Theoretical and Applied Informatics
ISSN 1896–5334
Vol.24 (2012), no. 2
pp. 119–133
DOI: 10.2478/v10179-012-0006-3

Representation of direct product of permutation groups as symmetry groups of boolean functions

PAWEŁ JASIONOWSKI

Faculty of Applied Mathematics
Silesian University of Technology
ul. Kaszubska 23, 44-100 Gliwice, Poland
e-mail: pawel.jasionowski@polsl.pl

Received 18 April 2012, Revised 4 June 2012, Accepted 22 June 2012.

Abstract: We consider boolean functions invariance groups $S(f)$ and some special kinds of boolean functions. We construct a k valued boolean function f which represent a direct product of two permutation groups and give an upper bound of k. Moreover we show that in some cases we can construct 2 valued boolean function which represent this product.

1. Introduction

We consider a module M with n possible input states, each of which can assume one of two possible states 0 or 1. Moreover module M on output can assume 0 or 1 too (we also consider a generalization of this device by putting more then two possible output states). The starting point of this paper is [3] where basic structures and some specific constructions of boolean functions are given. We consider a problem of placement of such modules on the integrated circuit (or chip) where permutation of inputs is allowed or when order of input values is not given, or is partially given. Following [3] we believe that studies of the symmetry groups of boolean functions may lead to algorithms optimization the space in VLSI technology. It help answer for question that we can changed a place of certain module in block without changing an output value of computed function.

The main objects of study of this paper are symmetry groups of boolean functions. We want to provide some algorithms to describe special kinds of permutation groups.

This paper show how we can construct a boolean function for direct product of permutation groups. In the last section we show that we can give an exact construction of 2 valued boolean function when we consider some direct product of symmetric groups.

2. Preliminaries

2.1. Basic definition

A set of all boolean vectors is denoted as a $\{0,1\}^n$. Let $f : X^n \to Y$, $f(x_1, x_2, ..., x_n) = y$ is a function with n variables ($n \geq 1$).

The set $S(f)$ of all permutation $\sigma \in S_n$ such that

$$f(x_1, x_2, ..., x_n) = f(x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(n)})$$

for every $(x_1, x_2, ..., x_n) \in X^n$ is a subgroup of the symmetric group S_n and is called the boolean function invariance group or symmetry group of function f. Let f is k-valued boolean function ($k \geq 1$) such that:

$$f : \{0,1\}^n \to \{0, 1, ..., k-1\}$$

A permutation group $G \leq S_n$, such that $G = S(f)$, for some k-valued boolean function f is called a group which is representable by f (or k-representable). Then function f is called invariant at the group G.

We consider a group $G \leq S_n$. To check that a group G is representable, we must check, how G act on the set $X := \{0,1\}^n$. We consider the action given by:

$$x \to x^\sigma : (x_1, x_2, ..., x_n) \to (x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(n)}) \tag{1}$$

An orbit of element $x \in X$ is defined by:

$$x^G = \{x^\sigma, \sigma \in G\}$$

Let $G = S(f)$ for some boolean function $f : \{0,1\}^n \to \{0, 1, ..., k-1\}$. If elements $x, y \in X$ are in the same orbit, then $f(x) = f(y)$ and for every permutation $\tau \notin G$ there must exist $x \in X$, such that x i x^τ are in the different orbits and $f(x) \neq f^\tau(x)$.

This situation take places, because, if there exists a permutation $\tau \in G$, such that for every $x \in X$ elements x and x^τ are always in the same orbit, then the group G and $G' = \langle G, \tau \rangle$ has exactly the same orbits in X. So every function $f : \{0,1\}^n \to \{0, 1, ..., k-1\}$ invariant at the group G is invariant at the group G'. Now we can see that G can not be k-representable for any k.

2.2. Integrated circuits and optimization

Let M is a module (or a device) with n possible inputs, each of which can assume one of two states 0 or 1. In general we consider an module with k possible outputs from the set $\{0, 1, ..., k-1\}$, $k > 1$. We can understand input values as a boolean vector $(x_1, x_2, ..., x_n)$. A output of module depend in general on the order of $x_1, x_2, ..., x_n$. There exist some permutations of inputs which leave output invariant. For example if output of module M do not depend on order of $x_1, x_2, ..., x_n$ then we call such a module symmetric. In general set of all permutation which do not change output in any initial state create a group.

Let there are two possible output states of module which we denote as 0 and 1. Operation on inputs by the module M can be represent by boolean function $f : \{0, 1\}^n \to \{0, 1\}$. If we understand inputs as a boolean vector $(x_1, x_2, ..., x_n)$ then $f(x_1, x_2, ..., x_n)$ is an output value of M. So let boolean function f represents M. A set $S(f)$ of all permutation σ of $\{1, 2, ..., n\}$ such, that for every inputs $x_i \in \{0, 1\}, i = 1, 2, ..., n$ we have

$$f(x_1, x_2, ..., x_n) = f(x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(n)})$$

create subgroup of S_n. So the module is called symmetric if $S(f) = S_n$.

A starting point of this research were [3] and [4] where many basic concepts and new techniques of study of symmetric groups of boolean function are given. One of the problem mentioned there is the problem of placement of such a module on the integrated circuit, where permutations of inputs are permitted, or its order is only partially-given. We expect that research on symmetry groups of boolean function can provide algorithms to optimizing VLSI technology. For example it can help arrange place of module on the chip and answer where we can put this module without changing of output value.

2.3. Anonymous network

In [4] authors show that boolean function invariance groups are important to computing values of functions in anonymous network. When we say anonymous network we think about network which hold following conditions: (1) processors known topology and size (all number of processors) of network; (2) processors do not known their identity and identity of other processors; (3) processors are identical – use the same algorithm; (4) processors are deterministic; (5) network does not need synchronization by outside device; (6) network is labeled; (7) network connections are First In First Out – type.

