f_{i \leftarrow j}) = \sum_{u<v} cmr(g_{u \leftarrow v}) = cmr(g),$$

and $mnr_m(f_{i \leftarrow j}) = mnr_m(g_{u \leftarrow v})$, where $u = \pi(i)$ and $v = \pi(j)$ for some $\pi \in S_n$ and $m = 0, \ldots, n-1$. □

Thus, the complexity $cmr(f)$ is an invariant of the group CM_k^n, and the complexity $mnr(f)$ is an invariant of the group MN_k^n.

It is naturally to ask which groups among "traditional" transformation groups are subgroups of the groups NF_k^n or CM_k^n and which of these groups include NF_k^n, MN_k^n or CM_k^n as their subgroups.

Let $\sigma : Z_k \to Z_k$ be a mapping and let $\psi_\sigma : P_k^n \to P_k^n$ be a transformation of P_k^n generated by σ as follows $\psi_\sigma(f)(\overline{a}) = \sigma(f(\overline{a}))$ for all $\overline{a} \in Z_k^n$.

Theorem 3.9. *The transformation ψ_σ preserves $\simeq_{cmr}$ if and only if σ is a permutation of Z_k, $k > 2$.*

Proof. Let $\sigma : Z_k \to Z_k$.

First, let σ be a permutation of Z_k. Let $f \in P_k^n$ be an arbitrary function. If $ess(f) \leqslant 1$ then $ess(\psi_\sigma(f)) = ess(f)$ and we are clearly done. Let $ess(f) = ess(g) = n \geqslant 2$ and let i and j be two arbitrary natural numbers with $1 \leqslant j < i \leqslant n$. Then we have

$$[\psi_\sigma(f)]_{i \leftarrow j}(x_1, \ldots, x_n) = \sigma(f_{i \leftarrow j}(x_1, \ldots, x_n)).$$

Since σ is a permutation, it follows that $f_{i \leftarrow j} \simeq_{cmr} [\psi_\sigma(f)]_{i \leftarrow j}$ which shows that $f \simeq_{cmr} \psi_\sigma(f)$.

Second, let σ be not a permutation of Z_k. Hence, there exist two constants a_1 and a_2 from Z_k such that $a_1 \neq a_2$ and $\sigma(a_1) = \sigma(a_2)$. Let $\overline{b} = (b_1, \ldots, b_n) \in Z_k^n$, $n \geqslant 2$ be a vector of constants from Z_k. Then we define the following function from P_k^n:

$$f(x_1, \ldots, x_n) = \begin{cases} a_1 & \text{if} \quad x_i = b_i \text{ for } i = 1, \ldots, n \\ a_2 & \text{otherwise.} \end{cases}$$

Clearly, $Ess(f) = X_n$ and the range of f consists of two numbers a_1 and a_2. Then $\sigma(\{a_1, a_2\}) = \{\sigma(a_1)\}$, implies that $\psi_\sigma(f)(c_1, \ldots, c_n) = \sigma(a_1)$ for all $(c_1, \ldots, c_n) \in Z_k^n$. Hence, $Ess(\psi_\sigma(f)) = \emptyset$ which shows that $f \neg\simeq_{cmr} \psi_\sigma(f)$ and $\psi_\sigma \notin CM_k^n$. □

We deal with "natural" equivalence relations which involve variables of functions. Such relations induce permutations of the domain Z_k^n of the functions. These mappings form a transformation group whose number of equivalence classes can be determined. The restricted affine group (RAG) is defined as a subgroup of the symmetric group on the direct sum of the module Z_k^n of arguments of functions and the ring Z_k of their outputs. The group RAG permutes the direct sum $Z_k^n + Z_k$ under restrictions which preserve single-valuedness of all functions from P_k^n [8,10].

In the model of RAG an affine transformation ψ operates on the domain or space of inputs $\mathbf{x} = (x_1, \ldots, x_n)$ to produce the output $\mathbf{y} = \mathbf{xA} \oplus \mathbf{c}$, which might be used as an input in the function f. Its output $f(\mathbf{y})$ together with the function variables $x_1, \ldots, x_n$ are linearly combined by a range transformation which defines the image $g = \psi(f)$ of f as follows:

$$g(\mathbf{x}) = \psi(f)(\mathbf{x}) = f(\mathbf{y}) \oplus a_1 x_1 \oplus \ldots \oplus a_n x_n \oplus d = f(\mathbf{xA} \oplus \mathbf{c}) \oplus \mathbf{a}^{\mathbf{tx}} \oplus d, \tag{5}$$

where d and a_i for $i = 1, \ldots, n$ are constants from Z_k, $\mathbf{c} \in Z_k^n$, and $\mathbf{a} = (a_1, \ldots, a_n) \in Z_k^n$. Such a transformation belongs to RAG if $\mathbf{A}$ is a non-singular matrix.

