MINOR COMPLEXITIES OF FINITE OPERATIONS

SLAVCHO SHTRAKOV

Abstract. In this paper we present a new class of complexity measures, induced by a new data structure for representing \( k \)-valued functions (operations), called minor decision diagram. The results are presented in terms of Multi-Valued Logic circuits (MVL-circuits), ordered decision diagrams, formulas and minor decomposition trees. When assigning values to some variables in a function \( f \) the resulting function is a subfunction of \( f \), and when identifying some variables the resulting function is a minor of \( f \). A set \( M \) of essential variables in \( f \) is separable if there is a subfunction of \( f \), whose set of essential variables is \( M \). The essential arity gap \( \text{gap}(f) \) of the function \( f \) is the minimum number of essential variables in \( f \) which become fictive when identifying distinct essential variables in \( f \). We prove that, if a function \( f \) has non-trivial arity gap \( \text{gap}(f) \geq 2 \), then all sets of essential variables in \( f \) are separable. We define equivalence relations which classify the functions of \( k \)-valued logic into classes with the same minor complexities. These relations induce transformation groups which are compared with the subgroups of the restricted affine group (RAG) and the groups determined by the equivalence relations with respect to the subfunctions, implementations and separable sets in functions. These methods provide a detailed classification of \( n \)-ary \( k \)-valued functions for small values of \( n \) and \( k \).

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 directed acyclic graph in which all vertices except input vertices carry the labels of gates. When realizing functions are taken from the \( k \)-valued logic 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\(^2\)L (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 designs. For instance, MVL-circuits are well-applied in memory technology as flash memory, dynamic RAM, and in algebraic circuits [8].

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

2010 Mathematics Subject Classification. Primary: 94C10; Secondary: 06E30, 68Q25, 68Q15.
Key words and phrases. identification minor, minor decision diagram, minor complexity.

arXiv:1611.05633v1 [cs.DM] 17 Nov 2016
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 finite 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, implementations, 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 $P^n_k$ of $k$-valued functions determines a transformation group whose orbits are the equivalence classes. Using the lattice of Restricted Affine Groups (RAG) in [13] we have obtained upper bounds of different combinatorial parameters of several natural equivalences in $P^n_k$ 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 minor decomposition trees (MDTs), minor decision diagrams (MDDs), and minor complexities of discrete functions. Equivalence relations and transformation groups concerning the number of minors in functions is the topic of Section 4. Classification of Boolean (switching) functions of a "small" number of their essential variables is presented in Section 5. The Appendix examines an algorithm for counting the minor complexities of functions and provides a full classification of all ternary Boolean functions by these complexities.

2. Subfunctions and minors of functions

A discrete function $f$ is defined as a mapping: $f : A \rightarrow B$ 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 $k \geq 2$. Let $Z_k$ denote the set $Z_k = \{0, 1, \ldots, k-1\}$. The operations addition "⊕" and product "" modulo $k$ constitute $Z_k$ as a ring. An $n$-ary $k$-valued function (operation) on $Z_k$ is a mapping $f : Z_k^n \rightarrow Z_k$ for some natural number $n$, called the arity of $f$. $P^n_k$ denotes the set of all $n$-ary $k$-valued functions and $P_k = \bigcup_{n=1}^{\infty} P^n_k$ is called the algebra of $k$-valued logic. It is well-known fact that there are $k^{k^n}$ functions in $P^n_k$.

For simplicity, let us assume that throughout the paper we shall consider $k$-valued functions, only.

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

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

The ring-sum expansion (RSE) of a function $f$ is the sum modulo $k$ of a constant and products of variables $x_i$ or $x_i^\alpha$, (for $\alpha, \beta \in Z_k$) of $f$. For example, $1 \oplus x_1 x_2^2$ is a RSE of the function $f$ in the algebra $P^2_3$, 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 \( \text{Ess}(f) \) and \( \text{ess}(f) = |\text{Ess}(f)| \). 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 \( \succeq \). \( \text{Sub}(f) = \{g \mid f \succeq g\} \) is the set of all subfunctions of \( f \) and \( \text{sub}(f) = |\text{Sub}(f)| \).

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

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

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 \( \text{Ess}(f(x_i = c_i)) = \text{Ess}(f) \setminus \{x_i\} \). The set of all strongly essential variables in \( f \) is denoted \( \text{SEss}(f) \).

The following lemma is independently proved by K. Chimev \[3\] and A. Salomaa \[12\] in different variations.

**Lemma 2.1.** \[3\] Let \( f \) be a function. If \( \text{ess}(f) \geq 2 \) then \( f \) has at least two strongly essential variables, i.e. \( |\text{SEss}(f)| \geq 2 \).

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 \( h = f_{i \leftrightarrow j} \) and \( h \) is called a simple identification minor of \( f \) \[4\]. Clearly, \( \text{ess}(f_{i \leftrightarrow j}) < \text{ess}(f) \), because \( x_i \notin \text{Ess}(f_{i \leftrightarrow j}) \), but it has to be essential in \( f \). When \( h \) is a simple identification minor of \( f \) we write \( f \triangleright h \). The transitive closure of \( \triangleright \) is denoted \( \trianglerighteq \). \( \text{Mnr}(f) = \{h \mid f \trianglerighteq h\} \) is the set of all distinct minors of \( f \) and \( \text{mnr}(f) = |\text{Mnr}(f)| \). Let \( h, f \trianglerighteq h \) be an identification minor of \( f \). The natural number \( r = \text{ess}(f) - \text{ess}(h), r \geq 1 \) 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 \( \text{Mnr}_m(f) \) denote the set \( \text{Mnr}_m(f) = \{g \mid g \in \text{Mnr}(f) \& \text{ess}(g) = m \} \) and let \( \text{mnr}_m(f) = |\text{Mnr}_m(f)| \), for all \( m, m \leq n - 1 \).

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

\[
gap(f) = \text{ess}(f) - \max_{h \in \text{Mnr}(f)} \text{ess}(h).
\]
Let $2 \leq p \leq m$. 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.

$$G_{p,k}^m = \{f \in P_k^n \mid \text{ess}(f) = m \& \text{gap}(f) = p\}.$$  

We say that the arity gap of $f$ is non-trivial if $\text{gap}(f) \geq 2$. 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. For more results about the arity gap we refer \[3, 4, 12, 14, 16, 17, 20\].

**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.

![Figure 1. Simple subfunction and simple minor of functions](image)

Figure 1 illustrates the difference between these two models of computing. The reduction to simple subfunction $f(x_1 = c_1)$ is presented in the top of the figure, and computing the identification minor $f_{2\leftarrow 1}$ is shown in the bottom of the figure. The resulting functions of the both computations have the same domain - $Z_k^{n-1}$ and the same range - $Z_k$ (see the right side of Figure 1). In practice, both resulting functions can be seen as partial mappings of $Z_k^n$ into $Z_k$. The subfunction $f(x_1 = c_1)$ is obtained by fixing the first input ($x_1$), whereas the minor $f_{2\leftarrow 1}$ has two identical inputs (the left-lower corner of the figure). Another 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, 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 $f$.

