Серия «Математика»
2018. Т. 25. С. 144—158

Онлайн-доступ к журналу:
http://mathizv.isu.ru

ИЗВЕСТИЯ
Иркутского
государственного
университета

УДК 519.714.71
MSC 94C10
DOI https://doi.org/10.26516/1997-7670.2018.25.144

Алгоритм минимизации функций алгебры логики в классе обратимых схем Тоффоли

А. С. Францева
Иркутский государственный университет, Иркутск, Российская Федерация

Аннотация. Рассматривается задача представления функций алгебры логики обратимыми схемами, построенными из элементов Тоффоли. Интерес к данной задаче связан с актуальными исследованиями возможности организовать «холодные» вычисления с помощью дискретных преобразователей информации, т. е. такие вычисления, при выполнении которых технические устройства, их реализующие, не выделяли бы тепла.

Поскольку обратимые схемы реализуют в общем случае обратимые функции? в исследовании использован метод Тоффоли – Фредкина для представления функций алгебры логики обратимыми функциями.

В работе описывается алгоритм нахождения минимального представления функции алгебры логики в классе обратимых схем, построенных из элементов Тоффоли. Алгоритм использует полиномиальную нормальную форму функции алгебры логики в операторном представлении и задачу нахождения минимального представления функции алгебры логики в классе операторных пучков определенного вида. Выбранный класс операторных пучков соответствует классу расширенных поляризованных полиномов Жегалкина (далее расширенных полиномов), который включает в себя известный класс поляризованных полиномов Жегалкина.

В заключение приводятся вычислительные результаты алгоритма минимизации функций алгебры логики в классе обратимых схем.

Ключевые слова: обратимая схема, функции Тоффоли, функции алгебры логики, поляризованные полиномы Жегалкина.

1. Введение

В работе представлены результаты исследования задачи представления функций алгебры логики в классе обратимых схем.

Предварительно описывается представление функции алгебры логики в виде обратимой функции. Для этого используется метод Фредкина – Тоффоли, предложенный в [9]. Обратимые функции реализуются обратимыми схемами, построенными из элементов Тоффоли [10]. Понятие обратимой функции, способы ее задания, структурное описание обратимых схем, реализующих обратимые функции, и их функционирование подробно изложены в [2]. В [3] описан вид обратимых схем, реализующих такие обратимые функции, которые представляют функции алгебры логики. В работе приведены только ключевые определения необходимые для представления функций алгебры логики обратимыми схемами.

Для того чтобы функцию алгебры логики представить в виде обратимой схемы, составленной из элементов Тоффоли, требуется получить ее полиномиальную нормальную форму (ПНФ). Для этого в работе используется операторный подход, подробно изложенный в [4].

На основе введенных определений операторного подхода предлагается алгоритм 1 минимизации функций алгебры логики в одном из классов операторных пучков KE. Важно, что для работы данного алгоритма несущественным является выбор базисной функции, алгоритм использует только базисный пучок.

Алгоритм 2 минимизации функций алгебры логики в классе обратимых схем RS опирается на предыдущий алгоритм 1 и для его работы требуется выбрать в качестве базисной функции функцию конъюнкции, в результате чего класс операторных пучков KE ассоциируется с классом расширенных поляризованных полиномов Жегалкина ZhE. Класс ZhE содержит известный класс поляризованных полиномов Жегалкина (или форм Рида-Маллера) Zh. Сложность представлений функции в данном классе существенно меньше, чем в классе Zh, что, в свою очередь, влияет на сложность представлений в классе обратимых схем RS.

В заключение работы представлены вычислительные результаты алгоритма 2.

2. Представления функций алгебры логики обратимыми схемами Тоффоли

Для удобства в дальнейшем множество переменных $x_1, ..., x_n$ обозначим через $\tilde{x}$, а множество $x_0, x_1, ..., x_n$ – через $\tilde{x}_0$.

Обратимым представлением функции алгебры логики $f(\tilde{x})$ будем называть обратимую функцию следующего вида:

$$F(\tilde{x}_0) = (f_0(\tilde{x}_0), f_1(\tilde{x}_0), ..., f_n(\tilde{x}_0)), \text{ где}$$
$$f_0(x_0, x_1, ..., x_n) = x_0 \oplus f(\tilde{x}),$$

$$f_i(x_0, x_1, ..., x_n) = x_i,$$

по всем $1 \le i \le n$, т.е. функция $F(\tilde{x}_0)$ осуществляет следующее взаимно однозначное отображение:

$$F(\tilde{x}_0) : \ (x_0, x_1, ..., x_n) \to (x_0 \oplus f(\tilde{x}), x_1, ..., x_n). \tag{2.1}$$

Пусть даны две обратимые функции $F(\tilde{x}_0)$ и $G(\tilde{x}_0)$, которые представляют функции $f(\tilde{x})$ и $g(\tilde{x})$, соответственно. Тогда по определению суперпозиции обратимых функций [2] функция $H(\tilde{x}_0) = F(G(\tilde{x}_0))$ представляет функцию $h(\tilde{x}) = f(\tilde{x}) \oplus g(\tilde{x})$.

