Discrete Mathematics 340 (2017) 1957–1966

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

Block composition algorithm for constructing orthogonal n-ary operations

Iryna V. Fryz [a,*], Fedir M. Sokhatsky [b,**]

[a] Khmelnytskyi National University, Instytuts'ka str. 11, Khmelnytskyi, 29016, Ukraine

[b] Donetsk National University, 600-richia str. 21, Vinnytsia, 21000, Ukraine

ARTICLE INFO

Article history:
Received 22 July 2015
Received in revised form 23 May 2016
Accepted 9 November 2016
Available online 4 December 2016

Keywords:
Orthogonality of operations
Retract orthogonality of operations
Composition algorithm
Block-wise recursive algorithm
Block composition algorithm

ABSTRACT

We propose an algorithm for constructing orthogonal n-ary operations which is called a *block composition algorithm* here. Input data of the algorithm are two series of different arity operations being distributed by blocks. The algorithm consists of two parts: composition algorithm for constructing n-ary operations with orthogonal retracts from given blocks of operations and block-wise recursive algorithm for constructing orthogonal n-ary operations from obtained operations. Obtained results are illustrated by examples of orthogonal n-ary operations which are constructible by block-wise recursive algorithm and non-constructible by the well-known trivial recursive algorithm.

1. Introduction

Construction of MDS-codes, hash-functions, and secret-sharing schemes is reduced to construction of orthogonal n-ary operations, partial orthogonal n-ary operations, orthogonal n-ary quasigroups, for example, [2,4,6,8].

Even in the case $n = 2$, constructing orthogonal operations is non-trivial and many questions remain open although its study started in L. Euler's work and attracts attention up to the present time. The detailed review of well-known methods of construction of orthogonal binary operations is considered in [5].

For $n > 2$ constructing orthogonal operations is less investigated. Here we consider only one of the constructing methods of orthogonal operations, namely a generalization of the method of defining recursive derivative functions which are connected with recursive MDS codes via orthogonality as shown by E. Couselo and others in [4]. This idea was developed by G.B. Belyavskaya, G.L. Mullen [3]. Their method is a special case of *trivial recursive algorithm* which is given here. By virtue of trivial recursion, every next operation is constructed from a new operation and operations which have been constructed previously.

In this article, we propose a block composition algorithm for constructing orthogonal operations (Theorem 8). It contains two parts: *composition algorithm* for constructing operations with orthogonal retracts (Theorem 3) and *block-wise recursive algorithm* for constructing multiary orthogonal operations using some partition of a variable set into blocks (Theorem 5). By view of block recursion, the next block of operations is created from a block of new operations and operations which have already been constructed. A special case of this algorithm was published in [7].

The concept of retract orthogonality can be considered as a generalization of invertibility of an operation. Block-wise recursive algorithm becomes trivial when blocks of the partition are trivial, i.e., singletons (Corollary 6). Example 1

[*] Corresponding author.

[**] Corresponding author.

E-mail addresses: iryna.fryz@ukr.net (I.V. Fryz), fmsokha@ukr.net (F.M. Sokhatsky).

http://dx.doi.org/10.1016/j.disc.2016.11.012
0012-365X/

demonstrates a construction of tuples of 4-ary orthogonal operations using a block-wise recursive algorithm. These tuples cannot be constructed using a trivial recursive algorithm.

2. Preliminaries

In this article, all operations are defined on the same arbitrary set which we call a carrier and denote by Q. For notational convenience x_i^j denotes the sequence $x_i, x_{i+1}, \dots, x_j$ if $i \le j$ or empty sequence otherwise.

An operation f is called *i-invertible*, if for arbitrary elements $a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n$ there exists a unique element x such that

$$f(a_1, \dots, a_{i-1}, x, a_{i+1}, \dots, a_n) = b.$$

If f is i-invertible for all $i \in \overline{1, n} := \{1, 2, \dots, n\}$,[1] then it is called an *invertible* or *quasigroup operation*. Operation f, being the mapping $Q^n \to Q$, is called *surjective*, if for every $b \in Q$ there exist $a_1, \dots, a_n \in Q^n$ such that $f(a_1, \dots, a_n) = b$.

For every permutation σ of the set $\overline{1, n+1}$ and invertible operation f a *σ-parastrophe* ${}^{\sigma}\!f$ is defined by

$${}^{\sigma}\!f(x_{1\sigma}, \dots, x_{n\sigma}) = x_{(n+1)\sigma} :\Longleftrightarrow f(x_1, \dots, x_n) = x_{n+1}.$$

In particular, a σ-parastrophe is called

(1) an *i-th division* if $\sigma = (i, n+1)$;
(2) *principal* if $(n+1)\sigma = n+1$.

It is easy to see that a principal σ-parastrophe can be defined by

$${}^{\sigma}\!f(x_{1\sigma}, \dots, x_{n\sigma}) = f(x_1, \dots, x_n) \tag{1}$$

or

$${}^{\sigma}\!f(x_1, \dots, x_n) = f(x_{1\sigma^{-1}}, \dots, x_{n\sigma^{-1}}). \tag{2}$$

Throughout the article the symbol S_A denotes the set of all permutations of the set $A \subset \overline{1, n}$, but S_{n+1} refers to the set of all permutations of the set $\overline{1, n+1}$, where n is a natural number, and

$$\mathrm{S}'_{n+1} := \{\sigma \in \mathrm{S}_{n+1} \mid (n+1)\sigma = n+1\}.$$

It is obvious that S'_{n+1} is a subgroup of the symmetric group S_{n+1}.

Usually, for $\tau \in \mathrm{S}_{n+1}$ the symbol $(X)\tau$ denotes the image of a set X under transformation τ, i.e., $(X)\tau := \{x\tau \mid x \in X\}$. Also we need the following subset of S_{n+1}:

$$\mathrm{S}^A_{n+1} := \left\{\tau \in S'_{n+1} \mid (A)\tau = \{1, \dots, |A|\}\right\}.$$

A k-tuple of n-ary operations $(f_1, \dots, f_k)$ defined on Q of the order m is called *orthogonal* if every system

$$\begin{cases} f_1(x_1, \dots, x_n) = a_1, \\ \dots\dots\dots\dots\dots\dots \\ f_k(x_1, \dots, x_n) = a_k, \end{cases}$$

where $a_1, \dots, a_k \in Q$, has exactly m^{n-k} solutions. If $k = n$, then each of these systems has a unique solution.