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. [10, 18] 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 [10]. An ARS in $P_k^n$ is a structure $W = \langle P_k^n, \{\rightarrow_i\}_{i \in I} \rangle$, where $\{\rightarrow_i\}_{i \in I}$
is a family of binary relations on $P^n_k$, called reductions or rewrite relations. For a reduction $\rightarrow_i$, the transitive and reflexive closure is denoted $\rightarrow_i$. A function $g \in P^n_k$ is a normal form if there is no $h \in P^n_k$ such that $g \rightarrow_i h$. 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 $t, r \in P^n_k$ are normal forms obtained by applying the reductions $\rightarrow_i$, on a function $f \in P^n_k$ 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: $W = \langle P^n_k, \{>, \triangleright\} \rangle$. Next, we show that $\triangleright$ completes the reduction process with unique normal form, whereas $>$ 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 $w \in P^n_k$ such that $r \rightarrow w$ and $v \rightarrow w$.

**Theorem 2.3.**

(i) The reduction $\triangleright$ is UN;

(ii) The reduction $>$ is not WCR, but it is SN.

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

(WCR) Let $f$ be a function and $f \triangleright g$, and $f \triangleright h$. Let $t$ and $r$ be normal forms such that $g \triangleright t$ and $h \triangleright r$. Note that each normal form is a resulting minor obtained by collapsing all the essential variables in $f$. Hence, $\text{ess}(t) \leq 1$ and $\text{ess}(r) \leq 1$. Then we have $t = f(x_j, \ldots, x_j)$, for some $x_j \in \text{Ess}(f)$ and $r = f(x_i, \ldots, x_i)$, for some $x_i \in \text{Ess}(f)$, 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. [10]), which states that WCR & SN $\Rightarrow$ UN. □

(ii) is clear. □

Below, we also establish a key theorem which states that the functions with simplest minor complexity (with non-trivial arity gap) have extremely complex representations with respect to the number of their subfunctions, and separable sets.

**Lemma 2.4.** Let $N \in \text{Sep}(f)$. If there exist $m$ constants $c_1, \ldots, c_m \in Z_k$ such that $N \cap \text{Ess}(g_i) = \emptyset$ where $g_i = f(x_i = c_i)$ for $1 \leq i \leq m$ then $M \cup N \in \text{Sep}(f)$ 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 $N = \{x_{m+1}, \ldots, x_s\}$, $m < s \leq n$. Since $N \in \text{Sep}(f)$, there exists a vector of constants, say $\vec{od} = (d_{s+1}, \ldots, d_n) \in Z_k^{n-s}$ such that $N \subseteq \text{Ess}(g)$, 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 $x_s \in \text{Ess}(h)$ where
\[ h = g(x_{m+1} = d_{m+1}, \ldots, x_{s-1} = d_{s-1}). \]

We have to prove that $M \subseteq \text{Ess}(h)$. 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 $N \cap \text{Ess}(t) = \emptyset$ where $t = f(x_1 = c_1)$. Our supposition shows that $h = h(x_1 = c_1)$ and hence, $N \cap \text{Ess}(h) = \emptyset$, i.e. $x_s \notin \text{Ess}(h)$, which is a contradiction. Consequently, $M = \text{Ess}(h)$. Then $g \geq h$ implies $M \subseteq \text{Ess}(g)$ and hence, $M \cup N = \text{Ess}(g)$ which establishes that $M \cup N \in \text{Sep}(f)$.

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

Corollary 2.5. 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 $\{x_i, x_j\} \in \text{Sep}(f)$.

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

Theorem 2.6. Let $f \in P^n_k$. If $\text{gap}(f) \geq 2$ then all non-empty sets of essential variables are separable in $f$.

Proof. Without loss of generality, let us assume that $\text{Ess}(f) = \{x_1, \ldots, x_n\}$. Let $M$ be an arbitrary non-empty set of essential variables in $f$. We shall prove that $M \in \text{Sep}(f)$ by considering cases. The theorem is given to be true if $n \leq 2$. Next, we assume $n > 2$.

Case 1: $\text{gap}(f) = 2$, $n \geq 3$ and $k = 2$.

If $n \leq 4$ then we are done because of Theorem 3.2 and Theorem 3.3 in [14].

Let $n \geq 5$. From Theorem 3.4 in [14] it follows that
\[ f = \bigoplus_{\alpha_1 \oplus \cdots \oplus \alpha_n = 1} x_1^{\alpha_1} \cdots x_n^{\alpha_n} \quad \text{or} \quad f = \bigoplus_{\alpha_1 \oplus \cdots \oplus \alpha_n = 0} x_1^{\alpha_1} \cdots x_n^{\alpha_n}. \]

Clearly, $\text{Ess}(f) = X_n$. Suppose, with no loss of generality that $M = \{x_1, \ldots, x_m\}$, $m < n$ and
\[ f = \bigoplus_{\alpha_1 \oplus \cdots \oplus \alpha_n = 1} x_1^{\alpha_1} \cdots x_n^{\alpha_n}. \]

Let $c_1, \ldots, c_n \in Z_k$ and $c_1 \oplus \cdots \oplus c_n = 1$. We can pick $t = f(x_{m+1} = c_{m+1}, \ldots, x_n = c_n)$ and $r = c_{m+1} \oplus \cdots \oplus c_n$. Assume, without loss of generality, that $r = 1$. Then we have
\[ t = \bigoplus_{\alpha_1 \oplus \cdots \oplus \alpha_n = 0} x_1^{\alpha_1} \cdots x_m^{\alpha_m}. \]

It must be shown that $M = \text{Ess}(t)$. By symmetry, it suffices to show that $x_1 \in \text{Ess}(t)$. Let $c_2, \ldots, c_m \in Z_k$ be $m - 1$ values, such that $c_2 \oplus \cdots \oplus c_m = 0$. Then we have
\[ t(0, c_2, \ldots, c_m) = 1 \quad \text{and} \quad t(1, c_2, \ldots, c_m) = 0, \]
which establishes that $x_1 \in \text{Ess}(t)$.

Case 2: $\text{gap}(f) = 2$, $2 < k < n$. 

Theorem 2.1 from [20] implies that $Ess(f) = X_n$ and $f$ is a symmetric function. According to Theorem 4.1 [17], it follows that all the non-empty subsets of $X$ are separable in $f$.

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

Using Theorem 5.1 and Proposition 5.1 in [16], one can show that $Ess(f) = X_n$ and $M \subseteq X_n \Rightarrow M \in Sep(f)$, analogous to Case 1.

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

If $f$ is a symmetric function then we are done because of Theorem 4.1 in [17] and if $f$ is not a symmetric function then the proof is done as a part of Case 6, given below.

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

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

$$f = a_0 \left[ \bigoplus_{\alpha \in Eq_n^k} x^\alpha \right] + \left[ \bigoplus_{\beta \in Dis_n^k} a_\beta x^\beta \right],$$

where $Eq_n^k = \{ \gamma \in Z_n \mid \exists i, j, 1 \leq i < j \leq s, \gamma_i = \gamma_j \}$, $\gamma = \gamma_1 \ldots \gamma_s$, $x^\gamma = x_1^{\gamma_1} x_2^{\gamma_2} \ldots x_k^{\gamma_s}$, and $Dis_n^k = Z_n^s \setminus Eq_n^k$, for $s, s \geq 2$. Moreover, there exist at least two distinct coefficients among $a_r \in Z_k$ for $r \in \{0, 1, \ldots, k^n - 1\}$.