For example we want to compute value of n-ary boolean function n in the network with n nodes. To compute this value for inputs $x_1, x_2, ..., x_n$ processors $p_1, p_2, ..., p_n$ are initialized with inputs $x_1, x_2, ..., x_n$ respectively. By exchanging messages through the links all processors must compute the same bit which we understand as $f(x_1, x_2, ..., x_n)$.

There exist research ([1]) where the main purpose is define and investigate networks which hold following condition:

$$f \text{ is computable } \Leftrightarrow Aut(N) \subseteq S(f)$$

where $Aut(N)$ is a automorphism group of this network.

2.4. Complexity of formal languages

Many of fundamental questions in complexity theory can be formulated as a problem in formal language. By language L we mean a set of finite words over finite alphabet, for example $\Sigma = \{0, 1\}$. The basic problem to investigate is to decide how much resources like time, memory, number of processors we need to compute whether $x \in L$ for any word x over alphabet Σ. If for any n function f_L is a characteristic function of language $L \subseteq \{0, 1\}^n$ (i.e. for element from $x \in L$ function $f(x) = 1$ and 0 otherwise) then above question could be show as a problem of determine needed resources to compute value of boolean function f_L for any x and n. We define here a symmetry group $S(L)$ of language L as a symmetry group of f_L.

Considerations of symmetry groups of boolean functions can be used to analysis properties such languages. A results in [3] and [4] relate to palindrome languages and regular languages (languages which could be recognized by finite automatas). We often try to construct language for given permutation group G which L is invariant i.e.

$$G = S(L), \text{ where } L \subseteq \{0, 1\}^n$$

This problem is equivalent to construct a symmetry group $S(f)$ of characteristic function f for L. Moreover significant problem is to construct an algorithm which create invariant permutation group of language L.

2.5. Circuits with n nodes

We consider a circuit α_n with n nodes. We can understand it as a direct graph with labeled nodes $x_1, x_2, ..x_n$ (inputs) and $\vee, \wedge, \neg$ (gates). Input nodes are of in-degree equal 0 (0 input edges) and there exists exactly one output node with out-degree equal 0 (0 output edges). Complexity $c(\alpha_n)$ of this circuit is equal to number of all internal (non input or output) nodes. Depth $d(\alpha_n)$ of this circuit is equal to maximal length of path from input node to output node.

A word $x \in \{0, 1\}^n$ is acceptable (or recognized) by the circuit α_n if we can choose labels of x_i of input nodes that x_i has equal value to i-th bit of word x. A language $L \subseteq \{0, 1\}^n$ (or its characteristic function f) is recognized by α_n if for all words $x \in \{0, 1\}^n$ following condition is hold:

$$x \in L_n(\text{ or } f(x) = 1) \Leftrightarrow \alpha_n \text{ recognized } x$$

For boolean function f we define a it's complexity as a

$$c(f) = min\{(c(\alpha_n)), \alpha_n \text{ recognize } f\}$$

For example for any symmetric boolean function $f \in B_n$ we have $c(f) = O(n)$. Knowledge about symmetric groups of boolean function could be used to time or memory optimization such circuits during the process of checking if a word x is recognized by device. We can notice here that if we know a word x recognized by α_n then work x' is recognized too if there exist $\sigma \in S(L)$ that $x' = \sigma(x)$. Moreover such a research could help to investigate techniques to new network and circuit design by using symmetry groups $S(f)$ of characteristic function of language $L \in \{0, 1\}$.

3. Upper bound for k representability of direct product of permutation group

The main point of this section is to construct boolean functions which represent some direct products of groups. We build exact examples of boolean functions and show why these functions represent above classes of permutation groups. This approach show us which feature of boolean functions are important from that point of view.

We consider here permutations which preserve division of the set of all inputs of module M. We divide inputs on m equinumerous sets A_i. Considered permutations act in the following way: permutation of inputs inside the blocks A_i are allowed, but in the same way in each of them. Moreover changing places of blocks are allowed. In is possible because blocks has the same length.

From mathematical point of view we can characterize this action in the following way: we consider a direct product of permutation groups. So, let (G, X) and (H, Y) are permutation groups where $|X| = n$, $|Y| = m$. We take a group $(G \times H, X \times Y)$. An action on the set $X \times Y$ is given by the rule

$$(x, y)^{(g,h)} = (x^g, y^h)$$

where $g \in G, h \in H$. This action can be thought as an action on the set $A = \{1, 2, ..., nm\}$. This set is divided into sets $A_i = \{(i-1)n+1, ..., in\}$, for $i = 1, 2, ..., m$. Group G act inside each A_i in the same way, H act on the indexes of A_i.

First, we want to construct a boolean function $f : \{0, 1\}^{nm} \to \{0, 1, ..., k\}$ which represent $G \times H$ when we only know that G and H are k_1, k_2 representable respectively. In this general case we can show only upper bound of k.

Theorem 1 *Let $k_1 > 1, k_2 > 1$ are any positive integer number and $|X| = n, |Y| = m$. For any permutation groups (G, X) and (H, Y) where G is k_1 representable and H is k_2 representable there exist a boolean function f with at most $k_1 + k_2 + 2$ values such, that*

$$S(f) = G \times H$$

Proof So let $G = S(g)$, $g : \{0,1\}^n \to \{0,1,...,k_1 - 1\}$ and $H = S(h)$, $h : \{0,1\}^m \to \{0,1,...,k_2 - 1\}$.