We want to extract basic facts for several subgroups of RAG which are "neighbourhoods" or "relatives" of our transformation groups NF_k^n, CM_k^n and MN_k^n.

First, a classification occurs when permuting arguments of functions. If $\pi \in S_n$ then π acts on variables by: $\pi(x_1, \cdots, x_n) = (x_{\pi(1)}, \cdots, x_{\pi(n)})$. Each permutation generates a map on the domain Z_k^n.

For example, the permutation $\pi = (1, 2, 3)$ generates a permutation of the domain $\{0, 1, 2\}^3$ of the functions from P_3^3. Then we have $\pi : 001 \to 010 \to 100$ and in cyclic decimal notation this permutation can be written as $(1, 3, 9)$. The remaining elements of Z_3^3 are mapped according to the following cycles of π in decimal notation - $(2, 6, 18)(4, 12, 10)$ $(5, 15, 19)(7, 21, 11)(8, 24, 20)$ $(14, 16, 22)(17, 25, 23)$. Note that each permutation from S_n keeps fixed all k constant tuples from Z_k^n. In case of Z_3^3, these tuples $(0, 0, 0)$, $(1, 1, 1)$ and $(2, 2, 2)$ are presented by the decimal numbers 0, 13 and 26.

S_k^n denotes the transformation group induced by permuting of variables. Boolean functions of two variables are classified into twelve S_2^2-classes [8], as it is shown in Table 1. M. Harrison has determined the cycle index of the group S_2^n. Using Polya's counting theorem he has counted the number of equivalence classes under permuting arguments (see [8] and Table 3, below).

The subgroups of RAG, defined according to (5) which are "relatives" to the groups NF_k^n, CM_k^n and MN_k^n are determined as follows: RAG when $\mathbf{A}$-non-singular; CF_k^n when $\mathbf{A} = \mathbf{I}, \mathbf{a} = \mathbf{0}, \mathbf{c} = \mathbf{0}$; LF_k^n when $\mathbf{A} = \mathbf{I}, \mathbf{c} = \mathbf{0}, d = 0$; CA_k^n when $\mathbf{A} = \mathbf{I}, \mathbf{a} = \mathbf{0}, d = 0$; LG_k^n $\mathbf{c} = \mathbf{0}, \mathbf{a} = \mathbf{0}, d = 0$; S_k^n when $\mathbf{A} = \mathbf{P}, \mathbf{c} = \mathbf{0}, \mathbf{a} = \mathbf{0}, d = 0$, where $\mathbf{P}$ denotes a permutation matrix, $\mathbf{I}$ is the identity matrix, $\mathbf{b}$and$\mathbf{c}$ are n-dimensional vectors from Z_k^n and $d \in Z_k$.

It is naturally to ask which subgroups of RAG are subgroups of NF_k^n, CM_k^n and MN_k^n. Theorem 3.8 shows that CF_k^n and S_k^n are subgroups of MN_k^n. Clearly, $S_k^n \leqslant NF_k^n$.

Theorem 3.10.

(i) $CF_k^n \leq CM_k^n$;
(ii) $S_k^n \leq CM_k^n$;
(iii) $S_k^n \leq NF_k^n$;
(iv) $NF_k^n \neg\leq RAG$;
(v) $CM_k^n \neg\leq RAG$;
(vi) $LG_k^n \neg\leq MN_k^n$;
(vii) $CA_k^n \neg\leq MN_k^n$;
(viii) $CA_k^n \neg\leq NF_k^n$;
(ix) $LG_k^n \neg\leq NF_k^n$;
(x) $CM_k^n \neg\leq NF_k^n$;
(xi) $NF_k^n \neg\leq MN_k^n$.