It is easy to see that $Ess(f) = X_n$. Let $M$ be an arbitrary non-empty set of essential variables in $f$. We have to show that $M \in Sep(f)$. Without loss of generality let us assume that $M = \{x_1, \ldots, x_m\}$, $1 \leq m \leq n$. If $m = n$ or $m = 1$, we are clearly done. Let $1 < m < n$ and let $\beta_{m+1}, \ldots, \beta_n \in Z_k$ be $n - m$ constants such that the subfunction $f_1 = f(x_{m+1} = \beta_{m+1}, \ldots, x_n = \beta_n)$ is not a constant function.

By symmetry, it suffices to show that $x_1 \in Ess(f_1)$. Without loss of generality let us assume that $a_0 \neq b_1$, where $b_1 = f_1(\gamma_1, \gamma_2, \ldots, \gamma_m)$ for some $\gamma_1, \ldots, \gamma_m \in Z_k$. Then, [1] implies

$$b_1 = f_1(\gamma_1, \gamma_2, \ldots, \gamma_m) \neq a_0 = f_1(\gamma_2, \gamma_2, \ldots, \gamma_m),$$

which shows that $x_1 \in Ess(f_1)$ and $Ess(f_1) = M$, and hence, $M \in Sep(f)$.

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

If $p > 2$ then according to Theorem 3.4 in [16], there exist two functions $h$ and $g$, such that $f = h \oplus g$, where $g \in G_{n,k}^n$ and $ess(h) = n - p$. Moreover, $g_{i \leftrightarrow j} = 0$ for all $i$ and $j$, $1 \leq j < i \leq n$. The same representation of $f$ is established when $p = 2$, in Theorem 4.2 [16]. Without loss of generality, let us assume that $Ess(h) = \{x_1, \ldots, x_{n-p}\}$.

Clearly, $Ess(f) = X_n$ and according to (1) we can pick $g = u \oplus v$, where

$$u = \left[ \bigoplus_{\beta \in Dis_n^k} a_\beta x^\beta \right] \text{ and } v = a_0 \left[ \bigoplus_{\alpha \in Eq_n^k} x^\alpha \right].$$

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

$$g_{i \leftrightarrow j} = u_{i \leftrightarrow j} \oplus v_{i \leftrightarrow j} = 0 \oplus a_0 \left[ \bigoplus_{\delta \in Z_n^{n-2}} \tilde{x}^\delta \right] = a_0.$$
be separable set in $g$, according to Case 5 and if $M \cap \text{Ess}(h) = \emptyset$ then it is also separable in $f$.

We have to prove that $M \in \text{Sep}(f)$ in all other cases. We argue by induction on $n$ - the number of essential variables in $f$ and $g$.

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

First, let $|M| = 2$ and $p = 2$. Clearly, if $M \subseteq \text{Ess}(h)$ then (2) and (3) show that $M \in \text{Sep}(f)$. Next, let us assume that $M = \{x_1, x_3\}$ and $\text{Ess}(h) = \{x_1, x_2\}$.

Let $c_2, c_4 \in Z_k$ be two constants, such that $\text{Ess}(f_1) = \{x_1, x_3\}$, where $t_1 = g(x_2 = c_2, x_4 = c_4)$. Hence, $x_3 \in \text{Ess}(f_1)$, where $f_1 = f(x_2 = c_2, x_4 = c_4)$. Let $h_1 = h(x_2 = c_2)$. If $x_1 \notin \text{Ess}(h_1)$ then $x_1 \in \text{Ess}(f_1)$ and obviously, $M \in \text{Sep}(f)$. If $x_1 \in \text{Ess}(h_1)$ then $f_1 = h_1(x_1) \oplus t_1(x_1, x_3)$. According to (2) and (3) there is a constant $c_3 \in Z_k$, such that $\text{Ess}(f_1(x_3 = c_3)) = \emptyset$. Hence, $x_1 \in \text{Ess}(f_1(x_3 = c_3))$ and $M \in \text{Sep}(f)$, again.

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

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

Let $x_1 \notin M$. Then there is a constant $d_4 \in Z_k$, such that $x_3 \in \text{Ess}(t_2)$, where $t_2 = g(x_4 = d_4)$. Clearly, $x_3 \in \text{Ess}(f_3)$, 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 $\{x_1, x_2, x_3\} = \text{Ess}(f_3)$, and hence $M \in \text{Sep}(f)$.

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

Let us assume that for some natural number $l$, $l \geq 4$, if $l < n$, $2 \leq p < l$ and $f \in G^n_{p,k}$, then all non-empty sets of essential variables in $f$ are separable.

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

$$t_3 = g(x_i = k - 1) = \left[ \bigoplus_{\beta \in \text{Dis}^{l-1}_{i-1}} b_{i} \tilde{x}^{\beta} \right],$$

and the coefficients $b_{i}$ linearly depend on $a_0, \ldots, a_{k-1}$, and $\tilde{x}^{\beta} = x_1^{\beta_1} \ldots x_{l-1}^{\beta_{l-1}}$. By $p \geq 2$ we may reorder the variables in $h$ such that $\text{Ess}(h) = \{x_1, \ldots, x_{l-p}\}$ with $l - p < l - 1$.

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

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

for all $i = 1, \ldots, l - p$. Now, Lemma 2.4 implies $N \cup \{x_1, \ldots, x_{l-p}\} \in \text{Sep}(f_4)$. Hence, $\text{Ess}(f_4) = X_{l-1}$. According to (2), it follows that $f_4 \in G^{l-1}_{p,k-1}$.

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

\[ \square \]

3. Minor decision diagrams of functions

Intuitively, it seems that a function $f$ has high complexity 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, 2]). Each path from the root (function node) to a terminal node (leaf) of BDD is called an implementation of f. In [15] we count the subfunction complexities imp(f), sub(f) and sep(f) of all implementations obtained under all n! variable orderings, subfunctions, and separable sets of n-ary Boolean functions for n, n \leq 5.

Example 3.1. Let f = x_1 \oplus x_2 \oplus x_3 (mod 2) and g = x_0^0 x_2 \oplus x_1 x_3 (mod 2) be two Boolean functions. Figure 2 presents their BDDs. All sets of essential variables in f are separable, whereas the set \{x_2, x_3\} is inseparable in g. Obviously, the BDD of the function f is extremely complex with respect to the numbers of its subfunctions and implementations (see [15]), whereas the BDD of g is very simple. Thus we have imp(f) = 48, sub(f) = 15, sep(f) = 7 and imp(g) = 28, sub(g) = 11, and sep(g) = 6.

This example shows that the functions having non-empty inseparable sets of essential variables must be quite simple with respect to the subfunction complexities, whereas Theorem 2.6 shows that the functions without inseparable sets of variables have trivial arity gap, which means that they must be with high minor complexity.

Roughly spoken, the complexity of functions, is a mapping (evaluation) Val : \mathcal{P}^n_k \rightarrow \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) \geq c for all f \in \mathcal{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, etc.

As mentioned, we have used the computational complexities sub(f), imp(f) and sep(f) in [15] to classify the functions from the algebra \mathcal{P}^n_k. These complexities are invariants under the action of the groups \mathcal{SB}^n_k, \mathcal{IM}^n_k and \mathcal{SP}^n_k.

Figure 3 shows the minor decomposition tree, constructed for the function f = x_0^0 x_1^1 \oplus x_2 x_3 x_4^2 (mod 3), which essentially depends on all of its four variables x_1, x_2, x_3 and x_4. The node at the left, labelled f - is the function node. The nodes represented as ovals and labelled with minor names are the internal (non-terminal) nodes, and the rectangular nodes (leaves of the tree) are the terminal nodes. The terminal nodes are labelled with the same name of a function (atomic
minors) from $P^1_k$ (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 $i = 1, \ldots, n - 1$.