Algorithm of construction boolean function:

Step 1: We create operator: $^C : \{0,1\}^n \to \{0,1\}^{nm}, \underline{x} \to \underline{x}^C$ in the following way:

input: vector $\underline{x} = (x_1, x_2, ..., x_n)$;

from $i = 1$ **to** $i < m + 1$ make a copy of vector $\underline{x}$;

output: vector $\underline{x}^C = (x_1, x_2..., x_n, x_1, x_2, ..., x_n, ..., x_1, x_2, ..., x_n)$ of length nm;

Step 2: We create second operator $^E : \{0,1\}^m \to \{0,1\}^{nm}, \underline{x} \to \underline{x}^E$ in the following way:

input: vector $\underline{x} = (x_1, x_2, ..., x_m)$;

from $i = 1$ **to** $i < m + 1$ make n copies of coordinate x_i

output: vector $\underline{x}^E = (x_1, x_1..., x_1, x_2, x_2, ..., x_2, ..., x_m, x_m, ...x_m)$ of length nm

Step 3: We construct sets

$$Oc = \{\underline{x}_i \in \{0,1\}^n, \underline{x}_i = \underbrace{0...0\overset{i}{1}0...0}_{n}, g(\underline{x}_i) = 0, i = 1, ..., n\}$$

$$Ob = \{\underline{x}_i \in \{0,1\}^m, \underline{x}_i = \underbrace{1...1\overset{i}{0}1...1}_{m}, h(\underline{x}_i) = 0, i = 1, ..., m\}$$

where g, h are boolean functions which represent G and H respectively. Moreover $Oc^C = \{\underline{x}^C, \underline{x} \in Oc\}$ and $Ob^E = \{\underline{x}^E, \underline{x} \in Ob\}$.

Step 4: We create a boolean function f

$$f(\underline{x}) = \begin{cases} g(\underline{x}_0) & \underline{x} = \underline{x}_0^C, \underline{x}_0 \in \{0,1\}^n \backslash Oc \\ h(x_0) + k_1 & \underline{x} = \underline{x}_0^E, x_0 \in \{0,1\}^m \backslash Ob, \underline{x}_0 \neq 0^m, \underline{x}_0 \neq 1^m \\ k_1 + k_2 & \underline{x} = Oc^C \\ k_1 + k_2 + 1 & \underline{x} \in Ob^E \\ 0 & \text{otherwise} \end{cases}$$

Now we show that $G \times H = S(f)$.

($\subseteq$) Let $A_1 = \{1, 2, ..., n\}, A_2\{n + 1, n + 2, ..., 2n\}...A_m\{nm - n + 1, nm - n + 2, ..., nm\}$. Each permutation $\sigma \in G \times H$ can be thought as a pair (ρ, τ) where $\rho \in G$ act inside blocks A_i (in the same way in each block) and $\tau \in H$ act on the indexes A_i. If vector $\underline{x} \in \{0,1\}^{nm}$ is in the form $\underline{x} = \underline{x}_0^C$ and $\underline{x}_0 \in \{0,1\}^n \backslash Oc$ then

$$f(\underline{x}) = f(\underline{x}_0^C) = g(\underline{x}_0) = g^\rho(\underline{x}_0) = f^\sigma(\underline{x}_0^C) = f^\sigma(\underline{x})$$

If vector $\underline{x} \in \{0,1\}^{nm}$ is in the form $\underline{x} = \underline{x}_0^E$ and $\underline{x}_0 \in \{0,1\}^m \backslash Ob$, $\underline{x}_0 \neq 0^m, \underline{x}_0 \neq 1^m$ then we have

$$f(\underline{x}) = f(\underline{x}_0^E) = h(\underline{x}_0) = h^\tau(\underline{x}_0) = f^\sigma(\underline{x}_0^E) = f(\underline{x})$$

If $\underline{x} \in Oc^C$ then $\underline{x}^\sigma \in Oc^C$ because G is k_1 representable group so for $\underline{x} = \underline{x}_0^C$ we have $g(\underline{x}_0) = 0 = g^\rho(\underline{x}_0)$. Similarly if $\underline{x} \in Ob^E$ then $\underline{x}^\sigma \in Ob^E$ because of representability of group H. If vector $\underline{x}$ is "otherwise" type from the definition of function f then $\underline{x}^\sigma$ is the same type. Moreover we can notice that the set Oc^E there is m coordinates equal 1 in the each vector but in the set Ob^E there is $(m-1)n$ such coordinates in each vector. This numbers are different when $n > 2$ or $m > 2$ because $(m-1)n = m$ is equivalent to equation $\frac{1}{n} + \frac{1}{m} = 1$.
($\supseteq$) Let $\sigma \notin G \times H$. We have following possibilities
(a) permutation σ act only inside blocks A_i, $i = 1, 2, ..., m$ but there exist two of them where σ act in a different way (possibly $\sigma(A_i) = A_j$ and $i \neq j$ but this situation does not change much) i.e. there exist m', m'' such, that σ act in the different way in $A_{m'}$ and $A_{m''}$. So there exist $j \in \{1, 2, ..., n\}$ such, that $\sigma((m'-1)n + j) = (m'-1)n + p'$, $\sigma((m''-1)n + j) = (m''-1)n + p''$ and $p' \neq p''$. In that situation we take a vector $\underline{x} = \underline{x}_0^C$ where $\underline{x}_0 = \underbrace{0...0\overset{j}{1}0...0}_{n}$. Then $f(\underline{x}) \neq 0$ and $f^\sigma(\underline{x}) = 0$.
(b) there exists positive integer i such that $\sigma(A_i) \neq A_j$, for $j = 1, 2, ..., m$. Then we take a vector $\underline{x} = \underline{x}_0^E$ where $\underline{x}_0 = \underbrace{1...1\overset{i}{0}1...1}_{m}$. We have two possibilities: $f(\underline{x}) = h(\underline{x}_0)$ or $f(\underline{x}) = k_1 + k_2 + 1$ so $f(\underline{x}) \in \{k_1, k_1+1, ..., k_1+k_2-1\} \cup \{k_1+k_2+1\}$ but $f^\sigma(\underline{x}) = 0$ or $f^\sigma(\underline{x}) = g(\underline{x}_0)$ so $f^\sigma(\underline{x}) \in \{0, 1, ..., k_1 - 1\}$. Then we have $f^\sigma(\underline{x}) \neq f(\underline{x})$.
(c) permutation $\sigma \notin G \times H$ but $\sigma \in S_n \times S_m$. As before we can understand σ as a pair (ρ, τ) so we have that $\rho \notin G$ or $\tau \notin H$.
If $\rho \notin G$ there exist vector $\underline{x}_0 \in \{0,1\}^n$ such, that $g(\underline{x}_0) \neq g^\rho(\underline{x}_0)$. If $\underline{x}_0 \notin Oc$ and $\underline{x}_0^\rho \notin Oc$ then

$$f(\underline{x}_0^C) = g(\underline{x}_0) \neq g^\rho(\underline{x}) = f^\sigma(\underline{x}_0^C)$$

