УДК 519.4

М.И. Рожков

О некоторых характеристиках булевых функций без запрета от четырех переменных в связи с построением биективных отображений специального вида

Понижающие пары натуральных чисел $(h,t), h>t$, для функций без запрета $f=f(x_1,x_2,...,x_k)$ изучались ранее автором в связи с построением биективных отображений

$$B_{f,L}:(F_2)^n\to(F_2)^n, B_{f,L}(x)=(f(x),f(\delta(x)),...,f(\delta^{n-1}(x)),x\in(F_2)^n,$$

набор координатных функций которых задается преобразованием $\delta=\delta_L$ регистра сдвига длины n с функцией обратной связи L, существенно зависящей от ограниченного числа $s(1)$ начальных и $s(2)$ конечных аргументов, и нелинейной функцией съема $f=f(x_1,x_2,...,x_k)$ от k аргументов ($k<<n$). Наличие понижающей пары (h,t) сводит исходную задачу проверки биективности $B_{f,L}$ при больших значениях длины регистра n к проверке биективности соответствующих отображений применительно к регистрам сдвига ограниченной длины

$$n=n_0\in\{t+s(1)+s(2)-1,t+s(1)+s(2),...,h+s(1)+s(2)-2\},$$

что позволяет эффективно использовать для ее решения вычислительную технику. В настоящей работе рассматриваются алгоритмы нахождения понижающих пар (h,t) для функций без запрета от четырех переменных.

Ключевые слова: ортогональные системы функций, регистр сдвига, фильтрующий генератор, понижающее множество.

Основные понятия и обозначения. Далее в работе будем придерживаться следующих основных понятий и обозначений: F_2 – поле из двух элементов $\{0,1\}$; $(F_2)^n$ – пространство двоичных векторов длины n; $(f_1,f_2,...,f_m)$ – задание отображения $(F_2)^n\to(F_2)^m$ в виде системы координатных функций

$$L(x_1,x_2,...,x_n)= L(x_1,x_2,...,x_{s(1)},x_{n-s(2)+1},x_{n-s(2)+2},...,x_n),$$

$L(x_1, x_2, \ldots, x_n) = L(x_1, x_2,\ldots, x_{s(1)}, x_{n-s(2)+1}, x_{n-s(2)+2}, \ldots, x_n)$ – функция обратной связи регистра сдвига длины n, линейная по переменной x_1 (т.е. $L(x_1,x_2,...,x_n)=x_1+\lambda(x_2,x_3,...,x_n)$) и существенно зависящая от ограниченного числа крайних переменных ($s(1)\geq 1$, $s(2)\geq 0$, $n\geq s(1)+s(2)$ – заданные параметры); $\delta=\delta_L$ – преобразование векторов пространства $(F_2)^n$, осуществляемое регистром сдвига с функцией обратной связи $L=L(x_1,x_2,...,x_n)$, действующее на вектор $\mathbf{x}=(x_1,x_2,...,x_n)\in(F_2)^n$ по правилу

$$\delta(x_1,x_2,...,x_n) = (x_2,x_3,...,x_n, L(x_1,x_2,...,x_n));$$

$f(x_1,x_2,...,x_k)$ – функция от $k\geq 3$ аргументов без запретов (являющаяся фильтрующей функцией съема с соответствующего регистра сдвига); $B_{f,L}$ – преобразование двоичных векторов длины n, задаваемое следующей системой координатных функций:

$$B_{f,L}(x)=(f(x),f(\delta(x)),...,f(\delta^{n-1}(x)),x\in(F_2)^n.$$

Отметим, что преобразование $B_{f,L}$ можно рассматривать как отображение множества начальных заполнений двоичного регистра сдвига длины n с обратной связью L в множество наборов n символов выходной последовательности, снимаемой с данного регистра с помощью функции f.

В работах [1–4] рассматриваются вопросы выбора нелинейной функции съема $f:(F_2)^n\to F_2$, а также функции обратной связи L, при которых отображение $B_{f,L}$ является биективным. При этом биективность отображения $B_{f,L}$ равносильна ортогональности системы его координатных функций.

