Math-Net.Ru

Общероссийский математический портал

S. A. Lozhkin, M. S. Shupletsov, B. R. Danilov, Синтез асимптотически оптимальных по сложности схем из функциональных элементов, защищенных от раскрытия функциональности, *Матем. вопр. криптогр.*, 2017, том 8, выпуск 2, 87–96

DOI: <https://doi.org/10.4213/mvk226>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 54.196.121.252

26 апреля 2023 г., 13:55:40

Synthesis of asymptotically size-optimal Boolean circuits protected from functionality inference

S. A. Lozhkin, M. S. Shupletsov, B. R. Danilov

Lomonosov Moscow State University, Moscow

Получено 11.VI.2016

Abstract. We propose several asymptotically size-optimal Boolean circuits synthesis methods that implement arbitrary Boolean functions of a given number of Boolean variables with a given protection level from functionality inference when concealing some number of local interconnections. These methods rely on the structure of Boolean circuits over arbitrary finite complete basis. Constructed by methods of generalized decomposition and universal systems of Boolean functions.

Keywords: Boolean circuits, asymptotic methods, obscuration of functionality

Синтез асимптотически оптимальных по сложности схем из функциональных элементов, защищенных от раскрытия функциональности

С. А. Ложкин, М. С. Шуплецов, Б. Р. Данилов

Московский государственный университет им. М. В. Ломоносова, Москва

Аннотация. Предлагаются методы синтеза асимптотически оптимальных по сложности схем из функциональных элементов, которые реализуют произвольные функции алгебры логики от заданного числа переменных и обладают заданным уровнем защищенности от раскрытия их функциональности при сокрытии определенного числа локальных соединений. Эти методы опираются на особенности структуры схем из функциональных элементов в произвольном базисе, построенных с использованием методов обобщенного разложения и универсальных систем функций алгебры логики.

Ключевые слова: схема из функциональных элементов, асимптотические методы, сокрытие закона функционирования

Citation: *Mathematical Aspects of Cryptography*, 2017, v. 8, № 2, pp. 87–96 (Russian)

© Академия криптографии Российской Федерации, 2017 г.

1. Generalized decomposition and universal sets of Boolean functions

Let $B = \{0, 1\}$ and B^n , $n = 1, 2, \dots$, be n -dimensional Boolean cube, that is a set of n -length Boolean vectors. Suppose $\Delta = (\delta_1, \dots, \delta_t)$ is a partition of a Boolean cube B^n into disjoint subsets δ_i , ($\delta_i \subseteq B^n$), $x = (x_1, \dots, x_n)$ is a vector of Boolean variables corresponding to dimensions of a Boolean cube B^n , $y = (y_1, \dots, y_t)$ is a vector of Boolean variables corresponding to components $\delta_1, \dots, \delta_t$ of the partition Δ . By $\mu_\Delta(x, y)$ we denote a so called *multiplexor Boolean function of the partition Δ* which for any i , α ($1 \leq i \leq t$, $\alpha \in \delta_i$) satisfies the equation $\mu_\Delta(\alpha, y) = y_i$, or in other terms the equation

$$\mu_\Delta(x, y) = \bigvee_{i=1}^t \chi_i(x) y_i,$$

where $\chi_i(x)$ is the characteristic Boolean function of the set δ_i .

Multiplexor μ_n of order n , that is Boolean function of the form

$$\mu_n(x_1, \dots, x_n, y_0, \dots, y_{2^n-1}) = \bigvee_{(\sigma_1, \dots, \sigma_n) \in B^n} x_1^{\sigma_1} \dots x_n^{\sigma_n} y_{\nu(\sigma_1, \dots, \sigma_n)},$$

where $x_i^0 = \bar{x}_i$, $x_i^1 = x_i$, and $\nu(\sigma_1, \dots, \sigma_n) = \sigma_1 2^{n-1} + \dots + \sigma_n 2^0$, is a special case of Boolean function $\mu_\Delta(x, y)$ that corresponds to a *trivial* partition Δ of Boolean cube B^n into independent Boolean vectors ordered in accordance with their lexicographic numeration ν .