If $\underline{x}_0 \in Oc$ or $\underline{x}_0^\rho \in Oc$ then it is easy to see that $f(\underline{x}_0^C) \neq f^\sigma(\underline{x}_0^C)$.
If $\tau \notin H$ there exist $\underline{x}_0 \in \{0,1\}^m$ such, that $h(\underline{x}_0) \neq h^\tau(\underline{x}_0)$. When $\underline{x}_0 \notin Ob$ and $\underline{x}_0^\tau \notin Ob$ then

$$f(\underline{x}_0^E) = f(\underline{x}_0) \neq h^\tau(\underline{x}_0) = f^\sigma(\underline{x}_0^E)$$

As before if $\underline{x}_0 \in Ob$ or $\underline{x}_0^\tau \in Ob$ then it is easy to see that $f(\underline{x}_0^E) \neq f^\sigma(\underline{x}_0^E)$ $\square$

4. Direct product of symmetric groups

The theorem from previous section show, that there exist boolean function f which represent direct product $G \times H$, but it give us only upper bound on number of possible values of f. In some cases we can construct a 2 valued boolean function which represent direct product of groups.
Now we take a direct product of k symmetric groups

$$S(f) = S_{n_1} \times S_{n_2} \times ... \times S_{n_k}$$

in the case when we have different length of blocks.

Theorem 2 *For any positive integer $n_1, n_2, ..., n_k$ such that $n_i \neq n_j$, $i \neq j$ there exist a 2 valued boolean function f such that*

$$S(f) = S_{n_1} \times S_{n_2} \times ... \times S_{n_k}$$

Proof
Algorithm of construction boolean function:
Step 1: We take any positive integer numbers $n_1, n_2, ..., n_k$ such that $n_i \neq n_j$, $i \neq j$.
Step 2: We put $A_1^1 = \{1, 2, ..., n_1\}, A_2^1 = \{n_1 + 1, ..., 2n_1\}, ..., A_{n_2}^1 = \{(n_2 - 1)n_1 + 1, ..., n_1 n_2\}$. We construct a boolean function for a direct product of two symmetric group $S_{n_1} \times S_{n_2}$. Let's take a boolean function

$$f_2(\underline{x}) = \begin{cases} 1 & \underline{x} = \underline{x}_1\underline{x}_2...\underline{x}_{n_2} (\text{ type } 1) \\ 1 & \underline{x} = \underbrace{0^{n_1}...0^{n_1}\overset{i}{1}^{n_1}0^{n_1}...0^{n_1}}_{n_2}, i = 1, 2, ..., n_2 \\ 0 & \text{otherwise} \end{cases}$$

In (type 1) for $i = 1, 2, ..., n_1$ we take all vectors for which there exist exactly two numbers $r, r' \in \{1, 2, ..., n_2\}, r \neq r'$ such that x_r and $x_{r'}$ are in the form $\underbrace{0...0\overset{i}{1}0...0}_{n_1}$, and 0 otherwise (all x_i are the same length).
Now we show that $S(f_2) = S_{n_1} \times S_{n_2}$. It is easy to see that f_2 is invariant under operation σ. From the other hand if $\sigma \notin S_{n_1} \times S_{n_2}$ we have two possibilities: (a) permutation σ act only inside blocks A_i^1 but there exists two of them where σ act in a different way (possibly $\sigma(A_i^1) = A_j^1$ and $i \neq j$ but this situation does not change much); (b) there exists positive integer i such that $\sigma(A_i^1) \neq A_j^1$, for $j = 1, 2, ..., n_2$. In the case (a) without loose of generality we can assume that σ act in different way in A_1^1 and A_2^1. Then there exist $i \in \{1, 2, ..., n_1\}$ such that $\sigma(i) = j$ and $\sigma(n_1 + i) = n_1 + k$ and $j \neq k$. Then we take a vector $\underline{x} = \underline{x}_1\underline{x}_2...\underline{x}_{n_2}$ where $\underline{x}_1$ and $\underline{x}_2$ are in the form $0...010...0$ where

only i-th coordinate is equal 1, and 0 otherwise. Then $f(\underline{x}) \neq f^{\sigma}(\underline{x})$. In the case (b) without loose of generality we can assume that $i = 1$. Because $n_1 \neq n_2$ then we take boolean vector $\underline{x} = 1^{n_1}0^{n_1}...0^{n_1}$. Then $f(\underline{x}) \neq f^{\sigma}(\underline{x})$.