The relationship

$$\overline{\theta}(x_1, \dots, x_n) = (f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n)) \quad \left(\overline{\theta} = (f_1, \dots, f_n)\right)$$

defines a one-to-one correspondence between the set of all tuples of orthogonal operations defined on Q and the set of all permutations of Q^n.

If $\overline{\theta} := (f_1, \dots, f_n)$ is a permutation of Q^n and $(g_1, \dots, g_n) := \overline{\theta}^{-1}$, then the equality $\overline{\theta}^{-1}\overline{\theta} = \iota$ is equivalent to the following system of identities:

$$\begin{cases} g_1(f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n)) = x_1, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ g_n(f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n)) = x_n. \end{cases}$$

In every tuple of orthogonal operations, all operations are pairwise different and every two tuples differing only in the order of operations are orthogonal simultaneously. That is why an arbitrary tuple of orthogonal operations from the set $\{(f_{1\sigma}, \dots, f_{n\sigma}) \mid \sigma \in S_n\}$ will be denoted by $\{f_1, \dots, f_n\}$.

[1] Symbol := denotes "*is equal by definition*" and :⟺ denotes "*is equivalent by definition*".

There exist different kinds of visual methods for constructing orthogonal binary operations, but it is problematic to generalize them to multiary operations, so analytical approaches are reasonable. One of them can be found in [3, Theorem 3]. We present Belyavskaya and Mullen's algorithm:

n-ary operations $g_1, \dots, g_n$ are constructed using recursion of operations $f_1, \dots, f_n$ as follows

$$\begin{cases} g_1(x_1,\dots,x_n) = f_1(x_1,\dots,x_n), \\ g_2(x_1,\dots,x_n) = f_2(x_1,\dots,x_{n-1}, g_1(x_1,\dots,x_n)), \\ g_3(x_1,\dots,x_n) = f_3(x_1,\dots,x_{n-2}, g_1(x_1,\dots,x_n), g_2(x_1,\dots,x_n)), \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ g_i(x_1,\dots,x_n) = f_3(x_1,\dots,x_{n-i+1}, g_1(x_1,\dots,x_n),\dots,g_{i-1}(x_1,\dots,x_n)), \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ g_n(x_1,\dots,x_n) = f_n(x_n, g_1(x_1,\dots,x_n),\dots,g_{n-1}(x_1,\dots,x_n)). \end{cases} \tag{3}$$

3. Orthogonality of retracts

For orthogonality of n-ary operations, some problems which have no analogues in the binary case are left over without attention. In this section we propose an answer to the following problem: Do tuples of operations, whose retracts are orthogonal, exist?

Orthogonality and parastrophy.

Let f be an n-ary operation defined on a set Q and let

$$\delta := \{i_1,\dots,i_k\} \subseteq \overline{1,n}, \quad \{j_1,\dots,j_{n-k}\} := \overline{1,n} \setminus \delta, \quad \bar a := (a_{j_1},\dots,a_{j_{n-k}}).$$

An operation $f_{(\bar a,\delta)}$ which is defined by

$$f_{(\bar a,\delta)}(x_{i_1},\dots,x_{i_k}) := f(y_1,\dots,y_n),$$

where $y_i := \begin{cases} x_i, & \text{if } i \in \delta, \\ a_i, & \text{if } i \notin \delta \end{cases}$, is called $(\bar a, \delta)$*-retract* or δ*-retract* of f.

Operations $f_{1;(\bar a_1,\delta)}, f_{2;(\bar a_2,\delta)}, \dots, f_{k;(\bar a_k,\delta)}$ are called *similar* δ-retracts of n-ary operations $f_1, f_2, \dots, f_k$, if $\bar a_1 = \bar a_2 = \dots = \bar a_k$. If all similar δ-retracts of $f_1, f_2, \dots, f_k$ are orthogonal, then the operations $f_1, f_2, \dots, f_k$ are said to have *orthogonal* δ*-retracts*.

If $\delta = \overline{1,n}$, then δ-retract orthogonality is orthogonality given above. When $\delta = \{i\}$, then δ-retract orthogonality degenerates into i-invertibility of operation f_i, i.e., retract orthogonality can be considered as some generalization of invertibility.

Let $f_1, \dots, f_k$ be operations with orthogonal δ-retracts. By view of the definition, it means that for every $\bar a \in Q^{n-k}$ their $(\bar a, \delta)$-retracts $f_{1;(\bar a,\delta)}, \dots, f_{k;(\bar a,\delta)}$ are orthogonal. In other words, a transformation

$$\overline{\theta} := (f_{1;(\bar a,\delta)}, \dots, f_{k;(\bar a,\delta)})$$

of the set Q^k is its permutation. Let $(g_1,\dots,g_k) := \overline{\theta}^{-1}$, then the equality $\overline{\theta}^{-1}\overline{\theta} = \iota$ can be written as follows

$$\begin{cases} g_1(f_1(y_1,\dots,y_n),\dots,f_k(y_1,\dots,y_n)) = x_{i_1}, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ g_k(f_1(y_1,\dots,y_n),\dots,f_k(y_1,\dots,y_n)) = x_{i_k}, \end{cases} \tag{4}$$

where $y_i := \begin{cases} x_i, & \text{if } i \in \delta, \\ a_i, & \text{if } i \notin \delta. \end{cases}$

Note that the set of all x's indices in (4) is δ.