В работе [2] показано, что при $n \geq 2^{k-1}+k-1$ отсутствие запретов у функции $f(x_1,x_2,...,x_k)$ является необходимым условием биективности отображения $B_{f,L}$ (функции без запрета называют также функциями без потери информации, сильно равновероятными, а также совершенно уравновешенными [5, 6]).

Известно (см. [5]), что для функции без запретов $f(x_1,x_2,...,x_k)$ при любом фиксированном выходном слове $\mathbf{Y} = y(1), y(2),..., y(n)$ длины $n \geq 1$ существует ровно 2^{k-1} входных слов $x_1,x_2,...,x_k,x_{k+1},...,x_{n+k-1}$ (их множество обозначим $f^{-1}(\mathbf{Y})$), перерабатываемых данной функцией в **Y** по закону

$$y(j) = f(x_j,x_{j+1},...,x_{j+k-1}), j = 1,2,...,n.$$

Биективность отображения $B_{f,L}$ равносильна тому, что среди 2^{k-1} входных слов множества $f^{-1}(\mathbf{Y})$ ровно одно слово будет удовлетворять ограничениям

$$x_{n+1} = L(x) = L(x_1,x_2,...,x_n), x_{n+2} = L(\delta_L(x)),...,x_{n+k-1} = L((\delta_L)^{k-2}(x)). \quad (1)$$

При этом $x_{n+1},...,x_{n+k-1}$ как функции от независимых переменных $x_1,x_2,...,x_n$ (в силу ограничений на вид функции обратной связи *L*) зависят лишь от $k + s(1) - 2$ начальных переменных и от $s(2)$ последних переменных. Таким образом, выполняется ограничение (1) или нет (для данного входного слова $x_1,x_2,...,x_k,x_{k+1},...,x_{n+k-1}$) зависит только от его начального отрезка $x_1,x_2,...,x_{k+s(1)-2}$ длины $k+s(1) - 2$ и конечного отрезка $x_{n-s(2)+1},...,x_n,x_{n+1},...,x_{n+k-1}$ длины $k + s(2) - 1$.

Для заданных функции $f = f(x_1, x_2, ..., x_k)$, натуральных r, $s \geq k-1$ и выходном слове $\mathbf{Y} = y(1), y(2),..., y(m)$ через $I = I(\mathbf{Y}) = I_{r,s}(\mathbf{Y})$ обозначим систему пар векторов

$$\{(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}) \mid i = 1,2,...,2^{k-1}\},$$

где $\boldsymbol{\alpha}^{(i)} = x_1, x_2,..., x_r$ и $\boldsymbol{\beta}^{(i)} = x_{m+k-s}, x_{m+k-s+1},..., x_{m+k-1}$ являются началом и концом входных слов $\mathbf{X} = x_1, x_2,..., x_k, x_{k+1},..., x_{n+k-1}$, перерабатываемых функцией *f* в выходное слово **Y**:

$$y(j) = f(x_j,x_{j+1},...,x_{j+k-1}), j = 1,2,...,m.$$

Так как число различных входов $X \in f^{-1}(\mathbf{Y})$, отвечающих заданному выходу **Y**, в точности равно 2^{k-1}, то полагаем, что $I(\mathbf{Y})$ состоит из 2^{k-1} элементов. При этом соответствующие системы $I(\mathbf{Y})$ и $I(\mathbf{Z})$ считаем равными ($I(\mathbf{Y}) = I(\mathbf{Z})$), если для любого $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in I(\mathbf{Y})$ данный элемент встречается в $I(\mathbf{Z})$ ровно столько раз, сколько он встречается в $I(\mathbf{Y})$.

Определение 1. Двоичные последовательности $\mathbf{Y} = y(1), y(2),..., y(n)$ и $\mathbf{Z} = z(1), z(2),..., z(m)$ назовем эквивалентными ($\mathbf{Y} \sim \mathbf{Z}$), если $I_{k-1,k-1}(\mathbf{Y}) = I_{k-1,k-1}(\mathbf{Z})$.