Proof. (i) Follows from Theorem 3.9.(ii) and (iii) – Let $\pi \in S_n$ and let $\phi_\pi : P_k^n \to P_k^n$ be a transformation of P_k^n defined as follows $\phi_\pi(f)(a_1, \cdots, a_n) = f(a_{\pi(1)}, \cdots, a_{\pi(n)})$ for all $(a_1, \cdots, a_n) \in Z_k^n$. We have to prove that the transformation ϕ_π preserves the equivalence relations $\simeq_{cmr}, \simeq_{mnr}$ and $\simeq_{nof}$ for all $\pi \in S_n$. It suffices to show that ϕ_π preserves $\simeq_{cmr}$ and $\simeq_{nof}$. Let $f \in P_k^n$ be a function and let us assume $Ess(f) = X_n, n \geqslant 2$. It must be shown that $f \simeq_{cmr} g$ and $f \simeq_{nof} g$, where $g(a_1, \cdots, a_n) = f(\pi(a_1), \cdots, \pi(a_n))$ for all $(a_1, \cdots, a_n) \in Z_k^n$. Since π is a permutation, we have

$$f(a_1, \cdots, a_n) = g\left(\pi^{-1}(a_1), \cdots, \pi^{-1}(a_n)\right),$$

for all $(a_1, \cdots, a_n) \in Z_k^n$ which shows that $f \simeq_{cmr} g$ and hence, $f \simeq_{cmr} \phi_\pi(f)$. Since $nof(f) = f(x_i, \cdots, x_j)$ and $nof(f) = f(x_{\pi(i)}, \cdots, x_{\pi(i)})$, it follows $nof(g) \equiv nof(g)$ and $f \simeq_{nof} g$.(iv) and (v) – Let $f = x_1 \oplus x_2 \oplus x_3 (mod\ 3)$ and $g = x_1x_2 \oplus x_1x_3 \oplus x_2x_3 (mod\ 3)$. Then we have $f_{i \leftarrow j} = 2x_j \oplus x_m (mod\ 3)$ and $g_{i \leftarrow j} = 2x_jx_m \oplus x_jx_j (mod\ 3)$ where $\{i, j, m\} = \{1, 2, 3\}$. Clearly, $f_{i \leftarrow j_{\leftarrow m}} = g_{i \leftarrow j_{\leftarrow m}} = 0$, and hence $f \simeq_{cmr} g$ and $f \simeq_{nof} g$. One can show that there is no transformation $\psi \in RAG$, defined as in (5), for which $g = \psi(f)$. Consequently, $CM_k^n\neg \leqslant RAG, NF_k^n\neg \leqslant RAG$ and $MN_k^n\neg \leqslant RAG$.(vi), (vii), (viii), (ix) and (xi) – Let $f = x_1^0x_2^1 \oplus x_2^0x_3^1x_4^2 (mod\ 3)$ and $g = x_0^1x_2^1 \oplus x_2^0x_3^1x_4^1 (mod\ 3)$ be the functions from P_3^4. Let

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

Then clearly, $f(\mathbf{x}) = g(\mathbf{Ax})$ and hence, f and g belong to the same equivalence class under the transformation group LG_3^4. Let $\mathbf{c} = (0, 0, 0, 1)$. Then we have $f(\mathbf{x}) = g(\mathbf{x}.\mathbf{I} \oplus \mathbf{c})$, which shows that f and g belong to the same equivalence class under the transformation group CA_3^4. One can show that $f \simeq_{nof} g$. Example 3.5 shows that $f \neg \simeq_{mnr} g$. Consequently, $NF_k^n\neg \leqslant MN_k^n, LG_k^n\neg \leqslant MN_k^n$ and $CA_k^n\neg \leqslant MN_k^n$. Theorem 3.8 shows that $NF_k^n\neg \leqslant CM_k^n$, $LG_k^n\neg \leqslant CM_k^n$ and $CA_k^n\neg \leqslant CM_k^n$.(x) - Let us pick $f = x_1^0x_2^0 (mod\ 2)$ and $g = x_1x_2 (mod\ 2)$. Clearly, $f \simeq_{cmr} g$, but $nof(f) = x_1^0\neg \equiv x_1 = nof(g)$. □

So, Theorem 3.10 summarizes results which determine the positions of the groups NF_k^n, CM_k^n and MN_k^n, with respect to the subgroups of RAG. It is well-illustrated by Figure 4, in the case of Boolean functions.

4. Classification of Boolean functions by minor complexities

Table 2 shows the four classes in P_2^2 under the equivalence $\simeq_{cmr}$. The $\simeq_{cmr}$-classes are represented as union of several classes under the permuting arguments, according to Theorem 3.10 (ii), which can be observed in Table 1 and Table 2, given below.

The number of types under permuting arguments, is an upper bound of the number of equivalence classes induced by the relations $\simeq_{nof}, \simeq_{cmr}$ and $\simeq_{mnr}$ (see Figure 4).

Table 1.
The twelve classes in P_2^2 under the permutating of variables.