Step 3: In this step we construct a boolean function which represent $S_{n_1} \times S_{n_2} \times S_{n_3}$. Let $A_1^2 = \{1, 2, ..., n_1n_2\}, A_2^2 = \{n_1n_2 + 1, ..., 2n_1n_2\}, ..., A_{n_3}^2 = \{(n_3 - 1)n_1n_2 + 1, ..., n_1n_2n_3\}$. Let's take a boolean function

$$f_3(\underline{x}) = \begin{cases} 1 & \underline{x} = \underline{x}_1\underline{x}_2...\underline{x}_{n_2n_3}(\text{ type } 1) \\ 1 & \underline{x} = x_1x_2...x_{n_3}(\text{ type } 2) \\ 1 & \underline{x} = \underbrace{0^{n_1n_2}...0^{n_1n_2}\overset{i}{1}^{n_1n_2}0^{n_1n_2}...0^{n_1n_2}}_{n_3}, i = 1, 2, ..., n_3 \\ 0 & \text{otherwise} \end{cases}$$

In (type 1) for $i = 1, 2, ..., n_1$ we take all vectors for which there exist exactly two numbers $r, r' \in \{1, 2, ..., n_2n_3\}, r \neq r'$ such that x_r and $x_{r'}$ are in the form $\underbrace{0...0\overset{i}{1}0...0}_{n_1}$, and 0 otherwise (all x_i are the same length). In (type 2) for $i = 1, 2, ..., n_2$ we take all vectors for which there exist exactly two numbers $r, r' \in \{1, 2, ..., n_3\}, r \neq r'$ such that x_r and $x_{r'}$ are in the form $\underbrace{0^{n_1}...0^{n_1}\overset{i}{1}^{n_1}0^{n_1}...0^{n_1}}_{n_2}$, and 0 otherwise (as previously all x_i are the same length).

Now we show that $S(f_3) = S_{n_1} \times S_{n_2} \times S_{n_3}$. It is easy to see that f_3 is invariant under operation σ. From the other hand if $\sigma \notin S_{n_1} \times S_{n_2} \times S_{n_3}$ we have two possibilities: (a) there does not exist positive integer i such that $\sigma(A_i^2) \neq A_j^2$, for $j = 1, 2, ..., n_3$ (possibly $\sigma(A_i^2) = A_j^2$ and $i \neq j$ but this situation does not change much); (b) there exists positive integer i such that $\sigma(A_i^2) \neq A_j^2$, for $j = 1, 2, ..., n_3$. In the case (a) previous step show us that we can construct a boolean vector $\underline{x}$ such that $f(\underline{x}) \neq f^{\sigma}(\underline{x})$. In the case (b) without loose of generality we can assume that $i = 1$. We can see that in the definition of boolean function we use 3 main type of vectors (three first lines). The number of coordinates equal 1 in each of them is equal n_2n_3, n_1n_3 and n_1n_2 respectively. Because $n_1 \neq n_2 \neq n_3$ then $n_1n_2 \neq n_1n_3 \neq n_2n_3$. So if we take boolean vector $\underline{x} = 1^{n_1n_2}0^{n_1n_2}...0^{n_1n_2}$ then $f(\underline{x}) \neq f^{\sigma}(\underline{x})$.

Step k: Now we construct a boolean function which represent $S_{n_1} \times S_{n_2} \times ... \times S_{n_k}$. Let $A_1^{k-1} = \{1, 2, ..., n_1n_2...n_{k-1}\}$, $A_2^{k-1} = \{n_1n_2...n_{k-1} + 1, ..., 2n_1n_2...n_{k-1}\}$, ...,

$A_{n_k}^{k-1} = \{(n_k - 1)n_1 n_2 ... n_{k-1} + 1, ..., n_1 n_2 ... n_k\}$. Let's take a boolean function

$$f_k(\underline{x}) = \begin{cases} 1 & \underline{x} = \underline{x}_1 \underline{x}_2 ... \underline{x}_{\alpha_2^k} (\text{ type } 1) \\ ... & ... \\ 1 & \underline{x} = x_1 x_2 ... x_{\alpha_{j+1}^k} (\text{ type } j) \\ ... & ... \\ 1 & \underline{x} = \underbrace{0^{\alpha_1^k} ... 0^{\alpha_1^k} \overset{i}{1}^{\alpha_1^k} 0^{\alpha_1^k} ... 0^{\alpha_1^k}}_{n_k}, i = 1, 2, ..., n_k \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_l^m = n_l n_{l+1} ... n_m$, for $l \leq m$.
In (type 1) for $i = 1, 2, ..., n_1$ we take all vectors for which there exist exactly two numbers $r, r' \in \{1, 2, ..., \alpha_2^k\}, r \neq r'$ such that x_r and $x_{r'}$ are in the form $\underbrace{0...0\overset{i}{1}0...0}_{n_1}$, and 0 otherwise (all x_i are the same length). In (type j) for $i = 1, 2, ..., n_j$ we take all vectors for which there exist exactly two numbers $r, r' \in \{1, 2, ..., \alpha_{j+1}^k\}, r \neq r'$ such that x_r and $x_{r'}$ are in the form $\underbrace{0^{\alpha_1^{j-1}} ... 0^{\alpha_1^{j-1}} \overset{i}{1}^{\alpha_1^{j-1}} 0^{\alpha_1^{j-1}} ... 0^{\alpha_1^{j-1}}}_{n_j}$, and 0 otherwise (as previously all x_i are the same length).
Now we show that $S(f_k) = S_{n_1} \times S_{n_2} \times ... \times S_{n_k}$. It is easy to see that f_k is invariant under operation σ. From the other hand if $\sigma \notin S_{n_1} \times S_{n_2} \times ... \times S_{n_k}$ we have two possibilities: (a) there does not exist positive integer i such that $\sigma(A_i^{k-1}) \neq A_j^{k-1}$, for $j = 1, 2, ..., n_k$ (possibly $\sigma(A_i^{k-1}) = A_j^{k-1}$ and $i \neq j$ but this situation as before does not change much); (b) there exists positive integer i such that $\sigma(A_i^{k-1}) \neq A_j^{k-1}$, for $j = 1, 2, ..., n_k$. In the case (a) previous step show us that we can construct a boolean vector $\underline{x}$ such that $f(\underline{x}) \neq f^\sigma(\underline{x})$. In the case (b) without loose of generality we can assume that $i = 1$. As before the number of coordinates equal 1 in each type of vector used in the definition of function f_k is different. So if we take boolean vector $\underline{x} = 1^{n_1 ... n_{k-1}} 0^{n_1 ... n_{k-1}} ... 0^{n_1 ... n_{k-1}}$ then $f(\underline{x}) \neq f^\sigma(\underline{x})$. □

Next theorem present construction of 2 valued boolean function f in the case when all lengths of blocks are equal n.