Определение 2. Пара натуральных чисел $(h,t), h > t$ называется понижающей парой для функции $f(x_1,x_2,...,x_k)$, если для любой последовательности **Y** длины *h* найдется эквивалентная ей последовательность **Z** длины *t*, причем каждая последовательность **Z** длины *t* эквивалентна некоторой последовательности **Y** длины *h*.

Известно [4], любая функция без запрета $f = f(x_1,x_2,...,x_k)$ обладает понижающей парой (*h, t*). Кроме того, если отображение $B_{f,L}$ биективно для

$$n = n_0 \in M = \{t + s(1) + s(2) - 1, t + s(1) + s(2),..., h + s(1) + s(2) - 2\},$$

тогда оно будет биективным при любом $n = n_0 + d \cdot (h - t), d = 0,1,...$. Тем самым вопрос о биективности отображений $B_{f,L}$ для всех достаточно больших n сводится к исследованию соответствующих отображений при ограниченных значениях n.

В работах [3–4] для некоторых функций f от $k \leq 6$ переменных были найдены понижающие пары путем полного перебора всех выходных слов **Y** длины $n = h$ и слов **Z** длины $m = t$, вычисления множеств $f^{-1}(\mathbf{Y})$ и $f^{-1}(\mathbf{Z})$ с последующим поиском для слов **Y** эквивалентных им слов **Z**. Слож-

ность данного метода оценивается величиной $O(2^{h+t})$ операций, и на его основе могут быть найдены пары (h,t) при сравнительно небольшой величине $(h+t)\approx 40$.

В настоящей работе предложены алгоритмы, позволяющие для функций от четырех переменных вычислять понижающие пары (h, t) и при большей величине $(h + t)$.

Для заданной функции без запретов $f = f(x_1, x_2, ..., x_k)$ через $R(f)$ обозначим множество функций

$$R(f) = \{f(\mathbf{x}), f(\mathbf{x})+1, f(\mathbf{x}+\mathbf{e}), f(\mathbf{x}+\mathbf{e})+1, f(s(\mathbf{x})), f(s(\mathbf{x}))+1, f(s(\mathbf{x})+\mathbf{e}), f(s(\mathbf{x})+\mathbf{e})+1\},$$

где $\mathbf{e}$ – двоичный вектор с единичными координатами (преобразование $\mathbf{x}+\mathbf{e}$ заключается в инвертировании координат двоичного вектора $\mathbf{x}$), $s(\mathbf{x}) = s(x_1, x_2, ..., x_k) = (x_k, x_{k-1}, ..., x_1)$.

При экспериментальных расчетах понижающих пар полезным является следующее утверждение.

Теорема 1 [4, утв. 5]. *Пусть функция без запрета $f = f(x_1, x_2, ..., x_k)$ обладает понижающей парой (h,t). Тогда (h,t) будет понижающей парой для любой функции $\varphi \in R(f)$.*

Алгоритм 1 (определение понижающей пары для функции, линейной по крайней переменной). Пусть $f(x_1, x_2, ..., x_k) = \varphi(x_1, x_2, ..., x_{k-1}) + x_k$. Известно [3], двоичные последовательности $\mathbf{Y} = y(1), y(2), ..., y(n)$ и $\mathbf{Z} = z(1), z(2), ..., z(m)$ являются эквивалентными, если и только если при любом $\boldsymbol{\alpha} \in (F_2)^{k-1}$

$$\delta_{y(n)}\delta_{y(n-1)}...\delta_{y(1)}(\alpha) =, \delta_{z(m)}\delta_{z(m-1)}...\delta_{z(1)}(\alpha) \qquad (2)$$

где $\delta_\varepsilon(x_1, x_2, ..., x_{k-1}) = (x_2, x_3, ..., x_{k-1}, \varphi(x_1, x_2, ..., x_{k-1}) + \varepsilon)$.