[0],	$[x_1^0x_2^0]$,	$[x_1^0x_{2,}\ x_1x_2^0]$,	$[x_{1,}\ x_2]$,
$[x_1 \oplus x_2]$,	$[x_1 \oplus x_2^0]$,	$[x_1^0 \oplus x_1x_{2,}\ x_2^0 \oplus x_1x_2]$,	$[x_1 \oplus x_1^0x_2]$,
$[x_1^0 \oplus x_1x_2^0]$,	$[x_1x_2]$,	$[x_1^0,\ x_2^0]$,	[1].

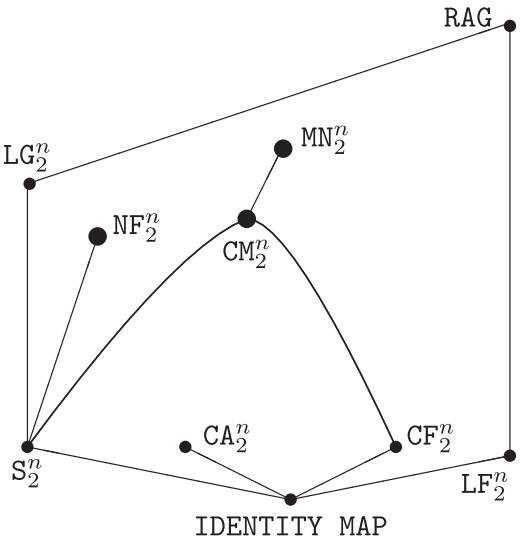


Figure 4. Transformation groups in P_2^n.

In Table 3 the columns named SB_2^n and SP_2^n are calculated in [15] and they present the number of classes under the complexities, determined by the number of subfunctions and separable sets in the functions. It is surprising that for $n \leqslant 3$ these columns are same as the columns CM_2^n and MN_2^n, i.e. the number of classes are the same, but these classes are very different as sets of functions, determined by these complexities.

Figure 4 presents the subgroups of RAG and transformation groups whose invariants are subfunction, and minor complexities of Boolean functions of n-variables. According to Theorem 3.10 the group CM_2^n has three subgroups from RAG, namely: S_2^n - the group of permuting arguments, trivial group consisting of the identity map, and $C\bar{F}_2^n$- the group of

[0, 1]	$[x_1^0x_2, x_1x_2^0, x_1 \oplus x_2, x_1 \oplus x_2^0, x_1^0 \oplus x_1x_2, x_2^0 \oplus x_1x_2]$,
$[x_1, x_2, x_1^0, x_2^0]$,	$[x_1x_2, x_1 \oplus x_1^0x_2, x_1^0 \oplus x_1x_2^0, x_1^0x_2^0]$.

Table 2. The four classes in P_2^2 under the *cmr*-complexity.

N	S_2^n	CM_2^n	MN_2^n	SB_2^n	SP_2^n
1	4	2	2	2	2
2	12	4	3	4	3
3	80	11	5	11	5
4	3984	*	*	74	11
5	37 333 248	*	*	*	38

Table 3. Number of equivalence classes in P_2^n under transformation groups.

complementing outputs. The groups NF_2^n and MN_2^n are not subgroups of any subgroup of RAG.

Next, we turn our attention on classifying the functions with respect to their *cmr*-complexity. This classification is based on the exhaustive Algorithm 1, given below.

Table 4 presents a complete classification of the Boolean functions of tree variables by the minor complexities *cmr* and *mnr*. If we agree to regard each 2^3-tuple as a binary number then the last column presents the vectors of values of all ternary Boolean functions in their table representation with the natural numbers from the set $\{0, \cdots, 127\}$. According to Theorem 3.9, if a natural number z, $0 \leqslant z \leqslant 255$, presents a function f which belongs to a *cmr*-class then the function $\hat{f}$ presented by $255 - z$ belongs to the same class. Thus the catalogue contents the numbers $\leqslant 127$, only (see the last column in Table 4). These numbers represent the functions which preserve zero, i.e. the functions f for which $f(0,0,0) = 0$. This classification shows that there are eleven equivalence classes under $\simeq_{cmr}$ and five classes under $\simeq_{mnr}$.

Theorem 3.8 shows that each *mnr*-class is a disjoint union of several *cmr*-classes. Thus the first *mnr*-class consists of all the functions which belong to the first and the second *cmr*-class (see fifth column in Table 4). The second *mnr*-class is equal to the third *cmr*-class. The fourth and the fifth *mnr*-classes are unions of three *cmr*-class, namely: sixth, seventh, and eight, and ninth, tenth, and eleventh, respectively.