Thus the following statement is true:

Lemma 1. *δ-retract orthogonality of n-ary operations $f_1,\dots,f_k$ means that for every $\bar a \in Q^{n-k}$ there exist k-ary operations $g_1,\dots,g_k$ such that the system of identities (4) holds.*

To show dependences among orthogonalities of retracts we introduce a new notation. Let $\delta := \{i_1,\dots,i_k\}, \sigma \in S'_{n+1}$ then

$$^{\sigma}\delta := \{(i_1)\sigma^{-1},\dots,(i_k)\sigma^{-1}\}. \tag{5}$$

Lemma 2. *Let $\sigma \in S'_{n+1}$ and $f_1,\dots,f_k$ be n-ary operations. A tuple $\{f_1,\dots,f_k\}$ has orthogonal δ-retracts if and only if the tuple $\{^{\sigma}f_1,\dots,{}^{\sigma}f_k\}$ has orthogonal $^{\sigma}\delta$-retracts.*

Proof. Let $\delta := \{i_1,\dots,i_k\}$. According to Lemma 1 and equality (1) the system (4) can be written as follows:

$$\begin{cases} g_1({}^{\sigma}f_1(y_{1\sigma},\dots,y_{n\sigma}),\dots,{}^{\sigma}f_k(y_{1\sigma},\dots,y_{n\sigma})) = x_{i_1}, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ g_k({}^{\sigma}f_1(y_{1\sigma},\dots,y_{n\sigma}),\dots,{}^{\sigma}f_k(y_{1\sigma},\dots,y_{n\sigma})) = x_{i_k}. \end{cases}$$

Here the variable x_{i_1} is in $i_1\sigma^{-1}$-th place and so on, x_{i_k} is in $i_k\sigma^{-1}$-th place. By (5) we conclude that the collection $\{{}^{\sigma}f_1, \dots, {}^{\sigma}f_k\}$ has orthogonal ${}^{\sigma}\delta$-retracts. □

This Lemma implies that it is enough to consider the case $\delta = \{1, \dots, k\}$.
Lemma 2 implies p.3 of Proposition 2.2 from [9] when $\delta = \{1, \dots, n\}$.

Constructing operations with orthogonal retracts .

Operations with orthogonal retracts can be constructed using repetition-free composition.

Theorem 3. *Let $p_1, \dots, p_k$ be arbitrary 1-invertible $(n-k+1)$-ary operations, $h_1, \dots, h_k$ be arbitrary k-ary operations, and let operations $f_1, \dots, f_k$ be defined by*

$$\begin{cases} f_1(x_1, \dots, x_n) := p_1(h_1(x_1, \dots, x_k), x_{k+1}, \dots, x_n), \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ f_k(x_1, \dots, x_n) := p_k(h_k(x_1, \dots, x_k), x_{k+1}, \dots, x_n). \end{cases} \tag{6}$$

Then n-ary operations $f_1, \dots, f_k$ have orthogonal $\overline{1, k}$-retracts if and only if k-ary operations $h_1, \dots, h_k$ are orthogonal.

Proof. In (6) we put $a_{k+1} \in Q$ instead of x_{k+1} and so on, $a_n \in Q$ instead of x_n. We consider for any $b_1, \dots, b_n \in Q$ the corresponding system. From 1-invertibility of $p_1, \dots, p_k$ the system

$$\begin{cases} h_1(x_1, \dots, x_k) = {}^{\ell}p_1(b_1, a_{k+1}, \dots, a_n), \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ h_k(x_1, \dots, x_k) = {}^{\ell}p_k(b_k, a_{k+1}, \dots, a_n) \end{cases}$$

is obtained. Because the operations $p_1, \dots, p_k$ are surjective, the last statement is equivalent to orthogonality of the operations $h_1, \dots, h_k$. □

For arbitrary δ, a tuple of orthogonal δ-retracts of n-ary operations can be constructed using the algorithm given below.
Composition algorithm. *Let $\delta \subseteq \overline{1, n}$, $n \geqslant k$ and let $h_1, \dots, h_k$ be k-ary operations, $p_1, \dots, p_k$ be $(n-k+1)$-ary operations, $\sigma \in S'_{n+1}$.*
Operations ${}^{\sigma}f_1, \dots, {}^{\sigma}f_k$ are constructed by the following items:

(1) *operations $f_1, \dots, f_k$ are constructed by (6);*
(2) *operations ${}^{\sigma}f_1, \dots, {}^{\sigma}f_k$ are obtained from $f_1, \dots, f_k$ using (2).*

Lemma 2 and Theorem 3 imply the following statement:

Theorem 4. *Let $p_1, \dots, p_k$ be 1-invertible $(n-k+1)$-ary operations and $h_1, \dots, h_k$ be k-ary orthogonal operations, $\sigma^{-1} \in S^{\delta}_{n+1}$. Then operations ${}^{\sigma}f_1, \dots, {}^{\sigma}f_n$ being constructed by composition algorithm have orthogonal δ-retracts.*

Proof. According to Theorem 3, the operations $f_1, \dots, f_k$ have orthogonal $\overline{1, k}$-retracts. By virtue of Lemma 2, the operations ${}^{\sigma}f_1, \dots, {}^{\sigma}f_k$ have orthogonal ${}^{\sigma}\overline{1, k}$-retracts. But $\sigma^{-1} \in S^{\delta}_{n+1}$, i.e., ${}^{\sigma^{-1}}\delta = \overline{1, k}$, wherefrom ${}^{\sigma}\overline{1, k} = \delta$. □

4. Block-wise recursive algorithm

The purpose of this section is to describe an algorithm for constructing an orthogonal tuple of n-ary operations from blocks of n-ary operations with orthogonal retracts. "Recursive" means that every next block of operations is constructed from all operations which have been constructed before.
Let n be an arbitrary natural number, $n \geqslant 2$. Denote classes of a partition π of the set $\overline{1, n}$ as follows

$$\pi = \{\pi_1, \pi_2 \dots, \pi_k\}.$$

Let $f_1, \dots, f_n$ be n-ary operations on Q. We distribute these operations according to partition π of their indices:

$$\{f_j \mid j \in \pi_i\}, \quad i = 1, 2, \dots, k.$$

*π**-block-wise recursive algorithm.*** *Let $\pi := \{\pi_1, \dots, \pi_k\}$ be a partition of $\overline{1, n}$ and $f_1, \dots, f_n$ be n-ary operations, $\tau_1 \in S_{\pi_1}$, $\tau_2 \in S_{\pi_1 \cup \pi_2}, \dots, \tau_{k-1} \in S_{\pi_1 \cup \dots \cup \pi_{k-1}}$.*
Operations $g_1, \dots, g_n$ are constructed by the following items

(1) *the first block of operations is*

$$g_j(x_1, \dots, x_n) := f_j(x_1, \dots, x_n), \quad j \in \pi_1;$$

(2) *for every $i = 2, \dots, k$ the i-th block of operations is*

$$g_j(x_1, \dots, x_n) := f_j(t_1, \dots, t_n), \quad j \in \pi_i,$$

where

$$t_s := \begin{cases} g_{s\tau_{i-1}}(x_1, \dots, x_n), & \text{if } s \in \pi_1 \cup \dots \cup \pi_{i-1}, \\ x_s & \text{otherwise.} \end{cases}$$

A tuple of operations $f_1, \dots, f_n$ will be called π*-block retract orthogonal* if for all $i \in \overline{1,k}$ a tuple $\{f_j \mid j \in \pi_i\}$ has orthogonal π_i-retracts.