Алгоритм расчета понижающих пар (h,t) для функций рассматриваемого вида основан на том, что в соответствии с равенством (2) множество окончаний длины k–1 векторов из множества $f^{-1}(\mathbf{Y})$ задается набором из 2^{k-1} двоичных векторов длины k–1 каждый, т.е. двоичным вектором длины $(k-1)\cdot 2^{k-1}$. Другими словами, для фиксации всех возможных элементов множества $I_s = \cup I_{k-1,k-1}(\mathbf{Y})$ (объединение проводится по всем словам $\mathbf{Y}$ длины s) достаточно иметь массив

$$\text{ARRAY}[\,\omega\,] \text{ бит}, \ \omega = 2^r, r = (k-1)\cdot 2^{k-1},$$

в который по соответствующему адресу ставится метка, если адрес принадлежит множеству I_s. При этом если массив ARRAY_1 заполнен для слов длины s, для вычисления множества I_{s+1} (т.е. заполнения массива ARRAY_2) достаточно перебрать адреса массива ARRAY_1, по которым установлена специальная метка. Из каждого такого адреса, интерпретируемого как набор 2^{k-1} двоичных векторов длины k–1, путем применения отображений δ_φ и $\delta_{\varphi+1}$ к его компонентам, вычисляется два новых вектора-адреса, по которым в массив ARRAY_2 заносятся соответствующие метки. Это позволяет вычислять множество I_s за $O(s\cdot\omega)$ операций. Данная сложность при фиксированном k и $s \to \infty$ существенно меньше величины $O(2^s)$, которой оценивается сложность вычисления элементов множества I_s путем прямой обработки всех 2^s слов длины s. И соответственно для проверки понижающей пары (h,t) потребуется $O((t+h)\cdot\omega)$ операций и $O(\omega)$ бит оперативной памяти.

Замечание. В силу теоремы 1 рассмотренный выше алгоритм применим и для функций, линейных по первой переменной $f(x_1, x_2, ..., x_k) = \varphi(x_2, x_3, ..., x_k) + x_1$. С помощью указанного алгоритма были найдены понижающие пары для всех нелинейных функций без запрета от 4 переменных $f(x_1, x_2, x_3, x_4)$, которые линейны по одной из крайних переменных. При этом для фиксации элементов каждого из множеств I_h и I_t требуется память объема 16 Мбит.

Алгоритм 2 (определение понижающей пары для функции общего вида). Пусть $f(x_1, x_2, ..., x_k)$ – произвольная булева функция без запрета, $\boldsymbol{\alpha}$ – заданный двоичный вектор длины k–1. Для заданном выходном слове $\mathbf{Y}$ длины s соответствующая система векторов

$$I_{k-1,k-1}(\mathbf{Y}) = \{(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}) \mid i = 1,2,...,2^{k-1}\},$$

однозначно задается видом векторов $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ и числом их вхождения в систему $I_{k-1,k-1}(\mathbf{Y})$.

Таким образом, число различных систем $I_{k-1,k-1}(\mathbf{Y})$ не более числа сочетаний с повторениями из $2^{2(k-1)}$ элементов по 2^{k-1}, т.е. величины [7]

$$\omega = \frac{(2^{2(k-1)} + 2^{k-1} - 1)!}{(2^{k-1}!) \cdot ((2^{2(k-1)} - 1)!)}.$$

Будем считать, что множеству различных систем $I_{k-1,k-1}(\mathbf{Y})$ поставлено во взаимно однозначное соответствие множество целых чисел $0 \le j \le \omega$. Следовательно, применительно к функциям от k переменных для фиксации всех наборов $I_s = \cup I_{k-1,k-1}(Y)$ (объединение проводится по всем словам $\mathbf{Y}$ длины s) достаточно иметь массив ARRAY[ω] объема ω бит. При этом по адресу j ставится метка (бит «1»), если система $\{(\alpha^{(i)},\beta^{(i)}) | i = 1,2,...,2^{k-1}\}$ принадлежит множеству I_s (в противном случае по адресу j находится «0»).