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.

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 4 A), 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 in Figure 3 are $(f, f_{2-1})$, $(f, f_{3-1})$, $(f, f_{4-1})$, $(f_{3-1}, 0)$, $(f_{3-2}, 0)$ and $(f_{4-1}, 0)$.

**Figure 3.** Minor decomposition tree of $f = x_1^0x_2^1 + x_2^1x_3^1x_4^2 \pmod{3}$.
Example 3.2. Let us build the MDDs of the following two functions according to Theorem 2.3, with unique terminal node. Clearly, the MDD and MDT are uniquely determined by the function $f$. Figure 4 A) shows the MDD of the function $f$, obtained from its MDT, given in Figure 3, after applying the reduction rules. Figure 4 B) presents the MDD of $g$. The identification minors of $f$ and $g$ are:

$$
\begin{align*}
\text{f}_{2\to 1} &= x^0_1 x^1_2 x^2_4 \pmod{3}, & \text{f}_{3\to 1} &= x^0_1 x^2_2 + x^0_2 x^1_1 x^2_4 \pmod{3}, \\
\text{f}_{4\to 1} &= x^0_1 x^1_2 + x^0_2 x^1_3 x^2_4 \pmod{3}, & \text{f}_{3\to 2} &= x^0_1 x^2_1 \pmod{3},
\end{align*}
$$

and

$$
\text{g}_{2\to 1} = x^0_1 x^1_2 \pmod{3},
\text{g}_{4\to 1} = x^0_1 x^2_2 + x^0_2 x^1_3 x^2_1 \pmod{3},
\text{g}_{3\to 2} = x^0_1 x^1_2 \pmod{3}.
$$

Clearly, $f_{3\to 2} = f_{4\to 2} = f_{4\to 3} = [f_{4\to 1}]_{3\to 1} = [f_{4\to 1}]_{3\to 2} = [f_{3\to 1}]_{4\to 1} = [f_{3\to 1}]_{4\to 2} = x^0_1 x^1_2 \pmod{3}$. The minors $f_{i\to 1}$ for $i = 2, 3, 4$ are of order 1, and the last minor $f_{3\to 2}$ is of order 2.

The label of the edge $(f, f_{3\to 2})$ is 3 because there are three identification minors, namely $f_{3\to 2}, f_{4\to 2}$ and $f_{4\to 3}$ of $f$ which are equivalent to $f_{3\to 2}$ (see the last three branches of the MDT in Figure 3). In a similar way we count the labels of the edges in Figure 4 B). Note that $g_{3\to 1} = g_{4\to 1}$ and $g_{2\to 1, 4\to 1, 3\to 1} = g_{3\to 2}$, which implies that the nodes labelled with these minors must be merged in the MDD of $g$. The functions $f$ and $g$ are very close (the difference is that $x^1_2$ in $f$ is changed to $x^1_1$ in $g$) in their formula representations, but the diagram of $g$ is more complex as we see in Figure 4 B).

The size of the MDD and the minor complexities are determined by the function, being represented. 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 $\text{mnr}(f)$ 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.

We define two new complexity measures which count the number of minors and the number of ways to obtain these minors. Our goal is to classify functions in finite algebras by these complexities.

Definition 3.3. Let $f \in P^n_k$ be a k-value function. Its cmr-complexity $\text{cmr}(f)$ is defined as follows:

(i) $\text{cmr}(f) = 1$ if $\text{ess}(f) \leq 1$;
(ii) $\text{cmr}(f) = 2$ if $\text{ess}(f) = 2$;
(iii) $\text{cmr}(f) = \sum_{j<i, x_i, x_j \in \text{Ess}(f)} \text{cmr}(f_{i\to j})$ if $\text{ess}(f) \geq 3$.

The minors $f_{i\to j}$ with $i < j$ are excluded because $f_{i\to j} \equiv f_{j\to i}$. The minor complexity $\text{cmr}$ can be inductively calculated using the MDDs of the functions...
as it is shown in Algorithm 1 below. We start to assign $cmr$-complexity equals to 1 for the terminal node, according to (i) of Definition 3.3. Next, we calculate the $cmr$-complexity of the minors of $f$ with lower order, applying (ii) and (iii) of Definition 3.3.

Example 3.4. We now count the $cmr$-complexity of the functions $f$ and $g$ from Example 3.2 using their MDDs given in Figure 4 A) and B), respectively. There is one minor ($f_{3-2}$) of order 2 and three minors of order 1. Thus we have $cmr(f_{3-2}) = 2$, $cmr(f_{2-1}) = 1.3 = 3$, $cmr(f_{3-1}) = 1.1+2.2 = 5$, and $cmr(f_{4-1}) = 1.1+2.2 = 5$.

Again, using (ii) and (iii) of Definition 3.3 we obtain $cmr(f) = 3 + 5 + 5 + 3.2 = 19$.

The MDD of $g$ in Figure 4 B) shows that:

$cmr(g_{3-2}) = cmr((g_{3-1})_{4-1}) = 2$,
$cmr(g_{3-1}) = 1.2 + 1.1 + 1.2 = 5$,
$cmr(g_{4-1}) = 2.2 + 1.2 = 6$,
$cmr(g) = 1.4 + 2.2 + 2.5 + 1.6 = 24$.

We clearly have: $mnr(f) = 5$ and $mnr(g) = 6$.

It is clear that the set $M = \{x_1, x_3, x_4\}$ is in separable in both $f$ and $g$.

Theorem 3.5. Let $f \in P_k^n$ with $2 \leq \text{ess}(f) = n \leq k$. Then

(i) $\frac{n(n-1)}{2} \leq cmr(f) \leq \frac{n(n-1)!}{2^{n-1}}$;

(ii) $1 \leq mnr(f) \leq \frac{n(n-1)!}{2^{n-2}}$.

Proof. The maximum number of minors of order 1 for a function $f$ is equal to $\binom{n}{2}$. Using this as an inductive basis one can show that the maximum number of minors of order $m$, $1 \leq m \leq n-1$ is equal to $\binom{n}{2} \binom{n-1}{2} \ldots \binom{n-m+1}{2}$. Pick

$$f = x_1(x_2 \oplus 1) \ldots (x_n \oplus (n-1)) \pmod{k}$$
with \( n \leq k \). One can inductively prove that all the minors of \( f \) are pairwise distinct, which shows that \( f \) reaches the upper bound of (i) and (ii), and establishes the right inequalities in (i) and (ii).

The lower bound in (ii) is clear. If \( n \leq k \) then the minimum number of minors in a function depending essentially on \( n \) variables is reached to the functions \( f \) from \( G_{n,k}^n \) with \( \text{gap}(f) = n \). It follows that \( f \) is represented as in (1), which shows that \( \text{cmr}(f) = \binom{n}{2} = \frac{n(n-1)}{2} \). \( \square \)

**Remark 3.6.** Note that the upper bound of \( \text{cmr} \) and \( \text{mnr} \) is independent on \( k \) and hence, it is satisfied in all the possible cases for \( 2 \leq n \leq k \).

Later, we shall discuss lower and upper bounds of \( \text{cmr} \) and \( \text{mnr} \) when \( k < n \), and for Boolean functions. A hypothesis, here is that the upper bound for (ii) is unreachable for \( k \) and \( n \) if \( k < n \).