Theorem 5. *Let operations $f_1, \dots, f_n$ be π-block retract orthogonal. Then the operations $g_1, \dots, g_n$ constructed by π-block-wise recursive algorithm are orthogonal.*

Proof. To prove the statement of the theorem we use induction on the number k of blocks of the partition π.

The basis. If $k = 1$, then $\pi = \{\pi_1\} = \{\overline{1,n}\}$. Thus, the operations $f_1, \dots, f_n$ are contained in one block, they are orthogonal according to the condition of the theorem. Consequently, for $k = 1$ the statement is true.

The inductive hypothesis. Suppose the statement of the theorem is true for $k = m$. It means that the partition has m blocks, and a tuple of operations being constructed by the block-wise recursive algorithm is orthogonal.

The inductive step. Consider the statement for $k = m+1$, i.e., the partition π has $m+1$ blocks; we have to prove that every arbitrary tuple of n-ary operations $g_1, \dots, g_n$ being constructed by some π-block-wise recursive algorithm is orthogonal. In other words, we have to prove that for all $b_1, \dots, b_n \in Q$ the system

$$\{g_j(x_1, \dots, x_n) = b_j, \quad j \in \overline{1,n} \tag{7}$$

has a unique solution.

For $k = m + 1$ the π-block-wise recursive algorithm for $g_1, \dots, g_n$ has such parameters:

- $\pi = \{\pi_1, \dots, \pi_m, \pi_{m+1}\}$;
- $f_1, \dots, f_n$ are π-block retract orthogonal operations, i.e., for every $i \in \overline{1, m+1}$ the tuple $\{f_j \mid j \in \pi_i\}$ has orthogonal π_i-retracts;
- $\tau_1 \in S_{\pi_1}, \tau_2 \in S_{\pi_1 \cup \pi_2}, \dots, \tau_m \in S_{\pi_1 \cup \dots \cup \pi_m}$.

For brevity, we provide the notation

$$\pi' := \{\pi_1, \dots, \pi_m\} = \pi \setminus \{\pi_{m+1}\}, \qquad \pi_0 := \pi_1 \cup \dots \cup \pi_m = \overline{1,n} \setminus \pi_{m+1}.$$

Let us consider a subsystem of (7)

$$\{g_j(x_1, \dots, x_n) = b_j, \quad j \in \pi_{m+1}. \tag{8}$$

Using the algorithm, we obtain

$$\begin{cases} f_j(t_1, \dots, t_n) = b_j, \quad j \in \pi_{m+1}, \\ t_s := \begin{cases} g_{s\tau_m}(x_1, \dots, x_n), & \text{if } s \in \pi_0, \\ x_s & \text{otherwise.} \end{cases} \end{cases}$$

In this system, we replace all subterms $g_{s\tau}(x_1, \dots, x_n)$, $s \in \pi_0$ with their values taken from (7):

$$\{f_j(t_1, \dots, t_n) = b_j, \quad j \in \pi_{m+1}, \tag{9}$$

where

$$t_s := \begin{cases} b_{s\tau_m}, & \text{if } s \in \pi_0, \\ x_s & \text{otherwise.} \end{cases}$$

Since τ_m is a permutation of π_0, (9) implies that the set of all x's indices is equal to

$$\overline{1,n} \setminus \pi_0 = \pi_{m+1},$$

because π is a partition of $\overline{1,n}$. Then all left sides of equations from (9) are π_{m+1}-retracts of operations f_j, $j \in \pi_{m+1}$. Since the operations $f_1, \dots, f_n$ are π-block retract orthogonal, then their π_{m+1}-retracts are orthogonal. Thus, (9) has a unique solution:

$$x_s := a_s, \qquad s \in \pi_{m+1}. \tag{10}$$

We substitute (10) in the other equations of (7), i.e., in all equations of (7) except Eqs. (8)

$$\{g_j(y_1, \dots, y_n) = b_j, \quad j \in \pi_0, \tag{11}$$

where

$$y_s := \begin{cases} x_s, & \text{if } s \in \pi_0, \\ a_s & \text{otherwise.} \end{cases}$$

In the left side, we obtain π_0-retracts of operations $g_j, j \in \pi_0$, and we follow the notation: $\{s_1, \ldots, s_\ell\} := \pi_0$,

$$\begin{aligned} &f'_j(x_{s_1}, \ldots, x_{s_\ell}) := f_j(y_1, \ldots, y_n), \\ &g'_j(x_{s_1}, \ldots, x_{s_\ell}) := g_j(y_1, \ldots, y_n) \end{aligned} \tag{12}$$

for all $j \in \pi_0$, where

$$y_s := \begin{cases} x_s, & \text{if } s \in \pi_0, \\ a_s & \text{otherwise.} \end{cases}$$

The induction hypothesis implies that the system (11) has a unique solution. To use this the following facts are to be proved:

1°. The operations $f'_j, j \in \pi_0$ are π'-block retract orthogonal;
2°. The operations $g'_j, j \in \pi_0$ are constructible by a π'-block-wise recursive algorithm.

Proof of 1°. Assumption of the theorem implies that for arbitrary $i = 1, \ldots, m$ a block of operations $f_j, j \in \pi_i$ has orthogonal π_i-retracts. This means that all similar π_i-retracts of operations $f_j, j \in \pi_i$ are orthogonal. Since operations $f'_j, j \in \pi_i$ are π_0-retracts of $f_j, j \in \pi_i$, then the set of all tuples of similar π_i-retracts of operations $f'_j, j \in \pi_i$ is a subset of the set of all tuples of similar π_i-retracts of operations $f_j, j \in \pi_i$. But if every tuple of the last set is orthogonal, then every tuple of the previous set is also orthogonal. Consequently, the operations $f'_j, j \in \pi_i$ have orthogonal π_i-retracts. Because i is arbitrary, then the operations $f'_j, j \in \pi_0$ are π'-block retract orthogonal, i.e., 1° has been proved.