Если заполнен массив ARRAY_1 для слов длины s, для вычисления множества I_{s+1} (т.е. заполнения массива ARRAY_2) перебираются адреса массива ARRAY_1, по которым установлена метка. Из каждого такого адреса, интерпретируемого как система $I_{k-1,k-1}(\mathbf{Y}) = \{(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}) \mid i = 1,2,...,2^{k-1}\}$, вычисляются две новых системы и соответствующие им адреса, по которым в массив ARRAY_2 заносятся соответствующие метки. Новые две системы строятся следующим образом. Первая система отвечает слову длины s+1, у которого последний знак равен 0. Вторая система отвечает случаю, когда последний знак равен 1. Пусть $(\boldsymbol{\alpha},\boldsymbol{\beta})$ встречается в исходной системе $d = d_{\alpha,\beta}$ раз, $\boldsymbol{\alpha} = (\delta_1,\delta_2,...,\delta_{k-1})$, $\boldsymbol{\beta} = (\theta_1,\theta_2,...,\theta_{k-1})$. Пусть при этом $(\alpha,\gamma) = 0$, $f(\theta_1,\theta_2,...,\theta_{k-1},\varepsilon) = 1$. Тогда в первой новой системе вектор $(\boldsymbol{\alpha},\boldsymbol{\gamma})$ встречается d раз, где $\boldsymbol{\gamma} = (\theta_2,\theta_3,...,\theta_{k-1},\varepsilon)$. И одновременно во второй новой системе d раз встретится вектор $(\boldsymbol{\alpha},\boldsymbol{\gamma}^{\bullet})$, где $\boldsymbol{\gamma}^{\bullet} = (\theta_2,\theta_3,...,\theta_{k-1},\varepsilon+1)$.

Замечание. Отметим, что разные вектора $\boldsymbol{\beta}$ исходной системы могут приводить к одинаковым векторам $\boldsymbol{\gamma}$ и $\boldsymbol{\gamma}^*$ в новой системе. Если же $f(\theta_1,\theta_2,...,\theta_{k-1},0) = f(\theta_1,\theta_2,...,\theta_{k-1},1) = 0$, тогда в первой новой системе вектор $(\boldsymbol{\alpha},\boldsymbol{\gamma})$, $\boldsymbol{\gamma} = (\theta_2,\theta_3,...,\theta_{k-1},0)$ встретится d раз и d раз встретится вектор $(\boldsymbol{\alpha},\boldsymbol{\gamma}^{\bullet})$, $\boldsymbol{\gamma}^{\bullet} = (\theta_2,\theta_3,...,\theta_{k-1},1)$. При этом во второй новой системе исходный вектор $(\boldsymbol{\alpha},\boldsymbol{\beta})$ ничего не порождает. Аналогичная ситуация возникает и при $f(\theta_1,\theta_2,...,\theta_{k-1},0) = f(\theta_1,\theta_2,...,\theta_{k-1},1) = 1$.

Указанный алгоритм позволяет вычислять множество I_s за $O(s \cdot \omega)$ операций. Данная сложность при $s \to \infty$ и фиксированном k существенно меньше величины $O(2^s)$, которой оценивается сложность вычисления элементов множества I_s путем прямой обработки всех 2^s слов длины s.

В частности, применительно к функции от 4 переменных для фиксации элементов каждого из множеств I_h и I_t требуется память объема $\omega = (71!)/((7!)\cdot(64!)) \cong 10^{10}$ бит.

Случай функции $f(x_1,x_2,...,x_k)$ при $k = 3$. Так как функции без запрета от трех переменных $f = f(x_1,x_2,x_3)$ являются линейными по одному из крайних переменных, то с учетом теоремы 1 множество понижающих пар нелинейных функций без запрета от k = 3 переменных задается понижающими парами (h, t) функций $f_1 = x_1x_2 + x_2 + x_3$, для которой (h, t) = (6,4), и $f_2 = x_1x_2 + x_3$, для которой (h, t) = (11,8).