Let's now consider a Boolean function $\psi(y_1, \dots, y_t)$ that essentially depends on all its variables. From [1, 2, 3] it follows that for Boolean function ψ and partition Δ of the cube B^n there always exist Boolean functions $g_1(x, y_1), \dots, g_t(x, y_t)$ such that

$$\mu_\Delta(x, y) = \psi(g_1(x, y_1), \dots, g_t(x, y_t)). \quad (1)$$

We will call the Boolean function ψ *external* function and a vector of functions $g = (g_1, \dots, g_t)$ a vector of *internal* Boolean functions of the representation (1). Let's notice that it is possible to choose each internal Boolean function $g_i(x, y_i)$, $1 \leq i \leq t$, of the representation (1) in such a way that for any $\alpha \in B^n$ values $g_i(\alpha, 0)$ and $g_i(\alpha, 1)$ depend only on the number of the partition Δ component which contains α and are different if and only if this number equals to i . We will refer to any vector $g = (g_1, \dots, g_t)$ satisfying the specified property as to *standart* for partition Δ . It is obvious that in such vector variable y_i of each Boolean function $g_i(x, y_i)$ is either monotonic or not inmonotonic.

From (1) it follows that for any Boolean function $f(x)$ the following representation holds:

$$f(x) = \psi(g_1(x, f_1(x)), \dots, g_t(x, f_t(x))), \quad (2)$$

where $f_i(x)$ for each $i \in \{1, \dots, t\}$ is an arbitrary Boolean function that equals to f on δ_i . The representation (2) is called a *generalized decomposition* of a Boolean function f based on the external Boolean function ψ and internal Boolean functions $g_1, \dots, g_t$ from (1) that are assigned to the partition Δ . A special case of the representation (2) is called a *generalized representation of Boolean function $f(x)$ with respect to Boolean variable $x' = (x_1, \dots, x_k)$* based on external Boolean function $\psi(y_0, \dots, y_{2^k-1})$ if:

- 1) in partition Δ (called a consequent segment partition) each set $\delta_i, 0 \leq i \leq 2^k - 1$, $\Delta = (\delta_0, \dots, \delta_{2^k-1})$, consists of all vectors $\alpha \in B^n$ such that

$$i 2^{n-k} \leq \nu(\alpha) < (i+1) 2^{n-k},$$

- 2) each Boolean function $g_i(x, y_i)$ corresponding to δ_i doesn't essentially depend on Boolean variable $x'' = (x_{k+1}, \dots, x_n)$,
- 3) Boolean function $f_i(x)$ equals to $f(\sigma', x'') = f_{\sigma'}(x'')$ if $\nu(\sigma') = i$. Representation (2) in this case takes the form

$$f(x) = \psi(g_0(x', f_{(0\dots 0)}(x'')), \dots, g_{2^k-1}(x', f_{(1\dots 1)}(x''))). \quad (3)$$

Note that representation (3) is based on the external Boolean function of the form

$$\psi(y_0, \dots, y_{2^k-1}) = y_0 \vee \dots \vee y_{2^k-1} \quad (4)$$

and the standard set of internal Boolean functions of the form $x_1^{\sigma_1} \dots x_k^{\sigma_k} y_{\nu(\sigma_1, \dots, \sigma_k)}$, corresponds to the ordinary representation of Boolean function f with respect to Boolean variable x' :

$$f(x) = \bigvee_{\sigma'=(\sigma_1, \dots, \sigma_k) \in B^k} x_1^{\sigma_1} \dots x_k^{\sigma_k} f_{\sigma'}(x''). \quad (5)$$

If $P_2(n)$ is the set of all Boolean functions that depend on Boolean variables $x_1, \dots, x_n$, F is a subset of $P_2(n)$, and Boolean function $\phi(y_1, \dots, y_t)$ essentially depends on all its variables, then we say that a set $G \subseteq P_2(n)$ is a *φ -universal set (φ -US) of functions for the set of functions F* if any Boolean function $f \in F$ may be represented in the form $f = \phi(g_1, \dots, g_t)$, where $g_i \in G$ for any i , $1 \leq i \leq t$. We will call the set G a *φ -universal set of order n* in the special case $F = P_2(n)$ which is usually considered.