Theorem 3 *For any positive integer number n there exist 2 valued boolean function f such, that*

$$S(f) = S_n \times S_n \times ... \times S_n$$

if this product contain more then two component.

Proof We consider a set $\{1, 2, ..., n^k\}$. We put

$$A_i^j = \{(i-1)n^j + 1, (i-1)n^j + 2, ..., in^j\}$$

for $i = 1, 2, ..., n^{k-j}$ and $j = 0, 1, ..., k$. We see that for $j = 0$ blocks A_i^0 contain only one element and for $j = k$ we have only one block equal all set. We can notice, that each block A_i^j is divided into n blocks A_k^{j-1} where $k = (i-1)n + 1, (i-1)n + 2, ..., in$.

Algorithm of construction boolean function:

Now we construct a family of boolean vectors needed to create a boolean function (we understand blocks A_i^j as a coordinates of boolean vector $\underline{x}$)

Step 1: For $i = 1, 2, ..., n$ we construct boolean vectors $\underline{x}$ of the length n^k each of which are in the following form: in exactly two blocks A_m^1 and $A_{m'}^1$ where $m, m' = 1, 2, ..., n^{k-1}, m \neq m'$ have i-th coordinate equal 1 and 0 otherwise. Precisely in blocks A_m^1 and $A_{m'}^1$ coordinates $(m-1)n + i$ and $(m'-1)n + i$ are equal 1, and 0 otherwise. The number of such vectors is equal $\frac{1}{2}n^k(n^{k-1} - 1)$, and each of this vector has only 2 coordinates equal 1.

Step j: $(j < k)$: For $i = 1, 2, ..., n$ we construct boolean vectors $\underline{x}$ of the length n^k each of which are in the following form: in exactly two blocks A_m^j and $A_{m'}^j$ where $m, m' = 1, 2, ..., n^{k-j}, m \neq m'$ blocks $A_{(m-1)n+i}^{j-1}$ and $A_{(m'-1)n+i}^{j-1}$ are equal 1, and 0 otherwise. The number of such vectors is equal $\frac{1}{2}n^{k-j+1}(n^{k-j} - 1)$, and each of this vector has $2n^{j-1}$ coordinates equal 1.

Step k: For $i = 1, 2, ..., n$ we construct boolean vectors $\underline{x}$ of the length n^k each of which are constructed in the following form: exactly two blocks A_m^{k-1} and $A_{m'}^{k-1}$ where $m, m' = 1, 2, ..., n, m \neq m'$ (in this one block A_1^k) are equal 1, and 0 otherwise. The number of such vectors is equal n, and each of this vector has $2n^{k-1}$ coordinates equal 1.

Now we create a boolean function f by putting 1 for all vectors constructed above, and 0 otherwise. We consider two cases:

Case (I): $n > 2$. It is easy to see that $S_n \times ... \times S_n \subseteq S(f)$. Now we show that inclusion $\supseteq$ is hold. We proof this by induction on j. So:

Step 1: let $j = 1$ and $\sigma \notin S_n \times ... \times S_n$

(a) Let permutation σ act only inside blocks A_i^1 but there exist two of them where σ act in a different way (possibly $\sigma(A_i^1) = A_j^1$ and $i \neq j$ but this situation does not change much). Then there exist $m, m' \in \{1, 2, ..., n^{k-1}\}, m \neq m'$ such, that σ act in a different way in A_m^1 and $A_{m'}^1$. So there exist $l \in \{1, 2, ..., n\}$ that $\sigma((m-1)n+l) = (m-1)n+p$ and $\sigma((m'-1)n + l) = (m'-1)n + p'$, $p, p' \in \{1, 2, ..., n\}, p \neq p'$. Then we take a boolean vector $\underline{x}$ such, that in blocks A_m^1 and $A_{m'}^1$ l-th coordinate (from the beginning of the block) is equal 1, and 0 otherwise. Then we have $f(\underline{x}) \neq f^\sigma(\underline{x})$.

(b) Let there exists positive integer i such that $\sigma(A_i^1) \neq A_j^1$, for $j = 1, 2, ..., n^{k-1}$. Without loose of generality we can assume that $i = 1$. Then inside the block A_1^1 there exist coordinate, for example $l \in \{1, 2, ..., n\}$ such, that $\sigma(l) \in A_{m'}^1, m' \neq 1$. More-