Случай функции $f(x_1,x_2,...,x_k)$ при k=4. С учетом теоремы 1 совокупность понижающих пар (h, t) для нелинейных функций ($\deg(f) \ge 2$) без запрета от k = 4 переменных, которые существенно зависят от крайних аргументов и одновременно линейны хотя бы по одному из них, задается парами (h, t) для нижеприведенных в таблице первых 62 функций вида $f(x_1,x_2,x_3,x_4) = \varphi(x_1,x_2,x_3) + x_4$. При этом функция f приводится в форме многочлена Жегалкина, а вспомогательная функция φ посредством целого числа $\varphi = c$, задающего ее значения на векторах $(x_1,x_2,x_3) = x = x_1 + 2\cdot x_2 + 4\cdot x_3$ по формуле $\varphi(x) = (c >> x)\%2$, (здесь правая часть задается соответствующими операторами языка программирования СИ). Понижающие пары (h, t) для этих функций

при $h + t > 40$ были найдены с использованием идей алгоритма 1. При этом для фиксации элементов каждого из множеств I_h и I_t требуется память объема 16 Мбит.

В таблице приведены также представители всех 8 классов $R(f)$ нелинейных функций без запрета, которые существенно зависят от крайних аргументов и одновременно не являются линейными ни по одному из крайних аргументов. Это функции с порядковыми номерами с 63 по 70. Соответствующие результаты были получены экспериментальными методами путем выделения функций $f(x_1,x_2,x_3,x_4)$ с равномерным распределением выходных $2^{k-1}= 8$ грамм. Кроме того, последние три функции таблицы являются линейными. Для всех этих функций понижающие пары вычислялись путем прямой обработки всех выходных слов длин h и t, т.е. без использования идей алгоритма 2.

Для трех функций из таблицы, для которых прямой метод не привел к нахождению понижающих пар, практическая сложность реализации идей алгоритма 2 связана с необходимостью использования памяти объема $\omega =(71!)/((7!)\cdot(64!))\cong 10^{10}$ бит (для фиксации элементов каждого из множеств I_h и I_t).

Перечень функций и их понижающих пар

п/п	φ	F	(h, t)
1	*2*	*3*	*4*
1	2	$x_1 + x_1x_2 + x_1x_3 + x_1x_2x_3 + x_4$	9,6
2	4	$x_2 + x_1x_2 + x_2x_3 + x_1x_2x_3 + x_4$	11,9
3	6	$x_1 + x_2 + x_1x_3 + x_2x_3 + x_4$	36,27
4	8	$x_1x_2 + x_1x_2x_3 + x_4$	8,5
5	10	$x_1 + x_1x_3 + x_4$	36,24
6	14	$x_1 + x_2 + x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_4$	12,10
7	18	$x_1 + x_1x_2 + x_3 + x_2x_3 + x_4$	50,46
8	20	$x_2 + x_1x_2 + x_3 + x_1x_3 + x_4$	51,27
9	22	$x_1 + x_2 + x_3 + x_1x_2x_3 + x_4$	49,45
10	24	$x_1x_2 + x_3 + x_1x_3 + x_2x_3 + x_4$	23,17
11	26	$x_1 + x_3 + x_2x_3 + x_1x_2x_3 + x_4$	74,68
12	28	$x_2 + x_3 + x_1x_3 + x_1x_2x_3 + x_4$	18,15
13	30	$x_1 + x_2 + x_1x_2 + x_3 + x_4$	50,46
14	34	$x_1 + x_1x_2 + x_4$	21,15
15	36	$x_2 + x_1x_2 + x_1x_3 + x_2x_3 + x_4$	59,55
16	38	$x_1 + x_2 + x_2x_3 + x_1x_2x_3 + x_4$	45,39
17	42	$x_1 + x_1x_2x_3 + x_4$	34,31
18	44	$x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_4$	19,16
19	46	$x_1 + x_2 + x_1x_2 + x_2x_3 + x_4$	20,17
20	50	$x_1 + x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_4$	25,21
21	54	$x_1 + x_2 + x_3 + x_1x_3 + x_4$	67,63
22	58	$x_1 + x_3 + x_1x_3 + x_2x_3 + x_4$	47,37
23	62	$x_1 + x_2 + x_1x_2 + x_3 + x_1x_3 + x_1x_2x_3 + x_4$	28,24
24	66	$x_1 + x_1x_2 + x_1x_3 + x_2x_3 + x_4$	20,17
25	70	$x_1 + x_2 + x_1x_3 + x_1x_2x_3 + x_4$	30,29
26	74	$x_1 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_4$	48,33
27	78	$x_1 + x_2 + x_1x_2 + x_1x_3 + x_4$	61,56
28	86	$x_1 + x_2 + x_3 + x_2x_3 + x_4$	23,21
29	94	$x_1 + x_2 + x_3 + x_1x_2 + x_2x_3 + x_1x_2x_3 + x_4$	44,40
30	110	$x_1 + x_2 + x_1x_2 + x_1x_2x_3 + x_4$	115,95
31	126	$x_1 + x_2 + x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_4$	59,55
32	142	$x_1 + x_2 + x_1x_2 + x_1x_3 + x_2x_3 + x_4$	21,15
33	158	$x_1 + x_2 + x_3 + x_1x_2 + x_1x_2x_3 + x_4$	29,28
34	166	$x_1 + x_2 + x_2x_3 + x_4$	11,9
35	174	$x_1 + x_2 + x_1x_2 + x_2x_3 + x_1x_2x_3 + x_4$	44,38