The main data structure which describes the nodes in the MDD of f is represented by a record declared as follows:

```
type minor=record
  ess: 1..n;
  val: 0..k^(n-1) − 1;
end;
```

The first field, named `ess` presents the number of essential variables in the minor (located on the corresponding node) and the second field `val` is a natural number whose k-ary representation is the last column `B` of the truth table (of size $k^n \times (n+1)$) of the minor.

Table 4 presents classification of ternary Boolean functions under the equivalences $\simeq_{cmr}$ and $\simeq_{mnr}$, including the catalogue of the equivalence classes (last column). Let us choose a natural number belonging to the seventh column of Table 4, say 24. It belongs to the row numbered 6. The binary representation of 24 is 00011000, because $24 = 1*2^4 + 1*2^3$. Hence, the function f corresponding to 24 is evaluated by 1 on the fourth and fifth miniterms, namely $x_1^0 x_2 x_3$ and $x_1 x_2^0 x_3^0$. Consequently, $f = x_1^0 x_2 x_3 \oplus x_1 x_2^0 x_3^0 (mod\, 2)$. Then we have $f_{2\leftarrow 1} = f_{3\leftarrow 1} = 0$ and $f_{3\leftarrow 2} = x_1^0 x_2 \oplus x_1 x_2^0 (mod\, 2)$. Clearly, $cmr(f) = 4$, which is written in the third cell of the sixth row. The MDD of f is shown in the second cell. The *cmr*-equivalence class containing f consists of 18 functions, according to the fourth cell of the sixth row and the *mnr*-equivalence class of f contains 108 functions (see whole fifth column of the table). The function $x_1 x_2 x_3^0 (mod\, 2)$ is representative for this class (sixth cell). The numerical list of the functions from this equivalence class is given in the last seventh cell of

Table 4. The record of the function f is presented as follows `f.ess=3` and `f.val=24`, where $k = 2$ and `B=` 00011000.

Algorithm 1 – Counting cmr(f)

```
1:  type minor=record
      ess: 1..n;
      val: 0..k^(k^n-1) − 1;
      end;
2:  var f:minor;
      cmr:integer;
3:  function GETMINOR(g:minor; i,j:integer): minor;   ▷ Getting minor
    var A,H: array[1..k^N, 1..N] of integer (mod k);
      B,L; array[1..k^N] of integer (mod k);
      h: minor;
5:    n:=g.ess;
6:    Create truth table AB of g;
7:    Create truth table HL of h:=g_{i←j};   ▷From the table AB
8:    Calculate - h.ess and h.val   ▷From the table HL
9:    GetMinor:=h;
10: end function;
11: function COMPLEXITY g:minor:integer;   ▷Counting complexity
12: n:=g.ess;
13: if n>2
14:   for j, 1⩽j⩽n-1 do
15:     for i, j+1⩽i⩽n do
16:     h:=GetMinor(g,i,j);
17:     Complexity:=Complexity+Complexity(h);
18:     end for
19:   end for
20: else   ▷Basis of recursion
21:   if n=2 then
22:     Complexity:=2
23:   else
24:     Complexity:=1
25:   end if
26: end if
27: end function
28: BEGIN   ▷Main program
29:   Read (k; f.ess; f.val);
30:   cmr:=Complexity(f);
31:   Print cmr;
32: END.
```

cmr-class		cmr	Functions per class		mnr	Repres. function	Catalogue
	MDD						
1	const	1	2			0	0
2	var	1	6	8	0	x_1	15,51,85
3		2	18	18	1	$x_1x_2^0$	10,12,34,48,60,68, 80,90,102
4		2	12			x_1x_2	3,5,17, 63,95,119
5		3	8	20	1	$x_1 \oplus x_2$ $\oplus x_3$	43,77, 105,113
6		4	18			$x_1x_2x_3^0$	2,4,8, 16,24,32, 36,64,66
7		5	36			$x_1x_2^0x_3$ $\oplus x_1x_2x_3^0$	6,18,20,26,28,38, 40,44,52,56,70,72, 74,82,88,96, 98,100
8		6	54	108	2	$x_1x_2^0$ $\oplus x_1x_2x_3^0$	14,22,30,42,46,50, 54,58,62,76,78,84, 86,92,94,104,106, 108,110,112,114,116, 118,120,122,124,126
9		4	50			x_1x_2 $\oplus x_1x_2^0x_3$	7,11,13,19,21,23, 31,35,41,47,49,55, 59,69,73,79,81,87, 93,97,107,109,115, 117,121
10		5	36			$x_1x_2x_3$ $\oplus x_1x_2^0x_3^0$	9,27,29,33,39,45, 53,57,65,71,75,83, 89,99,101, 111,123,125
11		6	16	102	2	$x_1x_2x_3$	1,25,37, 61,67,91, 103,127

Table 4.
Minor classification of ternary Boolean functions.