Let us note that the set of all different Boolean functions of the form $g_i(x, f_i(x))$, $1 \leq i \leq t$, that occur in the generalized decomposition (2) for all different Boolean functions f from F makes up a ψ -US for set F which we will call *standard* if the corresponding vector of internal Boolean functions $(g_1, \dots, g_t)$ is standard. In the latter case the specified ψ -US G of order n has the form

$$G = G^{(1)} \cup \dots \cup G^{(t)}, \quad (6)$$

where the set $G^{(i)}$, $1 \leq i \leq t$, consists of all $2^{|\delta_i|}$ different Boolean functions depending on Boolean variables x that equal to Boolean function $g_i(x, 0) = g_i(x, 1)$ on all components of the partition Δ except the component δ_i .

Let us consider a problem of a φ -US optimization using selector partitions. Let, as before, Boolean function $\psi(y_1, \dots, y_t)$ be essentially depending on all its variables that make up a set Y , and let D be a partition of Y into non-empty pairwise non-overlapping subsets $Y_1, \dots, Y_d$. We call partition D *selector* if for any i, j such that $1 \leq i \leq d$, $y_j \in Y_i$ there exist Boolean constants $\gamma_1^{(j)}, \dots, \gamma_{i-1}^{(j)}, \gamma_{i+1}^{(j)}, \dots, \gamma_d^{(j)}$ that (being substituted correspondingly in place of all variables from $Y_1, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_d$) transform Boolean function ψ into Boolean function $y_j^{\tau_j}$, where τ_j is a Boolean constant. Let's note that if D is a selector partition then for an arbitrary partition $\Delta = (\delta_1, \dots, \delta_t)$ of Boolean cube B^n the set of Boolean functions $g_k(x, y_k)$, $1 \leq k \leq t$, of the form $g_k(x, y_k) = \hat{g}_k(x, y_k^{\tau_k})$, where

- 1) $\hat{g}_k(\alpha, 0) = \hat{g}_k(\alpha, 1) = \gamma_p^{(j)}$ for any Boolean vector $\alpha \in \delta_j$, if $y_k \in Y_p$, $y_j \in Y_i$ and $p \neq i$,
- 2) $\hat{g}_k(\alpha, 0) = 0$ and $\hat{g}_k(\alpha, 1) = 1$ in other cases,

is the set of internal Boolean functions of the representation (1), which we will call a *selector* for ψ , D , and Δ . Based on such set of Boolean functions one can make corresponding selector ψ -US G of the form (compare with (6))

$$G = G^{(1)} \cup \dots \cup G^{(d)}, \quad (7)$$

where the set $G^{(i)}$ consists of 2^{r_i} , $r_i = \max_{y_j \in Y_i} |\delta_j|$, different Boolean functions of Boolean variable x which equal to $\gamma_i^{(j)}$ on component δ_j when $y_j \notin Y_i$, and run through all possible values on vectors from δ_j when $y_j \in Y_i$ taking equal values on vectors with equal ordinal numbers (relative to numbering ν) from different components δ_j of the specified form.

It is shown in [1] (see also [2]) that described earlier selector vectors of internal Boolean functions of the generalized decomposition and universal sets of Boolean functions of the form (7) may be implemented substantially simpler than similar standard sets. The mentioned simplification is the larger, the smaller the entropy $H(D)$ of the utilized partition D (see [1]):

$$H(D) = -\frac{1}{|Y|} \sum_{i=1}^d |Y_i| \log \frac{|Y_i|}{|Y|}.$$

Let D be a selector partition of variables set $Y = \{y_1, \dots, y_t\}$ of a Boolean function $\psi(y_1, \dots, y_t)$. Then for any natural s , $s > \log_2 t$, and any natural n , $2^n \leq t(s - H(D))$, it is possible to construct a ψ -universal set of order n consisting of at most 2^{s+2} Boolean functions.

2. Synthesis of Boolean circuits protected from functionality inference

Approaches based on generalized composition, universal Boolean function systems, and special Boolean cube partitions are taken as a principle of a number of asymptotically accurate methods of circuits synthesis. These approaches allow to establish so called high accuracy asymptotic bounds for Shannon functions [1, 3]. A number of classic methods like C. Shannon and O. B. Lupanov representations may be described using these constructions. In this article we consider a variant of these methods oriented on Boolean circuits synthesis in an arbitrary finite complete basis $= \{1, \dots, b\}$. The reduced weight ρ of basis is defined as follows:

$$\rho = \min_{1 \leq j \leq b, k_j \neq 1} \rho_j,$$

where $\rho_j = L_j / (k_j - 1)$ is the reduced weight of the gate j , L_j is the size-complexity of the gate j , k_j is the number of inputs of j . The size-complexity $L(\Sigma)$ of the circuit Σ , the size-complexity $L(f)$ of the Boolean function f , and Shannon function $L(n)$ for complexity of Boolean functions from $P_2(n)$ in the class of Boolean circuits over basis are defined as usual, see [1].