over there exist coordinate l' inside A_1^1 such, that $\sigma(l') \in A^1_{m''}, m'' \neq m'$. Now we have three possibilities: (b1) $m' = (s'-1)n + p', m'' = (s''-1)n + p'', p' \neq p''$. Then we consider a vector $\underline{x}$ which has blocks A_1^1 and A_{n+1}^1 all equal 1, 0 otherwise. Then $\underline{x}^\sigma$ has at least one coordinate equal 1 in $A^1_{m'}$ and $A^1_{m''}$ so $f^\sigma(\underline{x}) = 0$; (b2) $m' = (s'-1)n + p, m'' = (s''-1)n + p, p \neq 1$. Then we consider two vectors $\underline{x}_1$ which has blocks A_1^1 and A_{n+1}^1 all equal 1, 0 otherwise, and $\underline{x}_2$ which has blocks A_1^1 and A_{2n+1}^1 all equal 1, 0 otherwise. Then we see that $f^\sigma(\underline{x}_1) = 0$ or $f^\sigma(\underline{x}_2) = 0$; (b3) $m' = (s'-1)n + 1$ and $m'' = (s''-1)n + 1$. If $m'' \neq 1$ then we take two boolean vectors: $\underline{x}_1$ which has A_1^1 and $A^1_{m'}$ all equal 1, 0 otherwise and $\underline{x}_2$ which has A_1^1 and $A^1_{m''}$ all equal 1, 0 otherwise. If $f^\sigma(\underline{x}_1) = 0$ then it is the end of this case. If $f^\sigma(\underline{x}_1) = 1$ then there exist r' and r'' from the block $A^1_{m'}$ such, that $\sigma((m'-1)n + r') \in A^1_{m'}$ and $\sigma((m'-1)n + r'') \in A^1_{m''}$ (that mean some element from $A^1_{m'}$ stay inside this block, and some element go to block $A^1_{m''}$). Then $\underline{x}_2^\sigma$ give us a vector with 1 and 0 inside $A^1_{m'}$ so $f^\sigma(\underline{x}_2) = 0$. If $m'' = 1$ then for σ there exist l in some block $\overline{m}$ such, that $\sigma(l) \in A_1^1$. Then we take a boolean vector $\underline{x}$ with blocks A_1^1 equal 1 and $A^1_{\overline{m}}$ equal 0. Then vector $\underline{x}^\sigma$ give us a vector with 0 and 1 inside block A_1^1 so $f^\sigma(\underline{x}) = 0$.

Step j: $(1 < j < k)$ Because $\sigma \notin S_n \times ... \times S_n$ and from previous steps we know that there exist only two cases.

(a) If σ act only inside blocks A_i^j but there exist two of them where σ act in a different way (possibly $\sigma(A_i^j) = A_s^j$ and $i \neq s$ but this situation does not change much). Then there exist $m, m' \in \{1, 2, ..., n^{k-j}\}, m \neq m'$ such, that σ act in different way inside A_m^j and $A^j_{m'}$. From previous steps we know that we have following situation: there exist $l \in \{1, 2, ..., n\}$ such that $\sigma(A^{j-1}_{(m-1)n+l}) = A^{j-1}_{(m-1)n+l'}$ and $\sigma(A^{j-1}_{(m-1)n+l}) = A^{j-1}_{(m'-1)n+l''}$ and $l' \neq l''$. Then we take a vector $\underline{x}$ which in block A_m^j has block $A^{j-1}_{(m'-1)n+l}$ equal 1 and in $A^j_{m'}$ has block $A^{j-1}_{(m'-1)n+l}$ equal 1. Then $f(\underline{x}) \neq f^\sigma(\underline{x})$.

(b) Let there exists positive integer i such that $\sigma(A_i^j) \neq A_s^j$, for $s = 1, 2, ..., n^{k-j}$. Without loose of generality we can assume that $i = 1$. Inside block A_1^j there exist coordinates l and l' $(l \neq l')$ such, that $\sigma(l) \in A^j_{m'}, m' \neq 1$ and $\sigma(l') \in A^j_{m''}, m'' \neq m'$. As before we must consider all cases (in step 1 we denote its by b1,b2,b3). This considerations are similar so we present here only a case $b3$ because it is a bit more difficult. If $(m'' \neq 1)$ then we take a vector $\underline{x}_1$ with blocks A_1^j and $A^j_{m'}$ equal 1. The vector $\underline{x}_2$ has A_1^j and $A^j_{m''}$ equal 1. If $f^\sigma(\underline{x}_1) = 0$ then this is the end of this case. If $f^\sigma(\underline{x}_1) = 1$ then there exist r', r'' inside $A^j_{m'}$ such, that $\sigma((m'-1)n^j + r') \in A^j_{m'}$ and $\sigma((m'-1)n^j + r'') \in A^j_{m''}$. Then $\underline{x}_2^\sigma$ give us a vector with 1 and 0 inside $A^j_{m'}$ so $f^\sigma(\underline{x}_2) = 0$.

Step k: Here we have only one block $A_1^k = \{1, 2, ..., n^k\}$ which is divided into $A_1^{k-1}, ..., A_n^{k-1}$. Because $\sigma \notin S_n \times ... \times S_n$ we have to consider one case: there exists positive integer i such that $\sigma(A_i^{k-1}) \neq A_s^{k-1}$, for $s = 1, 2, ..., n$. Without loose of gen-

erality we can assume that $i = 1$. After considerations similar to case (b) in previous steps we conclude that $\sigma \notin S(f)$.

Case (II): $n = 2, k > 2$. Construction of boolean function f is the same as before in steps $1, 2, ..., k-1$. Now we present step k in this construction: We have blocks $A_1^{k-1} = \{1, 2, ..., n^{k-1}\}, A_2^{k-1} = \{n^{k-1}+1, n^{k-1}+2, ..., n^k\}$. We put one of them all equal 1 (of course here we have two cases which we consider), but inside second block we put only coordinate i (from beginning of that block) equal 1 and 0 otherwise, for $i = 1, 2, ..., n^{k-1}$. For that kind of vectors we put 1 as a value of boolean function f and 0 otherwise.

Steps of proof are the same from 1 to $k-1$. There is a difference only in step k: let $\sigma \notin S_2 \times ... \times S_2$ and σ is not reject in previous steps. So $\sigma(A_1^{k-1}) \neq A_1^{k-1}$ and $\sigma(A_1^{k-1}) \neq A_2^{k-1}$. Then there exist $l \in A_1^{k-1}$ such that $\sigma(l) = l' \in A_2^{k-1}$. We take a boolean vector $\underline{x}$ with all block A_1^{k-1} equal 1 and inside A_2^{k-1} coordinate l' is equal 1. If $f^\sigma(\underline{x}) = 0$ then this is the end of this case. If $f^\sigma(\underline{x}) = 1$ then we have two possibilities:
(a) there exist exactly one element l inside A_1^{k-1} such that $\sigma(l) \in A_2^{k-1}$ and there exist exactly one element l' inside A_2^{k-1} such that $\sigma(l') \in A_1^{k-1}$. Then we take vector $\underline{x}$ which has block A_1^{k-1} all equal 1 and inside A_2^{k-1} on the l' coordinate has 0. Then $\underline{x}^\sigma$ give us a vector with exactly one 0 and 1 otherwise in block A_1^{k-1} so $f^\sigma(\underline{x}) = 0$.
(b) there exist exactly one element l inside A_1^{k-1} such that $\sigma(l) \in A_1^{k-1}$ and there exist exactly one element l' inside A_2^{k-1} such that $\sigma(l') \in A_2^{k-1}$. When we consider the same boolean vector as in (a) we see that $\underline{x}^\sigma$ give as a vector with exactly one 0 and 1 otherwise in block A_2^{k-1} so $f^\sigma(\underline{x}) = 0$. □