Proof of 2°. Applying (10) and (12) to the first m blocks of the algorithm for $k = m + 1$, we obtain

(1) the first block of operations $g'_j(x_{s_1}, \ldots, x_{s_\ell}) = f'_j(x_{s_1}, \ldots, x_{s_\ell}), j \in \pi_1$;
(2) for every $i = 2, \ldots, m$ the i-th block of operations

$$g'_j(x_{s_1}, \ldots, x_{s_\ell}) = f'_j(t_{s_1}, \ldots, t_{s_\ell}), \quad j \in \pi_i,$$

where

$$t_s := \begin{cases} g'_{s\tau_{i-1}}(x_{s_1}, \ldots, x_{s_\ell}), & \text{if } s \in \pi_1 \cup \cdots \cup \pi_{i-1}, \\ x_s & \text{otherwise.} \end{cases}$$

According to the definition of block-wise recursive algorithm, this algorithm constructs $g'_j, j \in \pi_0$ from $f'_j, j \in \pi_0$, i.e., 2° holds.

By the induction hypothesis, we conclude that (11) has a unique solution:

$$x_s := a_s, \qquad s \in \pi_0. \tag{13}$$

Combining (10) and (13) we conclude that $(a_1, \ldots, a_n)$ is a solution of (7). This means that the operations $g_1, \ldots, g_n$ are orthogonal.

Thus, the statement of the theorem is true for all natural k, i.e., for an arbitrary number of blocks of partition π. □

Trivial recursive algorithms.

An algorithm is called *trivial recursive algorithm* if π is trivial, i.e., every block of partition π is a singleton.

Since blocks of operations are singletons, then every block contains only one operation. Retract orthogonality of an operation is its i-invertibility.

Let $\pi := \{\{i_1\}, \ldots, \{i_n\}\}$ be a partition of $\overline{1,n}$ and $f_1, \ldots, f_n$ be n-ary operations, $\tau_1 \in S_{\{i_1\}}$, $\tau_2 \in S_{\{i_1,i_2\}}, \ldots, \tau_{n-1} \in S_{\{i_1,\ldots,i_{n-1}\}}$.

Operations $g_1, \ldots, g_n$ are constructed by the following items

(1) the first operation is

$$g_{i_1}(x_1, \ldots, x_n) := f_{i_1}(x_1, \ldots, x_n);$$

(2) for every $j \in \{i_2, \ldots, i_n\}$ the j-th operation is

$$g_j(x_1, \ldots, x_n) := f_j(t_1, \ldots, t_n),$$

where

$$t_s := \begin{cases} g_{s\tau_{j-1}}(x_1, \ldots, x_n), & \text{if } s \in \{i_1, \ldots, i_{s-1}\}, \\ x_s & \text{otherwise.} \end{cases}$$

Corollary 6. *Let operations $f_1, \dots, f_n$ be i_1-, ..., i_n-invertible. Then the operations $g_1, \dots, g_n$ constructed by trivial recursive algorithm are orthogonal.*

If $i_1 = n, i_2 = n-1, \dots, i_n = 1$ and $\tau_1, \tau_2, \dots, \tau_{n-1}$ are trivial transformations, then we obtain the algorithm (3).

Corollary 7 ([3]). *Let f_i be $(n-i+1)$-invertible n-ary operations for all $i \in \overline{1,n}$. Then the operations $g_1, \dots, g_n$ constructed by (3) are orthogonal.*

The simple example given below shows existence of tuples of orthogonal operations which are constructible by block-wise recursive algorithm and non-constructible by trivial recursive algorithm.

Example 1. Consider the quadruple of operations

$$\begin{aligned}
&f_1(x,y,z,t) = 3x + 2y + 3z + 4t,\\
&f_2(x,y,z,t) = 2x + 3y + 3z + 2t,\\
&f_3(x,y,z,t) = 4x + 2y + 3z + 2t,\\
&f_4(x,y,z,t) = 2x + 2y + 2z + 3t
\end{aligned}$$

on $\mathbb{Z}_6$. Since f_1, f_2 have orthogonal $\{1,2\}$-retracts, f_3, f_4 have orthogonal $\{3,4\}$-retracts, then $\pi = \big\{\{1,2\},\{3,4\}\big\}$. There are two variants for choosing τ_1, because τ_1 is a permutation of the set $\{1,2\}$: $\tau_1 = \iota$ and $\tau_1 = (12)$, where ι is identity permutation. For each of these variants, we construct quadruple of orthogonal operations by a block-wise recursive algorithm.

Let $\tau_1 = \iota$, then π-block-wise recursive algorithm is

$$\begin{aligned}
&g_1(x,y,z,t) = 3x + 2y + 3z + 4t,\\
&g_2(x,y,z,t) = 2x + 3y + 3z + 2t,\\
&g_3(x,y,z,t) = 4(3x + 2y + 3z + 4t) + 2(2x + 3y + 3z + 2t) + 3z + 2t,\\
&g_4(x,y,z,t) = 2(3x + 2y + 3z + 4t) + 2(2x + 3y + 3z + 2t) + 2z + 3t,
\end{aligned}$$

i.e.,

$$\begin{aligned}
&g_1(x,y,z,t) = 3x + 2y + 3z + 4t,\\
&g_2(x,y,z,t) = 2x + 3y + 3z + 2t,\\
&g_3(x,y,z,t) = 4x + 2y + 3z + 4t,\\
&g_4(x,y,z,t) = 4x + 4y + 2z + 3t.
\end{aligned}$$

And let $\tau_1 = (12)$ then π-block-wise recursive algorithm is

$$\begin{aligned}
&g'_1(x,y,z,t) = 3x + 2y + 3z + 4t,\\
&g'_2(x,y,z,t) = 2x + 3y + 3z + 2t,\\
&g'_3(x,y,z,t) = 4(2x + 3y + 3z + 2t) + 2(3x + 2y + 3z + 4t) + 3z + 2t,\\
&g'_4(x,y,z,t) = 2(2x + 3y + 3z + 2t) + 2(3x + 2y + 3z + 4t) + 2z + 3t,
\end{aligned}$$

i.e.,

$$\begin{aligned}
&g'_1(x,y,z,t) = 3x + 2y + 3z + 4t,\\
&g'_2(x,y,z,t) = 2x + 3y + 3z + 2t,\\
&g'_3(x,y,z,t) = 2x + 4y + 3z + 0t,\\
&g'_4(x,y,z,t) = 4x + 4y + 2z + 3t.
\end{aligned}$$

Trivial recursive algorithm requires that at least one of the constructed operations must be i-invertible for some $i \in \{1,2,3,4\}$. Each of the constructed quadruples cannot be constructed by trivial recursive algorithm, because any of the operations from tuples g_1, g_2, g_3, g_4 and g'_1, g'_2, g'_3, g'_4 is not i-invertible for all $i \in \{1,2,3,4\}$.