The general structure of the asymptotically best circuits synthesis method for arbitrary basis is as follows. Initially Boolean function f , $f \in P_2(n)$, is decomposed according to the equation (3) by means of its generalized decomposition with respect to its last $(n - q)$ Boolean variable $x' = (x_{q+1}, \dots, x_n)$, where the number q is a parameter, and a Boolean function ψ , that is implemented with the formula class circuit Ψ .

Each function $f_\sigma(x'')$, $\sigma \in B^{n-q}$, of the decomposition (3) that depends on Boolean variable $x'' = (x_1, \dots, x_q)$ is implemented on the basis of the representation

$$f_\sigma = \phi(g_{\sigma,1}, \dots, g_{\sigma,t}) \quad (8)$$

via the function $\phi = \phi(y_1, \dots, y_t)$ that is evaluated with another formula type circuit Φ and ϕ -universal set G of order q based on this function (given by (7)) that is constructed using selector partition D of the set Y of the function ϕ variables. Therewith in (8) $g_{\sigma,i} \in G^{(s)}$ ($1 \leq i \leq t$, $1 \leq s \leq d$) if and only if $y_i \in Y_s$. Let Σ_A be the part of the circuit that implements functions of the set G . Let us denote by Σ_B the circuit that implements the system of functions $\{f_\sigma\}$ based on decomposition (8) using circuit Σ_A outputs for reading corresponding functions $g_{\sigma,i}$.

The methods due to C. Shannon and O. B. Lupanov are considered for the so called *standard* basis ϕ_0 , $\phi_0 = \{\&, \vee, \neg\}$. Boolean function ψ has the form (4) for both methods. For the Shannon method $t = 1$, $\phi(y_1) = y_1$, partition D is trivial: $Y = Y_1 = y_1$, and the set G equals to $P_2(q)$. For the Lupanov method t is still a parameter, partition D is trivial, i. e. $Y_j = \{y_j\}$, $1 \leq j \leq t$, Boolean function ϕ equals to the Boolean function $y_1 \vee \dots \vee y_t$, and the set G is a standard ϕ -universal set of order q . Therewith partition Δ that is implicitly used in (8) is a partition of Boolean cube B^q into consequent segments.

Let us note that any Boolean function from the corresponding subset $G^{(s)}$ of set G may be substituted in place of any Boolean variable y_i from component Y_s . Any two different substitutions of the given form correspond to implementations of different Boolean functions. Thereby we get the same circuit for the whole family of pairwise non-equivalent initial circuits by breaking all or some interconnections that link in the subcircuit Σ_B inputs of formulas Φ with the outputs of the subcircuit Σ_A .

Now we consider the standard basis ϕ_0 . Let, for instance, formula $\Phi = \Phi_r$ represents full binary tree of r levels that is constructed from blocks Φ_1 implementing the Boolean function $\phi_1(x_1, x_2, x_3, x_4) = x_1 x_3 \vee x_2 x_4$ by means of connection of $(2^r - 1)$ such blocks via inputs x_3, x_4 . It is to be noted that formula Φ has $t = t_r = 2 + 4 + \dots + 2^r + 2^r$ inputs and implements Boolean function $\phi = \phi_r(y_1, \dots, y_t)$ that has a so called «geometric» [1] (of factor 2) selector partition D of its Boolean variables that consists of $(2r + 1)$ components of cardinalities $1, 1, \dots, 2^r - 1, 2^r - 1, 2^r$. Therewith the last of the specified components is an additional component of the partition and consists of inputs with numbers 3, 4 of all 2^{r-1} blocks Φ_1 situated on the tree level number r of the formula the Φ .