5. Final remarks

The results of this paper show, that we can create specific algorithms of constructions of boolean functions which represents direct product of symmetric groups. In the section 3 we present an upper bound for k representability of direct product of permutation groups. We show, how we can construct boolean function with k_1+k_2+2 different values which represent direct product of two groups (k_1 and k_2 representable respectively). In section 4 we present two major cases of direct product of symmetric groups: (a) when we have finite numbers of symmetric groups, each of which act on the set with different number of components; (b) when we consider a direct product of finite number of copies of the same symmetric group. This considerations could be useful when we try to solve the problem of placement of modules on the integrated circuit, where permutation of inputs is allowed or when order of input values is only partially given.

References

1. L. Beame, H. Bodlaender: *Distributed computing and transitive networks*, 6th Annual Symposium of Theoretical Aspects of Computer Science, 1989.
2. N.L. Biggs, A.T. White: *Permutation groups and combinatorial structures*, London Math. Soc., Lecture Notes Series 33, Cambridge Univ. Press. Cambridge, 1979.
3. P. Clote, E. Kranakis: *Boolean function invariance groups, and parallel complexity*, J. Comput., Vol. 20, No. 3, 1991, 553-590.
4. P. Clote, E. Kranakis: *Boolean function and Computation Models*, Springer, Berlin, 2002.
5. M. Grech, A. Kisielewicz: *Direct product of automorphism groups of colored graphs*, Discrete Math., **283**, 2004, 81-86.
6. M. Harrison: *On the classification of Boolean functions by the general linear and affine groups*, J. Soc. Indust. Apply Math, **12**, 1964, 285-299.
7. A. Kisielewicz: *Symmetry groups of Boolean functions and constructions of permutation groups*, Journal of Algebra., **199** , 1998, 379-403.

Reprezentacja produktu prostego grup permutacji za pomocą grup symetrii funkcji boolowskich

Streszczenie

Rozważamy moduł M o n stanach wejściowych $x_1, x_2, ..., x_n$, z których każdy może przyjmować jeden z dwóch możliwych stanów 0 lub 1. Ponadto na wyjściu rozważany moduł przyjmuje także tylko te same dwa stany 0 lub 1 (uogólnieniem takich modułów są urządzenia przyjmujące na wyjściu k różnych wartości dla $k > 2$, o których także piszemy w jednym z rozdziałów tej pracy). Wyjście modułu w ogólności zależy od uporządkowania danych wejściowych $x_1, x_2, ..., x_n$. Istnieją pewne permutacje danych wejściowych które pozostawiają stan wyjściowy niezmieniony. Na przykład jeśli wyjście modułu M w ogóle nie zależy od uporządkowania danych wejściowych to taki moduł nazywamy symetrycznym. Każdy taki moduł może być jednoznacznie powiązany z opisującą go funkcją boolowską. Mając daną taką funkcję możemy skonstruować grupę permutacji (grupę symetrii) odpowiadającą tej funkcji. Dla funkcji boolowskiej $f : \{0,1\}^n \to \{0,1,...,k-1\}$ grupę taką definiujemy jako zbiór wszystkich permutacji σ należących do grupy symetrycznej zbioru n elementowego które spełniają warunek

$$f(x_1, x_2, ..., x_n) = f(x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(n)})$$

dla dowolnego wektora boolowskiego $(x_1, x_2, ..., x_n)$. Punktem wyjścia do naszych rozważań były prace P. Clote, E. Kranakis, *Boolean function invariance groups, and parallel complexity.*, J. Comput., Vol. 20, No. 3, 1991, 553-590. oraz P. Clote, E. Kranakis,

Boolean function and Computation Models, Springer, Berlin, 2002. W pracach tych pojawiły się pewne specjalne konstrukcje funkcji boolowskich oraz grup symetrii tych funkcji. Rozważany problem ma bezpośrednie odniesienie do ułożenia takich modułów w układzie scalonym, w którym permutacje danych wejściowych są dozwolone, lub gdy porządek wejść jest zadany tylko częściowo. Główne rezultaty uzyskane w tej pracy dotyczą iloczynów prostych grup permutacji, a w szczególności iloczynów prostych grup symetrycznych. Przedstawiono tu konkretne algorytmy tworzenia funkcji boolowskich w przypadkach gdy rozważamy iloczyny dowolnej (skończonej) liczby grup symetrycznych. Rozważania zostały podzielone na kilka odrębnych przypadków. W sytuacji ogólnej, gdy rozważamy iloczyn prosty dowolnych grup permutacji podajemy ograniczenie górne dla k reprezentowalności poprzez funkcję boolowską. Podobnie jak autorzy P. Clote, E. Kranakis uważamy, że rozważania dotyczące grup symetrii funkcji boolowskich doprowadzą do algorytmów optymalizujących przestrzeń w projektowaniu układów VLSI. W szczególności mogą one odpowiedzieć na pytanie, kiedy możemy zamienić miejscami poszczególne moduły lub bloki modułów nie zmieniając przy tym uzyskiwanej funkcji na wyjściu.