### 4. Equivalence relations with respect to minor complexities

Many of the problems in the applications of the \( k \)-valued logic are compounded because of the large number of the functions, namely \( k^n \). 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 \( S_{\{1,...,m\}} \) for a natural number \( m \), \( m \geq 1 \).

A **transformation** \( \psi : P_k^n \to P_k^n \) is an \( n \)-tuple of \( k \)-valued functions \( \psi = (g_1,\ldots,g_n) \), \( g_i \in P_k^n \), \( i = 1,\ldots,n \) acting on any 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 \( \psi \) 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** \( \Omega_k^n \) with unity - the identical transformation \( \epsilon = (x_1,\ldots,x_n) \). When taking only invertible transformations we obtain the **universal group** \( C_k^n \) isomorphic to the symmetric group \( S_{Z^n_k} \). The groups consisting of invertible transformations of \( P_k^n \) are called **transformation groups** (sometimes termed **permutation groups**).

Let \( \simeq \subseteq P_k^n \times P_k^n \) 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 \( P_1,\ldots,P_r \).

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 exactly the classes \( P_1,\ldots,P_r \). The number \( r \) of orbits of a group \( G_{\simeq} \) of transformations is denoted \( t(G_{\simeq}) \). Since we want to classify functions from \( P_k^n \) into equivalence classes by \( \simeq \), three natural problems occur, namely:

(i) Calculate the number \( t(G_{\simeq}) \) of \( G_{\simeq} \)-**types**;

(ii) Count the number of functions in different equivalence classes, i.e. compute the cardinalities of the sets \( P_1,\ldots,P_r \);

(iii) Make a catalogue (list) of functions belonging to different \( G_{\simeq} \)-**types**.
Let \( f \in P^n_k \) and let \( \text{nof}(f) \) denote the normal form obtained by applying the reduction \( \triangleright \) on \( f \). According to Theorem 2.3 the normal form \( \text{nof}(f) \) is unique and \( \text{nof}(f) \in P^n_k \). Thus, our first natural equivalence is defined as follows:

**Definition 4.1.** Let \( f \) and \( g \) be two functions from \( P^n_k \). We say that \( f \) and \( g \) are \( \text{nof} \)-equivalent (written \( f \equiv_{\text{nof}} g \)) if \( \text{nof}(f) = \text{nof}(g) \).

The transformation group induced by \( \text{nof} \)-equivalence is denoted \( NF^n_k \). The transformations in \( NF^n_k \) preserves \( \equiv_{\text{nof}} \), i.e. \( \text{nof}(g) = \text{nof}(\psi(g)) \) for all \( g \in P^n_k \) and \( \psi \in NF^n_k \). Since the atomic minors (labels of terminal nodes in MDD) depend on at most one essential variable, it follows that \( t(NF^n_k) = |P^n_k| = k^n \). These transformations involve permuting variables, only (see Theorem 4.6, below). The \( \text{nof} \)-equivalence is independent on the \( \text{cmr} \)-complexity of functions, defined by reduction via minors. For example, the functions \( f = 0 \) and \( g = x_1^3x_2 \oplus x_1x_2x_3^3 \mod 2 \) are \( \text{nof} \)-equivalent, but \( \text{cmr}(f) = 1 \) and \( \text{cmr}(g) = 6 \).

By analogy with the ordered decision diagrams \([2, 15]\), we define several equivalence relations in \( P^n_k \), which allow us to classify the functions by the complexity of their MDDs.

**Definition 4.2.** Let \( f \) and \( g \) be two functions from \( P^n_k \). We say that \( f \) and \( g \) are \( \text{cmr} \)-equivalent (written \( f \equiv_{\text{cmr}} g \)) if:

1. \( \text{ess}(f) \leq 1 \implies \text{ess}(f) = \text{ess}(g) \);
2. \( \text{ess}(f) \geq 2 \implies \text{ess}(f) = \text{ess}(g) \) and there exists a permutation \( \sigma \) of the set \( \{1, \ldots, n\} \) such that \( f_{i \rightarrow j} \equiv_{\text{cmr}} g_{\sigma(i) \leftarrow \sigma(j)} \) for all \( j, i \) with \( x_i, x_j \in \text{Ess}(f) \), \( j < i \).

Let \( CM^n_k \) denote the transformation group preserving the equivalence \( \equiv_{\text{cmr}} \). Note that if \( \text{ess}(f) \leq 1 \) then \( \text{Mnr}(f) = \emptyset \). Hence, if \( \text{ess}(f) = \text{ess}(g) \leq 1 \) then \( f \equiv_{\text{mnr}} g \). \( MN^n_k \) denotes the transformation group which preserves the equivalence \( \equiv_{\text{mnr}} \).

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

**Definition 4.3.** Let \( f \) and \( g \) be two functions from \( P^n_k \). We say that \( f \) and \( g \) are \( \text{mnr} \)-equivalent (written \( f \equiv_{\text{mnr}} g \)) if \( \text{mnr}(f) = \text{mnr}(g) \) for all \( m, 0 \leq m \leq \text{ess}(f) - 1 \).

Note that \( f \equiv_{\text{mnr}} g \) or \( f \equiv_{\text{nof}} g \) do not imply \( \text{ess}(f) = \text{ess}(g) \), which can be seen by the following functions: \( f = x_1^3x_2^2 \mod 3 \) and \( g = x_1^2x_2x_3^3 \mod 3 \). Clearly, \( f \equiv_{\text{mnr}} g \) and \( f \equiv_{\text{nof}} g \), but \( \text{ess}(f) = 2 \), and \( \text{ess}(g) = 3 \).

**Theorem 4.4.**

\[
\begin{align*}
(i) & \quad f \equiv_{\text{cmr}} g \implies \text{cmr}(f) = \text{cmr}(g) \\
(ii) & \quad f \equiv_{\text{cmr}} g \implies f \equiv_{\text{mnr}} g.
\end{align*}
\]

**Proof.** We argue by induction on the number \( n = \text{ess}(f) \).

If \( \text{ess}(f) \leq 2 \) (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 \( f \equiv_{\text{cmr}} g \). Then our inductive assumption implies

\[
\text{cmr}(f) = \sum_{j<i, \ x_i, x_j \in \text{Ess}(f)} \text{cmr}(f_{i \rightarrow j}) = 
\]
and \( mnr_m(f_{i-j}) = mnr_m(g_{u-v}) \), 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 \( CM^k_n \), so that if \( f \simeq cmr g \) then \( cmr(f) = cmr(g) \), and the complexity \( mnr(f) \) is an invariant of the group \( MN^k_n \), so that if \( f \simeq mnr g \) then \( mnr(f) = mnr(g) \).

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 \( \sigma : Z_k \rightarrow Z_k \) be a mapping and \( \psi : P^k_n \rightarrow P^k_n \) be a transformation of \( P^k_n \) generated by \( \sigma \) as follows \( \psi(f)(\omega a) = \sigma(f(\omega a)) \) for all \( \omega a \in Z^k_n \).

**Theorem 4.5.** The transformation \( \psi_{\sigma} \) preserves \( \simeq cmr \) if and only if \( \sigma \) is a permutation of \( Z_k \), \( k > 2 \).