Рассмотрим множество T обратимых функций (называемых функциями Тоффоли [10]) следующего вида:

1) $T_0^{n+1}(x_i) : \ (x_0, x_1, ..., x_i, ..., x_n) \to (x_0, x_1, ..., \bar{x}_i, ..., x_n)$, $i \in \{0, 1, ..., n\}$;

2) $T_k^{n+1}(x_{i_1}, ..., x_{i_k}, x_0) : \ (x_0, x_1, ..., x_n) \to (x_0 \oplus x_{i_1} \cdot ... \cdot x_{i_k}, x_1, ..., x_n)$, $k > 0, \ \{i_1, ..., i_k\} \subseteq \{1, ..., n\}$.

Обратимой схемой Тоффоли будем называть обратимую схему, которая составлена из элементов, реализующих функции множества T.

Обратимые схемы Тоффоли реализуют обратимые функции вида (2.1).

Класс обратимых схем Тоффоли обозначается через RS.

Обратимую функцию $F(\tilde{x}_0)$ реализуют несколько обратимых схем Тоффоли. Они зависят от вида полиномиальной нормальной формы (ПНФ) соответствующей функции алгебры логики $f(\tilde{x})$. Для описания используемого в работе класса ПНФ применяется операторный подход, подробно изложенный в [4].

3. Операторное описание функций алгебры логики

Двоичные наборы длины n будем обозначать греческими буквами, например, так: $\tilde{\delta} = \delta_1 ... \delta_n$; нулевой набор: $0...0 = \tilde{0}$; единичный набор: $1...1 = \tilde{1}$.

Под оператором t, представленным в виде последовательности $t = t_1 ... t_n$, $t_i \in \{e, d, p\}$, будем понимать отображение $t : \ B_n \to B_n$ из линейного пространства всех функций алгебры логики B_n в B_n. Элементы t_i последовательности $t_1 ... t_n$ называются компонентами оператора t, а n – его длиной.

Компонент t_i оператора t действует на функцию $f(\tilde{x})$ по переменной x_i следующим образом:

$$t_i f(\tilde{x}) = \begin{cases} f(\tilde{x}), \text{ если } t_i = e, \\ f(x_1, ..., \bar{x}_i, ..., x_n), \text{ если } t_i = p, \\ f'_{x_i}(\tilde{x}), \text{ если } t_i = d. \end{cases}$$

где $f'_{x_i}(\tilde{x})$ — производная функции $f(\tilde{x})$ по переменой x_i.

Действие оператора $t = t_1...t_n$ на функцию $f(\tilde{x})$ по переменным $x_1, ..., x_n$ определяется индуктивно: $t(f(\tilde{x})) = t_1(t_2...t_n f(\tilde{x}))$.

Набор $T = (t^{\tilde{0}}, ..., t^{\tilde{\tau}}, ..., t^{\tilde{1}})$ из 2^n операторов, где каждый оператор имеет длину n, называется пучком операторов. Пучок операторов $B = (b^{\tilde{0}}, ..., b^{\tilde{\tau}}, ..., b^{\tilde{1}})$ называется базисным, если существует функция $g(\tilde{x})$ такая, что операторные образы $b^{\tilde{0}}g(\tilde{x}), ..., b^{\tilde{\tau}}g(\tilde{x}), ..., b^{\tilde{1}}g(\tilde{x})$ образуют базис пространства B_n как линейного векторного пространства; функция $g(\tilde{x})$ называется базисной.

Операторный пучок $A = (a^{\tilde{0}}, ..., a^{\tilde{\tau}}, ..., a^{\tilde{1}})$ называется *однороднъм*, если существуют такие операторы $b = b_1...b_n$ и $c = c_1...c_n$, $b_i \neq c_i$ для любого i, что оператор $a^{\tilde{\tau}} = a_1...a_n$ пучка A определяется следующим образом:

$$a_i = \begin{cases} b_i, \text{ если } \tau_i = 0, \\ c_i, \text{ если } \tau_i = 1. \end{cases}$$

Любой однородный операторный пучок является базисным [4].

Оператор c будем называть суммой операторов a и b, если для любой функции алгебры логики $f(\tilde{x})$ верно следующее равенство: $cf(\tilde{x}) = af(\tilde{x}) \oplus bf(\tilde{x})$. Далее удобно записывать так: $c = a \oplus b$.

Для любого однородного операторного пучка $A = (a^{\tilde{0}}, ..., a^{\tilde{\tau}}, ..., a^{\tilde{1}})$ существует оператор c такой, что $c = \bigoplus_{\tilde{\tau} \in \{0,1\}^n} a^{\tilde{\tau}}$[4].