Продолжение таблицы

1	*2*	*3*	*4*
36	178	$x_1 + x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_4$	12,8
37	182	$x_1 + x_2 + x_3 + x_1x_3 + x_1x_2x_3 + x_4$	42,38
38	186	$x_1 + x_3 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_4$	65,35
39	190	$x_1 + x_2 + x_3 + x_1x_2 + x_1x_3 + x_4$	37,25
40	194	$x_1 + x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_4$	25,22
41	198	$x_1 + x_2 + x_1x_3 + x_4$	53,43
42	202	$x_1 + x_1x_3 + x_2x_3 + x_4$	47,32
43	206	$x_1 + x_2 + x_1x_2 + x_1x_3 + x_1x_2x_3 + x_4$	31,26
44	210	$x_1 + x_3 + x_1x_2 + x_4$	28,22
45	212	$x_2 + x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_4$	18,15
46	214	$x_1 + x_2 + x_3 + x_2x_3 + x_1x_2x_3 + x_4$	65,53
47	218	$x_1 + x_3 + x_1x_2x_3 + x_4$	108,101
48	220	$x_2 + x_3 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_4$	15,13
49	222	$x_1 + x_2 + x_3 + x_1x_2 + x_2x_3 + x_4$	13,9
50	226	$x_1 + x_1x_2 + x_2x_3 + x_4$	23,20
51	228	$x_2 + x_1x_2 + x_1x_3 + x_4$	44,38
52	230	$x_1 + x_2 + x_1x_2x_3 + x_4$	82,75
53	232	$x_1x_2 + x_1x_3 + x_2x_3 + x_4$	12,8
54	234	$x_1 + x_2x_3 + x_1x_2x_3 + x_4$	63,53
55	236	$x_2 + x_1x_3 + x_1x_2x_3 + x_4$	31,27
56	238	$x_1 + x_2 + x_1x_2 + x_4$	61,56
57	242	$x_1 + x_3 + x_1x_2 + x_1x_3 + x_1x_2x_3 + x_4$	18,12
58	244	$x_2 + x_3 + x_1x_2 + x_2x_3 + x_1x_2x_3 + x_4$	12,11
59	246	$x_1 + x_2 + x_3 + x_1x_3 + x_2x_3 + x_4$	45,33
60	248	$x_3 + x_1x_2 + x_1x_2x_3 + x_4$	18,12
61	250	$x_1 + x_3 + x_1x_3 + x_4$	31,23
62	254	$x_1 + x_2 + x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3 + x_4$	15,11
63		$g_1(x) = x_2 + x_3 + x_1x_3 + x_1x_4 + x_1x_2x_3 + x_1x_2x_4$	15,12
64		$g_2(x) = x_1 + x_2 + x_3 + x_1x_2 + x_2x_4 + x_1x_3x_4$	18,12
65		$g_3(x) = x_3 + x_1x_2 + x_2x_3 + x_2x_4 + x_1x_2x_3 + x_1x_2x_4$	14,8
66		$g_4(x) = x_2 + x_3 + x_1x_4 + x_2x_4 + x_3x_4 + x_1x_3x_4 + x_2x_3x_4$	Не найдено
67		$g_5(x) = x_2 + x_3 + x_4 + x_1x_4 + x_2x_4 + x_1x_3x_4 + x_2x_3x_4$	Не найдено
68		$g_6(x) = x_1 + x_3 + x_2x_4 + x_1x_2x_4$	Не найдено
69		$g_7(x) = x_2 + x_1x_3 + x_1x_3x_4$	9,7
70		$g_8(x) = x_2 + x_3 + x_1x_3 + x_3x_4 + x_1x_3x_4$	6,5
71		$l_1(x) = x_1 + x_4$	6,3
72		$l_2(x) = x_1 + x_2 + x_4$	10,3
72		$l_3(x) = x_1 + x_2 + x_3 + x_4$	7,3