5. Conclusion

The transformation groups whose invariants are the minor complexities have only three subgroups among the groups in RAG, namely trivial group (identity map), S_k^n and CF_k^n, whereas the groups whose invariants are the subfunction complexities have three subgroups more (see [15]). One of motivations to study the group NF_k^n is that the reductions are inexpensive and the number of classes is much smaller than the number of classes under the subgroups of RAG, because the order of NF_k^n is so large. As mentioned, the number of equivalence classes under NF_k^n equals to k^k. Hence, the order of NF_k^n is equal to $k^{k^n}/k^k = k^{k^n-1}$.

The most complex functions with respect to separable sets [15] are grouped in the largest equivalence class. J. Denev and I. Gyudzhenov in [7] proved that for almost all the k-valued functions all the sets of essential variables are separable. Similar results can not be proved for the minor complexities. For example, in P_2^3 the most complex functions belong to the class numbered as 11 (see Table 4), which consists of 16 functions. This class is not so large. It presents 1/16 of the all 256 ternary Boolean functions.

References

[1] R.E. Bryant, Graph-based algorithms for boolean function manipulation, IEEE Trans. Comput. C-35 (8) (1986) 677–691.

[2] K. Chimev, Separable sets of arguments of functions, MTA SzTAKI Tanulmanyok 80 (1986) 1–173.

[3] K. Chimev, S. Shtrakov, I. Giudjenov, M. Aslanski, Separable and dominating sets of variables for the functions, in: Math. and Educ. in Math., vol. 16, Union of Bulgarian Mathematicians, BAS, 1987, pp. 61–71.

[4] B. Choi, Advancing from two to four valued logic circuits, in: Proc. IEEE Intern. Conf. on Industrial Technology, IEEE, 2013.

[5] M. Couceiro, E. Lehtonen, T. Waldhauser, GAP vs. PAG, Multiple-Valued Logic 42 (2012) 268–273.

[6] M. Couceiro, E. Lehtonen, T. Waldhauser, Parametrized arity gap, Order 30 (2013) 557–572.

[7] J. Denev, I. Gyudzhenov, On separable subsets of arguments of functions from p_k ,MTA SzTAKI Tanulmanyok 26 (147) (1984) 47–50.

[8] M. Harrison, Counting theorems and their applications to classification of switching functions, in: A. Mikhopadhyay (Ed.), Recent Developments in Switching Theory, Academic Press, NY, 1971, pp. 85–120.

[9] J. Klop, R. de Vrijer, Term rewriting systems, Cambridge University Press, 2003.

[10] R.J. Lechner, Harmonic analysis of switching functions, in: A. Mikhopadhyay (Ed.), Recent Developments in Switching Theory, Academic Press, NY, 1971, pp. 121–228.

[11] A. Salomaa, On essential variables of functions, especially in the algebra of logic, Annales Academia Scientiarum Fennicae Ser. A 333 (1963) 1–11.

[12] S. Shtrakov, On some transformation groups in k-valued logic, General Algebra Appl. 20 (1993) 225–237.

[13] S. Shtrakov, Essential arity gap of boolean functions, Serdica J. Comput. 2 (3) (2008) 249–266.

[14] S. Shtrakov, Essential variables and positions in terms, Algebra Universalis 61 (3–4) (2009) 381–397.

[15] S. Shtrakov, I. Damyanov, On the computational complexity of finite operations, Int. J. Found. Comput. Sci. 27 (01) (2016) 15–38.

[16] S. Shtrakov, J. Koppitz, On finite functions with non-trivial arity gap, J. Discuss. Math. General Algebra Appl. 30 (2010) 217–245.

[17] S. Shtrakov, J. Koppitz, Finite symmetric functions with non-trivial arity gap, Serdica J. Comput. 6 (4) (2012) 419–436.

[18] R. Willard, Essential arities of term operations in finite algebras, Discrete Math. 149 (1996) 239–259.

Corresponding author
Slavcho Shtrakov can be contacted at: Shtrakovshtrakov@gmail.com