5. Block composition algorithm

In this section, we give an algorithm for constructing orthogonal n-ary operations from operations which are distributed into k blocks. For all $i \in \overline{1,k}$, every ith block contains n_i-tuple of n_i-ary orthogonal operations and $n_1 + n_2 + \dots + n_k = n$. This algorithm is a composition of the algorithms given above, namely a composition algorithm and a block-wise recursive algorithm.

Now we describe this algorithm in detail.

*π-**block composition algorithm.** Let $\pi = \{\pi_1, \dots, \pi_k\}$be a partition of the set $\overline{1,n}$ and let for all $i = 1, \dots, k$, the following conditions*

– $h_j, j \in \pi_i$ be $|\pi_i|$*-ary operations,*
– $p_j, j \in \pi_i$ be $(n - |\pi_i| + 1)$*-ary operations,*

- $\sigma_i \in S_{n+1}^{\pi_i}$,
- $\tau_{i-1} \in S_{\pi_1\cup\cdots\cup\pi_{i-1}}$, $i > 1$

hold.

Operations $g_1, \ldots, g_n$ are constructed by the following items

(1) *operations f_j are constructed by (6) for all $j \in \pi_i$, $i = 1, \ldots, k$:*

$$f_j(x_1, \ldots, x_n) := p_j\big(h_j(x_1, \ldots, x_{|\pi_i|}), x_{|\pi_i|+1}, \ldots, x_n\big);$$

(2) *parastrophes ${}^{\sigma_i}\!f_j$, $j \in \pi_i$ are formed from operations f_j, $i = 1, \ldots, k$;*

(3) *operations $g_1, \ldots, g_n$ are constructed by block-wise recursive algorithm:*

(a) *the first block of operations is*

$$g_j(x_1, \ldots, x_n) := {}^{\sigma_1}\!f_j(x_1, \ldots, x_n), \quad j \in \pi_1,$$

(b) *for every $i = 2, \ldots, k$, the i-th block of operations is*

$$g_j(x_1, \ldots, x_n) := {}^{\sigma_i}\!f_j(t_1, \ldots, t_n), \quad j \in \pi_i,$$

where

$$t_s := \begin{cases} g_{s\tau_{i-1}}(x_1, \ldots, x_n), & \text{if } s \in \pi_1 \cup \cdots \cup \pi_{i-1}, \\ x_s & \text{otherwise.} \end{cases}$$

Theorem 8. *Let $\pi = \{\pi_1, \ldots, \pi_k\}$ be a partition of $\overline{1, n}$ and for all $i = 1, \ldots, k$, the following conditions*

- *$h_j, j \in \pi_i$ be $|\pi_i|$-ary orthogonal operations,*
- *$p_j, j \in \pi_i$ be 1-invertible $(n - |\pi_i| + 1)$-ary operations*

hold. Then n-ary operations $g_1, \ldots, g_n$ constructed by block composition algorithm are orthogonal.

Proof. Let conditions of the theorem be satisfied. Consider the proof according to the items of block composition algorithm.

1. By virtue of Theorem 3, operations $f_j, j \in \pi_i$ have orthogonal $\{1, \ldots, |\pi_i|\}$-retracts for all $i \in \overline{1, k}$.

2. Since $\{1, \ldots, |\pi_i|\}\sigma_i^{-1} = \pi_i$, then the operations ${}^{\sigma_i}\!f_j, j \in \pi_i$ have orthogonal π_i-retracts by Lemma 2 for all $i \in \overline{1, k}$.

3. Since for all $i = 1, \ldots, k$, the operations ${}^{\sigma_i}\!f_j, j \in \pi_i$ have orthogonal π_i-retracts, we can apply block-wise recursive algorithm to them. According to Theorem 5, the operations $g_1, \ldots, g_n$ are orthogonal. □

Remark 1. If block π_i is a singleton, then it consists of j. Hence p_j is 1-invertible n-ary operation and h_j is a unary quasigroup, i.e., a permutation of the carrier. Thus operation f_j defined by

$$f_j(x_1, \ldots, x_n) := p_j(h_j(x_1), x_2, \ldots, x_n)$$

is isotopic to p_j. The parameter σ_i is a cycle $(1, j)$.

Each of the algorithms describes a series of algorithms. Generally speaking even for trivial classes of operations, every choice of parameters of the algorithm gives different tuples of orthogonal operations which are not necessarily parastrophic. We consider this in the following example.

Tuple of operations ${}^{\sigma}\!f_1, \ldots, {}^{\sigma}\!f_n$ is called *σ-parastrophic* to tuple $f_1, \ldots, f_n$, $\sigma \in S_{n+1}$.

Example 2. Let $\pi = \big\{\{1, 2\}, \{3, 4\}\big\}$, $\mathbb{Z}_6$ be a carrier and the pairs of operations h_1, h_2 and h_3, h_4, which are defined by

$$h_1(x_1, x_2) = 3x_1 + 2x_2, \quad h_3(x_1, x_2) = 3x_1 + 2x_2,$$
$$h_2(x_1, x_2) = 2x_1 + 3x_2, \quad h_4(x_1, x_2) = 2x_1 + 3x_2,$$

be orthogonal, the operations

$$p_1(u, x_3, x_4) = u + 2x_3 + 2x_4,$$
$$p_2(u, x_3, x_4) = u + 3x_3 + 4x_4,$$
$$p_3(u, x_3, x_4) = u + 4x_3 + 2x_4,$$
$$p_4(u, x_3, x_4) = u + 2x_3 + 2x_4$$

be 1-invertible, $\sigma_1 \in S_5^{\{1,2\}}$, $\sigma_2 \in S_5^{\{3,4\}}$, where

$$S_5^{\{1,2\}} = \{\iota,\ (12),\ (34),\ (12)(34)\},$$
$$S_5^{\{3,4\}} = \{(13)(24),\ (14)(23),\ (1324),\ (1423)\},$$

and $\tau_1 \in S_{\{1,2\}}$, where $S_{\{1,2\}} = \big\{\iota,\ (12)\big\}$.