**Proof.** First, let \( \sigma \in S_{Z_k} \) be a permutation of \( Z_k \). Let \( f \in P^k_n \) be an arbitrary function. If \( ess(f) \leq 1 \) then \( ess(\psi_{\sigma}(f)) = ess(f) \) and we are clearly done. Let \( ess(f) = ess(g) = n \geq 2 \) and let \( i \) and \( j \), \( 1 \leq j < i \leq n \) be two arbitrary natural numbers. Then we have

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

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

Second, let \( \sigma \) 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 \( \omega b = (b_1, \ldots, b_n) \in Z^k_n \), \( n \geq 2 \) be a vector of constants from \( Z_k \). Then we define the following function from \( P^k_n \):

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

Clearly, \( Ess(f) = X_n \) and the range of \( f \) consists of two numbers, i.e. \( A = \{a_1, a_2\} \). Then \( \sigma(A) = \{\sigma(a_1)\} \), 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 \( f \not\simeq cmr \psi_{\sigma}(f) \) and \( \psi_{\sigma} \notin CM^k_n \).}

Let \( \pi \in S_n \) and \( \phi_{\pi} : P^n_k \rightarrow P^n_k \) be a transformation of \( P^n_k \) defined as follows \( \phi_{\pi}(f)(a_1, \ldots, a_n) = f(\pi(a_1), \ldots, \pi(a_n)) \) for all \( (a_1, \ldots, a_n) \in Z^n_k \).

**Theorem 4.6.** The transformation \( \phi_{\pi} \) preserves the equivalence relations \( \simeq cmr \), \( \simeq mnr \) and \( \simeq nof \) for all \( \pi \in S_n \).

**Proof.** It suffices to show that \( \phi_{\pi} \) preserves \( \simeq cmr \) and \( \simeq nof \).

Let \( f \in P^n_k \) be a function and let us assume \( Ess(f) = X_n \), \( n \geq 2 \). It must be shown that \( f \simeq cmr g \) and \( f \simeq nof g \), where \( g(a_1, \ldots, a_n) = f(\pi(a_1), \ldots, \pi(a_n)) \) for all \( (a_1, \ldots, a_n) \in Z^n_k \). Since \( \pi \) is a permutation, we have

\[
f(a_1, \ldots, a_n) = g(\pi^{-1}(a_1), \ldots, \pi^{-1}(a_n)),
\]

for all \( (a_1, \ldots, a_n) \in Z^n_k \) and one can easily show that \( f \simeq cmr g \) and hence, \( f \simeq cmr \phi_{\pi}(f) \). Since \( nof(f) = f(x_1, \ldots, x_i) \) and \( nof(f) = f(x_{\pi(i)}, \ldots, x_{\pi(i)}) \), it follows \( nof(g) \equiv nof(f) \) and \( f \simeq nof g \).
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 $P_k^n$ [9] [11] [19].

In the model of RAG an affine transformation $\psi$ operates on the domain or space of inputs $x = (x_1, \ldots, x_n)$ to produce the output $y = xA + c$, which might be used as an input in the function $f$. Its output $f(y)$ 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:

$$g(x) = \psi(f)(x) = f(y) \oplus a_1x_1 \oplus \ldots \oplus a_nx_n \oplus d =$$

$$f(xA + c) \oplus a^*x \oplus d,$$

where $d$ and $a_i$ for $i = 1, \ldots, n$ are constants from $Z_k$. Such a transformation belongs to RAG if $A$ 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 $MN_k^n$. It is also interesting to compare these groups with the groups $IM_k^n, SB_k^n$ and $ST_k^n$, studied in [15].

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

For example, the permutation $\pi = (1,3,2)$ generates a permutation of the domain $\{0, 1, 2\}^3$ of the functions from $P_3^2$. Then we have $\pi: 001 \rightarrow 010 \rightarrow 100$ 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 $\pi$ 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 $Z_3^n$, 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 $S_k^2$-classes [9], as it is shown in Table 1.

| $[0]$, | $[x_1^0x_2^0]$, | $[x_1^0x_2^1, x_1x_2^0]$, | $[x_1^1x_2^1]$, | $[x_1^1, x_2^1]$, |
| $[x_1^0 \oplus x_2^0]$, | $[x_1^0 \oplus x_2^1]$, | $[x_1^0 \oplus x_1x_2^0]$, | $[x_1^1 \oplus x_1^{0}x_2^1]$, | $[x_1^0 \oplus x_1^{0}x_2^1]$, |

M. Harrison has determined the cycle index of the group $S_k^n$ and using Polya’s counting theorem he has counted the number of equivalence classes under permuting arguments (see [9] and Table 1 below).

The following proposition is obvious.

**Proposition 4.7.** The transformation group $S_k^n$ is induced by the equivalence relation $\equiv$ (see Definition 2.2).
The subgroups of RAG, defined according to \(5\) are determined by equivalence relations as it is shown in Table 2 where \(P\) denotes a permutation matrix, \(I\) is the identity matrix, \(b\) and \(c\) are \(n\)-dimensional vectors over \(\mathbb{Z}_k\) and \(d \in \mathbb{Z}_k\).

**Table 2. The subgroups of RAG**

| Groups | Equivalence relations | Determination |
|--------|----------------------|--------------|
| RAG    | Affine transformation| \(A\)-non-singular |
| \(GE_k^n\) | Genus | \(A = P, a = 0\) |
| \(CF_k^n\) | Complement arguments | \(A = I, a = 0, c = 0\) |
| \(G_k^n\) | Symmetry types | \(A = I, c = 0, d = 0\) |
| \(LF_k^n\) | Add linear function | \(A = I, c = 0, d = 0\) |
| \(CA_k^n\) | Complement arguments | \(A = I, a = 0, d = 0\) |
| \(LG_k^n\) | Linear transformation | \(c = 0, a = 0, d = 0\) |
| \(S_k^n\) | Permute variables | \(A = P, c = 0, a = 0, d = 0\) |

It is naturally to ask which subgroups of RAG are subgroups of the group \(NF_k^n\), \(CM_k^n\) and \(MN_k^n\). Theorem 4.5 and Theorem 4.6 show that \(CF_k^n\) and \(S_k^n\) are subgroups of \(CM_k^n\). Theorem 4.4 shows that they must also be subgroups of \(MN_k^n\). Clearly, \(S_k^n \leq NF_k^n\).

**Example 4.8.** Let \(f = x_1 + x_2 + x_3 (mod\ 3)\) and \(g = x_1x_2 + x_1x_3 + x_2x_3 (mod\ 3)\). Then we have \(f_{i-j} = 2x_j + x_m (mod\ 3)\) and \(g_{i-j} = 2x_j + x_m (mod\ 3)\) where \(\{i, j, m\} = \{1, 2, 3\}\). Clearly, \(f_{i-j, j, m} = g_{i-j, j, m} = 0\), 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, \(CM_k^n \not\leq RAG\), \(NF_k^n \not\leq RAG\) and \(MN_k^n \not\leq RAG\).