Заключение. В настоящей работе предложены алгоритмы вычисления понижающих пар для булевых функций без запрета от четырех переменных. Данная характеристика имеет важное значение для построения биективных отображений $B_{f,L}$, задаваемых регистром сдвига большой длины n с функцией обратной связи $L(x_1,x_2,...,x_n)$, которая зависит от ограниченного числа крайних переменных, и нелинейной функцией-фильтром $f = f(x_1,x_2,...,x_k)$ от небольшого числа переменных $k << n$.

На основе данных алгоритмов найдены понижающие пары для почти всех функций без запрета от четырех переменных. Ранее аналогичные результаты были известны только для некоторых таких функций.

Полученные результаты могут быть полезны при построении и обосновании статистических свойств датчиков случайных последовательностей на основе фильтрующих генераторов.

Литература

1. Саранцев А.В. Построение регулярных систем однотипных двоичных функций с использованием регистра сдвига // Лесной вестник. – 2004. – № 1 (32). – С. 164–169.

2. Рожков М.И. К вопросу построения ортогональных систем двоичных функций с использованием регистра сдвига // Лесной вестник. – 2011. – № 3 (79). – С. 180–185.

3. Рожков М.И. Ортогональные системы булевых функций на выходе фильтрующего генератора // Промышленные АСУ и контроллеры. – 2014. – № 1. – С. 31–36.

4. Рожков М.И. Биективные отображения, порождаемые фильтрующим генератором // Прикладная дискретная математика. – 2014. – № 1 (23). – С. 27–39.

5. Сумароков С. Н. Запреты двоичных функций и обратимость для одного класса кодирующих устройств // Обозрение прикл. и промышл. матем., сер. дискретн. матем. – 1994. – Т. 1, вып. 1. – С. 33–35.

6. Логачев О.А. Новые методы изучения совершенно уравновешенных булевых функций / О.А. Логачев, С.В. Смышляев, В.В. Ященко // Дискретная математика. – 2009. – Т. 22, № 2. – С. 51–74.

7. Холл М. Комбинаторика. – М.: Мир, 1970. – 424 с.

Рожков Михаил Иванович
Д-р техн. наук, канд. физ.-мат. наук, ст. науч. сотр., доцент каф. «Компьютерная безопасность» Национального исследовательского университета «Высшая школа экономики», Москва
Тел.: 8 (495) 916-35-04
Эл. почта: rozhkov.m.i@yandex.ru

Rozhkov M.I.

On some characteristics of Boolean functions without prohibition of four variables in connection with the construction of bijective mappings of a special type

In the work we consider the algorithms of lowering pairs finding for functions without prohibition of four variables. Lowering (or restrictive) pairs of natural numbers (h, t), $h > t$ for functions without prohibition was studied earlier by the author in connection with the construction of bijective mappings $B_{f,L}$ defined by the shift register of length n with feedback function L, essentially dependent on a limited number of initial and final arguments, and a nonlinear function of removal of k arguments ($k << n$).

Keywords: orthogonal system of Boolean functions, feedback shift register, filter generator, restrictive multitude.