Let us construct orthogonal operations by a block composition algorithm.

By p.1 from the given operations, we construct operations with orthogonal retracts:

$$B_1 : \begin{cases} f_1(x_1, x_2, x_3, x_4) := p_1(h_1(x_1, x_2), x_3, x_4) = 3x_1 + 2x_2 + 2x_3 + 2x_4, \\ f_2(x_1, x_2, x_3, x_4) := p_2(h_2(x_1, x_2), x_3, x_4) = 2x_1 + 3x_2 + 3x_3 + 4x_4, \end{cases}$$

$$B_2 : \begin{cases} f_3(x_1, x_2, x_3, x_4) := p_3(h_3(x_1, x_2), x_3, x_4) = 3x_1 + 2x_2 + 4x_3 + 2x_4, \\ f_4(x_1, x_2, x_3, x_4) := p_4(h_4(x_1, x_2), x_3, x_4) = 2x_1 + 3x_2 + 2x_3 + 2x_4. \end{cases}$$

By p.2, we apply permutations σ_1, σ_2 to the corresponding blocks of operations. There exist sixteen different cases. To show dependence among different tuples, we consider two of them. For example, if $\sigma_1 = \iota$, $\sigma_2 = (13)(24)$ and if $\sigma_1 = (12)$, $\sigma_2 = (23)(14)$.

If $\sigma_1 = \iota, \sigma_2 = (13)(24)$, then

$$B_1 : \begin{cases} {}^{\sigma_1}f_1(x_1, x_2, x_3, x_4) = 3x_1 + 2x_2 + 2x_3 + 2x_4, \\ {}^{\sigma_1}f_2(x_1, x_2, x_3, x_4) = 2x_1 + 3x_2 + 3x_3 + 4x_4, \end{cases}$$

$$B_2 : \begin{cases} {}^{\sigma_2}f_3(x_1, x_2, x_3, x_4) = 4x_1 + 2x_2 + 3x_3 + 2x_4, \\ {}^{\sigma_2}f_4(x_1, x_2, x_3, x_4) = 2x_1 + 2x_2 + 2x_3 + 3x_4. \end{cases}$$

If $\sigma_1 = (12), \sigma_2 = (23)(14)$, then

$$B_1 : \begin{cases} {}^{\sigma_1}f_1(x_1, x_2, x_3, x_4) = 2x_1 + 3x_2 + 2x_3 + 2x_4, \\ {}^{\sigma_1}f_2(x_1, x_2, x_3, x_4) = 3x_1 + 2x_2 + 3x_3 + 4x_4, \end{cases}$$

$$B_2 : \begin{cases} {}^{\sigma_2}f_3(x_1, x_2, x_3, x_4) = 2x_1 + 4x_2 + 2x_3 + 3x_4, \\ {}^{\sigma_2}f_4(x_1, x_2, x_3, x_4) = 2x_1 + 2x_2 + 3x_3 + 2x_4. \end{cases}$$

By p.3, we apply block-wise recursive algorithm when $\tau_1 = \iota$.

If $\sigma_1 = \iota, \sigma_2 = (13)(24), \tau_1 = \iota$, then

$$B_1 : \begin{cases} g_1(x_1, x_2, x_3, x_4) = 3x_1 + 2x_2 + 2x_3 + 2x_4, \\ g_2(x_1, x_2, x_3, x_4) = 2x_1 + 3x_2 + 3x_3 + 4x_4, \end{cases}$$

$$B_2 : \begin{cases} g_3(x_1, x_2, x_3, x_4) = 4x_1 + 2x_2 + 5x_3 + 0x_4, \\ g_4(x_1, x_2, x_3, x_4) = 4x_1 + 4x_2 + 0x_3 + 3x_4. \end{cases}$$

If $\sigma_1 = (12), \sigma_2 = (23)(14), \tau_1 = \iota$, then

$$B_1 : \begin{cases} g_1(x_1, x_2, x_3, x_4) = 2x_1 + 3x_2 + 2x_3 + 2x_4, \\ g_2(x_1, x_2, x_3, x_4) = 3x_1 + 2x_2 + 3x_3 + 4x_4, \end{cases}$$

$$B_2 : \begin{cases} g_3(x_1, x_2, x_3, x_4) = 4x_1 + 2x_2 + 0x_3 + 5x_4, \\ g_4(x_1, x_2, x_3, x_4) = 4x_1 + 4x_2 + x_3 + 2x_4. \end{cases}$$

The constructed tuples of orthogonal operations are different and by definition they are not parastrophic. There are thirty-two possible tuples, i.e.,

$$|S_5^{\{1,2\}}| \cdot |S_5^{\{3,4\}}| \cdot |S_{\{1,2\}}| = 32.$$

The block composition algorithm is convenient for implementation. The following example illustrates the construction by the block composition algorithm.

Example 3. Let $\pi = \big\{\{2\}, \{1, 3, 4\}\big\}$, $\mathbb{Z}_{15}$ be a carrier and the operations

$$\begin{aligned} &p_1(x_1, x_2, x_3, x_4) = x_1 + 4x_2 + 2x_3 + x_4, \\ &p_2(x_3, x_4) = x_3 + 5x_4, \\ &p_3(x_3, x_4) = 4x_3 + x_4, \\ &p_4(x_3, x_4) = x_3 + 11x_4 \end{aligned}$$

be 1-invertible, the operation $h_1(x_1) = 7x$ be a permutation, the operations

$$\begin{aligned} &h_2(x_1, x_2, x_3) = 3x_1 + 2x_2 + 6x_3, \\ &h_3(x_1, x_2, x_3) = 2x_1 + 3x_2 + 3x_3, \\ &h_4(x_1, x_2, x_3) = 2x_1 + x_2 + 2x_3 \end{aligned}$$

be orthogonal, $\sigma_1 = (12), \sigma_2 = (24), \tau_1 = \iota$.