**Example 4.9.** Let \(f = x_1^0x_2^1 + x_1^0x_3^2 (mod\ 3)\) and \(g = x_1^0x_2^1 + x_1^0x_3^2 (mod\ 3)\) be the functions from Example 3.4 whose MDDs are given in Figure 4. Let

\[
A = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 2
\end{pmatrix}
\]

Then clearly, \(f(x) = g(Ax)\) and hence, \(f\) and \(g\) belong to the same equivalence class under the transformation groups \(LG_k^n\). Let \(c = (0, 0, 0, 1)\). Then we have \(f(x) = g(xI + c)\), which shows that \(f\) and \(g\) belong to the same equivalence class under the transformation group \(CA_k^n\). One can show that \(f \simeq_{nof} g\). Example 3.4 shows that \(f \not\simeq_{cmr} g\). Consequently, \(NF_k^n \not\leq MN_k^n\), \(LG_k^n \not\leq MN_k^n\) and \(CA_k^n \not\leq MN_k^n\). Theorem 4.4 shows that \(NF_k^n \not\leq CM_k^n\), \(LG_k^n \not\leq CM_k^n\) and \(CA_k^n \not\leq CM_k^n\).

**Example 4.10.** Let \(f = x_1^0x_2 + x_1^0x_2x_3^1x_2^0 (mod\ 3)\) and \(g = x_1^0x_2 + x_1^0x_2x_3 (mod\ 3)\) be two functions. It is easy to see that \(f_{i-j} = g_{i-j}\) for all \(i, j\) with \(1 \leq j < i \leq 3\). Hence, \(f \simeq_{cmr} g\) and \(f \simeq_{nof} g\). Now, it is clear that each set of essential variables in \(f\) is separable in \(f\), but \(\{x_2, x_3\} \not\in \text{Sep}(g)\) which shows that \(f \not\simeq_{sep} g\), i.e. \(CM_k^n \not\leq SP_k^n\) and \(NF_k^n \not\leq SP_k^n\).
So, the next theorem summarizes results which determine the positions of the groups $\text{NF}_n$, $\text{CM}_n$ and $\text{MN}_n$, with respect to the subgroups of RAG and the groups induced by subfunction complexities [15].

**Theorem 4.11.**

(i) $\text{CF}_n \leq \text{CM}_n$; (ii) $\text{S}_n \leq \text{CM}_n$; (iii) $\text{S}_n \leq \text{NF}_n$;

(iv) $\text{NF}_n \not\leq \text{RAG}$; (v) $\text{CM}_n \not\leq \text{RAG}$; (vi) $\text{LG}_n \not\leq \text{MN}_n$;

(vii) $\text{CA}_n \not\leq \text{MN}_n$; (viii) $\text{CA}_n \not\leq \text{NF}_n$;

(ix) $\text{LG}_n \not\leq \text{NF}_n$; (x) $\text{CM}_n \not\leq \text{SP}_n$;

(xi) $\text{CM}_n \not\leq \text{SP}_n$; (xii) $\text{CM}_n \not\leq \text{NF}_n$;

(xiii) $\text{NF}_n \not\leq \text{MN}_n$.

**Proof.** (i) follows from Theorem 4.5, (ii) and (iii)- from Theorem 4.6, (iv) and (v) - from Example 4.8, (vi), (vii), (viii), (ix) and (xiii) - from Example 4.9, (x) and (xi)- from Example 4.10.

To show (xii) let us pick $f = x_0^1x_2^0 \pmod{2}$ and $g = x_1x_2^0 \pmod{2}$. Clearly, $f \simeq_{\text{cmr}} g$, but $\text{nof}(f) = x_0^1 \not\equiv x_1 = \text{nof}(g)$. □ □

Theorem 4.11 is well-illustrated by Figure 5, in the case of Boolean functions.

5. Classification of Boolean functions by minor complexities

Table 3 shows the four classes in $P_2^2$ under the equivalence $\simeq_{\text{cmr}}$. The $\simeq_{\text{cmr}}$-classes are represented as union of several classes under the permuting arguments, according to Theorem 4.11 (ii) (see Tables 1 and Table 3).

| $n$ | $\text{S}_n^2$ | $\text{CM}_n^2$ | $\text{MN}_n^2$ | $\text{IM}_n^2$ | $\text{SB}_n^2$ | $\text{SP}_n^2$ |
|-----|----------------|----------------|-----------------|----------------|----------------|----------------|
| 1   | 4              | 2              | 2               | 2              | 2              | 2              |
| 2   | 12             | 4              | 3               | 4              | 4              | 3              |
| 3   | 80             | 11             | 5               | 13             | 11             | 5              |
| 4   | 3984           | *              | *               | 104            | 74             | 11             |
| 5   | 37 333 248     | *              | *               | *              | *              | *              |
| 6   | 25 626 412 338 | 274 304        | *               | *              | *              | *              |

Table 4. Number of equivalence classes in $P_2^n$ under transformation groups.

Figure 5 presents the subgroups of RAG and transformation groups whose invariants are subfunction and minor complexities of Boolean functions of $n$-variables. According to Theorem 4.11 the group $\text{CM}_n^2$ has three subgroups from RAG, namely: $\text{S}_2^0$ - the group of permuting arguments, trivial group, consisting of the identity map,
only and $CF^n_2$ - the group of complementing outputs. The groups $NF^n_2$ and $MN^n_2$ are not subgroups of any subgroup of RAG and also, they are not subgroups of any group among $IM^n_2$, $SP^n_2$ and $SB^n_2$.

![Figure 5. Transformation groups in $P^n_2$.](image)

Table 5 presents a full 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 $\{0, \ldots, 127\}$. According to Theorem 4.5, if a natural number $z$ 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 (see the last column) contains the numbers $\leq 127$, only. They 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 4.4 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 5). 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.

6. Appendix

Table 5 presents classification of ternary Boolean functions under the equivalences $\simeq_{cmr}$ and $\simeq_{mnr}$, including the catalogue of the equivalence classes (last column).

**Example 6.1.** Let us choose a natural number belonging to the seventh column of Table 5, say 24. It belongs to the row numbered 6. The binary representation of 24 is 00011000, because $24 = 1 \cdot 2^4 + 1 \cdot 2^3$. Hence, the function $f$ corresponding to 24 is evaluated by $1$ on the fourth and fifth miniterms, namely $x_0^3x_2x_3$ and $x_1x_2^3x_3^0$. Consequently, $f = x_1^0x_2x_3 \oplus x_1x_2^3x_3^0 \pmod{2}$. Then we have $f_{2-1} = f_{3-1} = 0$ and $f_{3-2} = x_1^0x_2 \oplus x_1x_2^3 \pmod{2}$. 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 $x_1x_2x_3^0 (mod 2)$ is representative for this class (sixth cell). The numerical list of the function from this equivalence class is given in the last seventh cell of Table 5.

We also provide an algorithm to find the complexity $cmr(f)$ of an arbitrary $k$-valued function $f$. Similar algorithms for manipulation of Boolean functions are presented in [2, 13]. We shall express our algorithm in a pseudo-Pascal notation. The main data structure describes the nodes in the MDD of $f$. Each node is represented by a record declared as follows:

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

The first field named $ess$ presents the number of essential variables in $f$ and the second field $val$ is a natural number whose $k$-ary representation is the last column $b$ of the truth table (of size $k^n$) of $f$. For example, the function $f$ from Example 6.1 is presented as $f.ess=3$ and $f.val=24$, where $k=2$ and $b=00011000$.

The algorithm that computes $GetMinor(g,i,j)$ uses well-known manipulations on the rows and columns of the truth table, which realise collapsing the $i$-th and $j$-th column and removing $m$-th row if $a_{mi} \neq a_{mj}$ for all $m = 1, \ldots, k^n$. This algorithm has to use a procedure, which excludes all inessential variables in the resulting minor.