Thereby the number of possible “sensible” restorations of interconnections equals to q^{2^r} when the formula Φ inputs corresponding to the additional component of the partition D are detached from outputs of circuit Σ_A where q different Boolean functions from ϕ -US G corresponding to these inputs are implemented. It is to be noted that even in the special case when during specified restoration it is allowed to switch only inputs 3 and 4 of each element situated on the last level number r of the formula Φ tree the number of variants is 2^{2^r} . Note that if the Boolean function $f(x)$ is implemented by means of circuit Σ described then the number λ of formulas Φ being used equals to

$$\lambda = 2^k \left\lceil \frac{2^{n-k}}{t(s-H(D))} \right\rceil,$$

where (see statement 1) $s > \log_2 t$, $2^{n-k} \leq t(s-H(D))$, $|G| \leq 2^{s+2}$.

Thereby the number of different variants of “recovery” of the whole circuit Σ equals to $q^{2^r \cdot \lambda}$ and $2^{2^r \cdot \lambda}$ in the considered general and special cases of breaking connections correspondingly. To estimate the growth rate of the number of recovery variants we note that in asymptotically best methods the parameters s and t grow linearly in n and parameter k grows logarithmically.

Based on the constructions described the following theorem was proved.

Theorem 1. *There exists an algorithm that for any Boolean function $f \in P_2(n)$ constructs a Boolean circuit Σ over basis $_0$ that implements f and is such that*

$$L(\Sigma) \leq \frac{2^n}{n} \left(1 + \frac{2 \log_2 n + O(1)}{n} \right).$$

At the same time, there exists a set S in Σ that consists of connected disjoint single-output subcircuits which depths are at most 2 and which have 2 dedicated inputs each, such that:

- 1) *the order of value $|S|$ is at least $2^n/n$,*
- 2) *if p is a number of subcircuits in S such that each subcircuit has different Boolean functions on its input, then the circuit Σ' obtained from Σ by deletion of all connections to dedicated inputs from specified subcircuits allows for at least 2^p pairwise non-equivalent variants of the recovery (within the specified subcircuits and their dedicated inputs) of deleted connections.*

Remark 1. For almost all Boolean functions from $P_2(n)$ the value p from theorem 1 asymptotically equals $|S|$ and in this case for some constant c such that $\sqrt[3]{2} < c < 2$ the circuit Σ' allows for at least $c^{2^n/n}$ pairwise non-equivalent variants of recovery.

It is important to note that circuits constructed for an arbitrary Boolean function f by means of the formula $\Phi = \Phi_r$ described earlier have smaller complexity compared with the Lupanov method. Meanwhile the specified earlier “recovery” is unambiguous for circuits constructed by means of the Lupanov’s method because of specificity of the corresponding ϕ -universal set G .

From the other side, the Shannon method, which in essence assumes the use of representation (5) for Boolean function f and the implementation of the set G of all different “residual” Boolean functions f_σ of the given representation, allows to achieve the desired “ambiguity” of the stated representation. However in this case the complexity of the circuit Σ that implements Boolean function f is essentially (by a constant multiplier) higher than the complexity of the analogous circuit constructed by means of the Lupanov method.

Notice that logic synthesis methods described earlier are oriented on arbitrary (“random”) Boolean functions. Hence these methods couldn’t be called “specialized”. If it is necessary to construct circuits for Boolean functions from a special class then the described methods may be combined with the Lupanov method of the local encoding [4] for the reduction of the specialized Boolean function systems synthesis problem to the problem of synthesis of “random” Boolean functions that depend on smaller number of variables. Thereby asymptotic bounds mentioned above gain actuality in real applications too.

3. An example of the protected design of DES cipher substitution box

In this section we give an example of the DES cipher substitution box (S-box) logical structure design obtained by means of universal function systems. The S-box is one of the main blocks of the DES cipher. It implements 4 Boolean functions of 6 variables from the given class of Boolean functions.

We assume that the resulting Boolean circuit is designed to achieve the given level of protection from functionality inference of the implemented Boolean function system when parts of the integrated circuit are being manufactured separately on an untrusted facility. Thereby on each ciphering round the DES cipher uses 8 different S-boxes each implementing its own set of Boolean functions. Note that substitution boxes with different parameters are used in most other block ciphers (for instance, AES, GOST 28147-89, Blowfish and etc.) In addition it is supposed that the intruder was able to sort out the block from general circuit structure and probably believes that the separated block is the part of the block cipher.