We construct orthogonal operations from the given operations for partition π by the block composition algorithm.

By p.1 according to (6), we construct operations f_1, f_2, f_3, f_4:

$$B_1 : \Big\{ f_1(x_1, x_2, x_3, x_4) := p_1(h_1(x_1), x_2, x_3, x_4) = 7x_1 + 4x_2 + 2x_3 + x_4,$$

$$B_2 : \begin{cases} f_2(x_1, x_2, x_3, x_4) := p_2(h_2(x_1, x_2, x_3), x_4) = 3x_1 + 2x_2 + 6x_3 + 5x_4, \\ f_3(x_1, x_2, x_3, x_4) := p_3(h_3(x_1, x_2, x_3), x_4) = 8x_1 + 12x_2 + 12x_3 + x_4, \\ f_4(x_1, x_2, x_3, x_4) := p_4(h_4(x_1, x_2, x_3), x_4) = 2x_1 + x_2 + 2x_3 + 11x_4. \end{cases}$$

By p.2, we apply permutation σ_1 to the first block:

$$^{\sigma_1}f_1(x_1, x_2, x_3, x_4) = 4x_1 + 7x_2 + 2x_3 + x_4$$

and σ_2 to the second block:

$$\begin{aligned} &^{\sigma_2}f_2(x_1, x_2, x_3, x_4) = 3x_1 + 5x_2 + 6x_3 + 2x_4, \\ &^{\sigma_2}f_3(x_1, x_2, x_3, x_4) = 8x_1 + x_2 + 12x_3 + 12x_4, \\ &^{\sigma_2}f_4(x_1, x_2, x_3, x_4) = 2x_1 + 11x_2 + 2x_3 + x_4. \end{aligned}$$

By virtue of Lemma 2, the operation $^{\sigma_1}f_2$ is 2-invertible, $^{\sigma_2}f_2$, $^{\sigma_2}f_3$, $^{\sigma_2}f_4$ have orthogonal $\{1, 3, 4\}$-retracts.

By p.3, we write orthogonal operations g_1, g_2, g_3, g_4 for the given partition π:

$$\begin{aligned} &g_1(x_1, x_2, x_3, x_4) = 7x_1 + 4x_2 + 2x_3 + x_4, \\ &g_2(x_1, x_2, x_3, x_4) = 3x_1 + 5(7x_1 + 4x_2 + 2x_3 + x_4) + 6x_3 + 2x_4, \\ &g_3(x_1, x_2, x_3, x_4) = 8x_1 + (7x_1 + 4x_2 + 2x_3 + x_4) + 12x_3 + 12x_4, \\ &g_4(x_1, x_2, x_3, x_4) = 2x_1 + 11(7x_1 + 4x_2 + 2x_3 + x_4) + 2x_3 + x_4, \end{aligned}$$

i.e.,

$$\begin{aligned} &g_1(x_1, x_2, x_3, x_4) = 7x_1 + 4x_2 + 2x_3 + x_4, \\ &g_2(x_1, x_2, x_3, x_4) = 8x_1 + 5x_2 + x_3 + 7x_4, \\ &g_3(x_1, x_2, x_3, x_4) = 0x_1 + 4x_2 + 14x_3 + 13x_4, \\ &g_4(x_1, x_2, x_3, x_4) = 4x_1 + 14x_2 + 9x_3 + 12x_4. \end{aligned}$$

According to the block composition algorithm, the operations g_1, g_2, g_3, g_4 are orthogonal.

Remark 2. Article [1] implies that any subtuple of operations from tuple of orthogonal operations is orthogonal. In [3] it is proved that any k-tuple of orthogonal n-ary operations can be embedded into some orthogonal n-tuple of n-ary operations. Therefore without recourse to detailed proof, we claim that the block-wise recursive algorithm gives possibility to construct ℓ-tuple of orthogonal n-ary operations, $\ell \leqslant n$.

Conclusions. A partition of the set $\overline{1, n}$ is one of the parameters of block-wise recursive algorithm for constructing orthogonal n-ary operations. The algorithm is called trivial if the partition is trivial and non-trivial otherwise. A tuple of orthogonal operations being constructible only by non-trivial block-wise recursive algorithm is exemplified. But the obtained algorithm should be further investigated. For example, the following questions should be answered:

1. What part of tuples of orthogonal n-ary operations is constructible by the algorithm?
2. Under what conditions does the algorithm construct a) different tuples of operations; b) a tuple of quasigroups; c) a strong orthogonal tuple of quasigroups and etc.?

Acknowledgment

The authors are grateful to Vira Obshanska for her help and advice in the process of writing the English text of this article.

References

[1] G. Belyavskaya, Pairwise ortogonality of n-ary operations, Bul. Acad. Ştiinţe Repub. Mold. Mat. 3 (2005) 5–18.

[2] G. Belyavskaya, Secret-sharing schemes and orthogonal systems of k-ary operations, Quasigroups Related Systems 17 (2009) 161–176.

[3] G. Belyavskaya, G. Mullen, Orthogonal hypercubes and n-ary operations, Quasigroups Related Systems 13 (1) (2005) 73–86.

[4] E. Couselo, S. Gonzalez, V. Markov, A. Nechaev, Recursive MDS-codes and recursively differentiable quasigroups, Discrete Math. Appl. 8 (3) (1998) 217–246. http://dx.doi.org/10.1515/dma.1998.8.3.217.

[5] J. Denes, A. Keedwell, Latin Squares and their Applications, Academiai Kiado, Budapest, 1974.

[6] S. Dougherty, T. Szczepanski, Latin k-hypercubes, Australas. J. Combin. 40 (2008) 145–160.

[7] I. Fryz, On a construction of orthogonal operations, in: Proceedings of the Third Conference of Mathematical Society of the Republic of Moldova, 19–23 Aug. 2014, Chishinau, Moldova, 2014.

[8] V. Izbash, P. Syrbu, Recursively differentiable quasigroups and complete recursive codes, Comment. Math. Univ. Carolin. 45 (2) (2004) 257–263.

[9] P. Syrbu, Orthogonal and Self-orthogonal n-Operations (Ph.D. thesis), Academy of Science of Moldova SSR, 1990 (in Russian).