The function $GetMinor(g,i,j)$ realizes one step of the reduction $\triangleright$. It is also useful when constructing the MDD of a function. Then the basic algorithm in addition, has to calculate the labels of the edges in MDD and to test whether two minors are equivalent (in terms of Definition 2.2).
Table 5. Minor classification of ternary Boolean functions.

| 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         | $f$   | $x_0$               | 2,18| 1                | $x_1x_2^0$| 10,12,34,48,60,68,80,90,102 |
| 4         | $x_1$ | 2                   | 12  |                  | $x_1x_2$  | 3,5,17,63,95,119 |
| 5         | $f$   | $x_1$               | 3,8 | 20               | $x_1 \oplus x_2 \oplus x_3$ | 43,77,105,113 |
| 6         | $f$   | $x_0$               | 4,18|                   | $x_1x_2x_3^0$ | 2,4,8,16,24,32,36,64,66 |
| 7         | $f$   | $x_0$               | 5,36|                   | $x_1x_2^0x_3$ | 6,18,20,26,28,38,40,44,52,56,70,72,74,82,88,96,98,100 |
| 8         | $f$   | $x_0$               | 6,54,108 | 2           | $x_1x_2^0x_3$ | 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         | $f$   | $x_1x_2$            | 4,50|                   | $x_1x_2x_3^0$ | 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        | $f$   | $x_1x_2$            | 5,36|                   | $x_1x_2x_3$ | 9,27,29,33,39,45,53,57,65,71,75,83,89,99,101,111,123,125 |
| 11        | $f$   | $x_1x_2$            | 6,16| 102              | $x_1x_2x_3$ | 1,25,37,61,67,91,103,127 |
Algorithm 1 Counting cmr(f)

1: type minor=record
   ess: 1..n;
   val: 0..k^n − 1;
end;
2: var f:minor;
3: cmr:integer;
4: function GetMinor(g:minor; i,j:integer): minor;   ▷ Getting minor
5:   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;
6:   n := g.ess;
7:   Create truth table A_{k^n \times n}B of g;
8:   Create truth table H_{k^{n-1} \times n}L of h := g_{i→j};  ▷ Use truth table of g
9:   Calculate - h.ess and h.val from table HL;
10:  GetMinor := h;
11: end function;
12: function Complexity(g:minor):integer;   ▷ Counting complexity
13:   n := g.ess;
14:   if n > 2 then
15:      for j, 1 ≤ j ≤ n − 1 do
16:         for i, j + 1 ≤ i ≤ n do
17:            h := GetMinor(g,i,j);
18:            Complexity := Complexity + Complexity(h);
19:      end for
20:   end for
21:   else  ▷ Basis of recursion
22:      if n = 2 then
23:         Complexity := 2
24:      else
25:         Complexity := 1
26:      end if
27:   end if
28: end function
29: Input k; f.ess; f.val;
30: cmr:=Complexity(f);
31: Print cmr.

7. Conclusion

The traditional complexity measures of functions are based on the reductions downto subfunctions or values of functions. They are numerical parameters of the decision diagrams built under different variable orderings.

The minor complexities present another concept of computing finite functions, namely when identifying variables.

The relationship between subfunction and minor complexities in functions seems to be "strange". First of all, the functions with simplest minor representations (with non-trivial arity gap) has extremely complex representations with respect to their
subfunctions (Theorem 2.6). So, all functions with at least one inseparable set have trivial arity gap.

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, namely the groups listed above, and $CA_k^n$, $G_k^n$, and $GE_k^n$ (see Figure 3 and Table 2).

One of motivations to study the group $NF_k^n$ is that the reductions are inexpensive (see Algorithm 1, below) and the number of classes is much smaller than with (say) $GE_k^n$, because the order of $NF_k^n$ is so large. The order of $GE_k^n$ is $nk^n$ (see [11]). As mentioned, the number of equivalence classes under $NF_k^n$ equals to $k^{k^n}$. Hence, the order of $NF_k^n$ is equal to $k^{k^n}$.

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 $k$-valued functions all 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 5), which consists of 16 function. This class is not so large. It presents 1/16 of the all 256 ternary Boolean functions.

References

[1] H. Andersen. An introduction to binary decision diagrams, 1997. Lecture notes for 49285 Adv. Algorithms.
[2] R. E. Bryant. Graph-based algorithms for boolean function manipulation. IEEE Transactions on Computers, C-35(8):677–691, 1986.
[3] K. Chimev. Separable sets of arguments of functions. MTA SzTAKI Tanulmanyok, 80:1–173, 1986.
[4] M. Couceiro and E. Lehtonen. Generalizations of Świerczkowski’s lemma and the arity gap of finite functions. Discrete Mathematics, 309:5905–5912, 2009.
[5] M. Couceiro, E. Lehtonen, and T. Waldhauser. GAP vs. PAG. Multiple-Valued Logic, 42:268–273, 2012.
[6] M. Couceiro, E. Lehtonen, and T. Waldhauser. Parametrized arity gap. Order, 30:557–572, 2013.
[7] J. Denev and I. Gyudzhenov. On separable subsets of arguments of functions from $p_k$. MTA SzTAKI Tanulmanyok, 26(147):47–50, 1984.
[8] E. Dubrova. Multiple-valued logic in VLSI: challenges and opportunities. In Proceedings of NORCHIP’99, pages 340–350, Oslo, Norway, 1999.
[9] M. Harrison. Counting theorems and their applications to classification of switching functions. In A. Mikhopadhyay, editor, Recent Developments in Switching Theory, pages 85–120. NY, Academic Press, 1971.
[10] J. Klop and R. de Vrijer. Term rewriting systems. Cambridge University Press, 2003.
[11] R. J. Lechner. Harmonic analysis of switching functions. In A. Mikhopadhyay, editor, Recent Developments in Switching Theory, pages 121–228. NY, Academic Press, 1971.
[12] A. Salomaa. On essential variables of functions, especially in the algebra of logic. Annales Academia Scientiarum Fennicae, Ser. A(333):1–11, 1963.
[13] J. E. Savage. Models of computation, Exploring the Power of Computing. Brown University, http://cs.brown.edu/people/jes/book/, 2008.
[14] Sl. Shtrakov. Essential arity gap of boolean functions. Serdica, Journal of Computing, 2(3):249–266, 2008.
[15] Sl. Shtrakov and I. Damyanov. On the computational complexity of finite operations. International Journal of Foundations of Computer Science, 27(01):15–38, 2016.
[16] Sl. Shtrakov and J. Kopptitz. On finite functions with non-trivial arity gap. J. Discussiones Mathematicae - General Algebra and Applications, 30:217–245, 2010.
[17] Sl. Shtrakov and J. Koppitz. Finite symmetric functions with non-trivial arity gap. *Serdica, Journal of Computing*, 6(4):419–436, 2012.

[18] Sl. Shtrakov and J. Koppitz. Stable varieties of semigroups and groupoids. *Algebra Universalis*, 75(1):85–106, 2016.

[19] I. Strazdins. On fundamental transformation groups in the algebra of logic. In B. Csákány and I. Rosenberg, editors, *Finite algebra and multiple-valued logic*, volume 28, pages 669–691, Szeged, 1979. Colloquia Mathematica Societatis János Bolyai.

[20] R. Willard. Essential arities of term operations in finite algebras. *Discrete Mathematics*, 149:239–259, 1996.

E-mail address: shtrakov@swu.bg
URL: http://shtrakov.swu.bg/

Department of Computer Science, South-West University, Blagoevgrad