Let us construct the standard ϕ -universal set for the Boolean function $\varphi = y_1y_2 \vee y_3y_4 \vee y_5y_6 \vee y_7y_8$ based on selective partition $Y = Y_1 \cup Y_2 \cup Y_3 \cup Y_4 \cup Y_5$, where $Y_i = \{y_{2i-1}\}$, $i = \overline{1,4}$, and $Y_5 = \{y_2, y_4, y_6, y_8\}$. By G_{Y_j} , $j = \overline{1,5}$, we denote the component of the φ -universal set and by r_j we denote its cardinality. Note that parameters of the partition for ϕ -US were chosen based on the following two criteria: eligibility of the overall circuit complexity and simplicity of the circuit description in the example. Suppose each S-box depends on variables $(x_1, x_2, x_3, x_4, x_5, x_6)$, then main subcircuits of the designed block are as follows.

1. Circuit Σ_P that implements universal system $\vec{P}_2(x_4, x_5, x_6)$.
2. Circuit Σ_μ that is a union of $|G|$ multiplexors

$$\mu_3(x_1, x_2, x_3, t_1, t_2, t_3, t_4, t_5, t_6)$$

such that each of them implements the corresponding Boolean function from φ -universal set G if its data inputs receive necessary Boolean function of the universal Boolean function set $\vec{P}_2(x_4, x_5, x_6)$.

3. Circuit Σ_φ that is a union of 4 circuits such that each of them implements Boolean function φ and with the corresponding substitution of outputs of circuit Σ_μ implements corresponding Boolean function of the S -box.

The size-complexity of the S -box constructed this way is at most 36125. It is to be noted that the complexity estimation is rough and may be improved using various algorithms of logic structure local optimization for considered subcircuits and the use of local encoding approach.

Now we estimate the protection level from the functionality inference of the constructed circuit. We suppose that the split operation is used for functionality inference. There is an additional constraint that each section is made in predefined areas. In this case we consider the connections area between Σ_μ outputs and Σ_φ -inputs. The level of protection is a parameterized function of the initial Boolean function correct recovery probability. Introduce the notation:

1. $k^{(i)}$, $i = \overline{1,5}$, is the number of Σ_μ outputs that correspond to Boolean functions of the set G_i and that are taken to the upper level of the integrated circuit.
2. $k^{(j,r)}$, $j = \overline{1,8}$, $r = \overline{1,4}$, is the number of Σ_φ outputs that correspond to the j input of the element φ implementing r Boolean function of the S -box and that are taken to the upper circuit level (we suppose that input is detached from the corresponding output of circuit Σ_μ).
3. $p^{(r)}$ is the probability of the correct recovery of the r -th Boolean function of the S -box.

The probability $p^{(r)}$, $r = \overline{1, 4}$, satisfies the inequality

$$p^{(r)} \leqslant (k^{(5)})^{-\sum_{s=5}^8 k^{(s,r)}} \prod_{t=1}^4 (k^{(t)})^{-k^{(t,r)}}.$$

With an additional notation

$$k' = \min_r \sum_{s=1}^4 k^{(s,r)}, \quad k'' = \min_r \sum_{s=5}^8 k^{(s,r)}, \quad k''' = \min_i k^{(i)}$$

the probability p of the correct recovery of the whole S -box is estimated as follows:

$$p \leqslant \left((k^{(5)})^{k''} (k''')^{k'} \right)^{-4}.$$

The study was supported by the Academy of Cryptography of the Russian Federation.

References

- [1] Lozhkin S. A., “High-precision estimates for the complexity of control systems from some classes”, *Matem. voprosy kibernetiki*, **6** (1996), 189–214 (in Russian).
- [2] Lozhkin S. A., Shiganov A. E., “High accuracy asymptotic bounds on the BDD size and weight of the hardest functions”, *Fundamenta Informaticae*, **104**:3 (2010), 239–253.
- [3] Lupanov O. B., *Asymptotic estimates of complexity of control systems*, M.: Izdatel'stvo MGU, 1984 (in Russian).
- [4] Lupanov O. B., “On an approach to the synthesis of control systems — principle of local coding”, *Problemy kibernetiki*, **14** (1965), 31–110 (in Russian).