Пусть по некоторому однородному операторному пучку $A = (a^{\tilde{0}}, ..., a^{\tilde{\tau}}, ..., a^{\tilde{1}})$ функция $f(\tilde{x})$ имеет следующую операторную форму:

$$O(f) = \bigoplus_{\tilde{\tau} \in \{0,1\}^n} \alpha_{\tilde{\tau}} a^{\tilde{\tau}} g(\tilde{x}), \tag{3.1}$$

где $\alpha_{\tilde{\tau}} \in \{0, 1\}$.

В дальнейшем для удобства записи обозначим через $\bigoplus_{\tilde{\tau}}$ сумму вида $\bigoplus_{\tilde{\tau} \in \{0,1\}^n}$, а через $\bigcup_{\tilde{\delta}}$ — объединение вида $\bigcup_{\tilde{\delta} \in \{0,1\}^n}$.

В форме (3.1) для всех $\alpha_{\tilde{\tau}} = 1$ вместо оператора $a^{\tilde{\tau}}$ подставим сумму операторов соответствующего однородного операторного пучка $T^{\tilde{\tau}} = (t^{\tilde{\tau},\tilde{0}}, ..., t^{\tilde{\tau},\tilde{1}})$, проведем операцию сокращения пар одинаковых слагаемых и получим для функции $f(\tilde{x})$ операторное представление $SOF(f)$, называемое *специальной операторной формой*. В [1] доказано, что данное представление единственно с точностью до порядка слагаемых.

Через K_b, где b — оператор длины n, обозначается класс таких однородных операторных пучков $A = (a^{\tilde{0}}, ..., a^{\tilde{1}})$, в которых $a^{\tilde{0}} = b$.

Через H обозначается класс всех однородных операторных пучков, $H = \bigcup_b K_b$ (по всем операторам b).

Пусть $A = (a^{\tilde{0}}, ..., a^{\tilde{\tau}}, ..., a^{\tilde{1}})$ – однородный операторный пучок и $c = \bigoplus_{\tilde{\tau}} a^{\tilde{\tau}}$. Класс $HE_A = \bigcup_{\tilde{\delta}} \{T_{\tilde{\delta}}\} \bigcup \{A\}$ называется *расширением одно-*

родного пучка A, если пучки $T_{\tilde{\delta}} = (t^{\tilde{0}}, ..., t^{\tilde{\tau}}, ..., t^{\tilde{1}})$ построены следующим образом:

$$t^{\tilde{\tau}} = \begin{cases} a^{\tilde{\tau}}, \text{ если } \tilde{\tau} \neq \tilde{\delta}, \\ c, \text{ если } \tilde{\tau} = \tilde{\delta}. \end{cases} \tag{3.2}$$

Через HE обозначим класс *всех расширенных однородных операторных пучков*, $HE = \bigcup_{A \in H} HE_A$.

Если рассматривать в качестве базисной функции $g(\tilde{x})$ функцию произведения $x_1 \cdot ... \cdot x_n$, то классы операторных форм становятся классами полиномиальных нормальных форм [4]. В частности, класс H станет классом кронекеровых форм Kro, класс HE — классом расширенных кронекеровых форм $KroE$.

Пусть функция алгебры логики $f(\tilde{x})$ по некоторому однородному операторному пучку $A = (a^{\tilde{0}}, ..., a^{\tilde{1}}) \in H$ имеет операторную форму (3.1). По определению (3.2) по пучку $T_{\tilde{\delta}} = (t^{\tilde{0}}, ..., t^{\tilde{\tau}}, ..., t^{\tilde{1}}) \in HE_A$, $T_{\tilde{\delta}} \neq A$, функция $f(\tilde{x})$ имеет следующую операторную форму [3]:

$$O^+(f) = \bigoplus_{\tilde{\tau}} \bar{\alpha}_{\tilde{\tau}} b^{\tilde{\tau}} g(\tilde{x}) \oplus c g(\tilde{x}). \tag{3.3}$$

Сложность функции алгебры логики $f(\tilde{x})$ в классе H однородных операторных пучков определяется следующим образом:

$$L_H(f) = \min_{O(f)}(L(O(f)))$$

по всем операторным формам $O(f)$ вида (3.1), представляющим функцию $f(\tilde{x})$; $L(O(f))$ — сложность операторной формы $O(f)$, равная числу ненулевых коэффициентов $\alpha_{\tilde{\tau}}$ в разложении (3.1).

В классе расширенных однородных операторных форм HE сложность функции $f(\tilde{x})$ в соответствии с (3.3) равна:

$$L_{HE}(f) = \min_{O(f)}(\min\{L(O^+(f)), L(O(f))\}) = \\ = \min_{O(f)}(\min\{2^n - L(O(f)) + 1, L(O(f))\}).$$

В работе рассматривается класс операторных пучков $K_{d...d} \in H$, который при $g(\tilde{x}) = x_1 \cdot ... \cdot x_n$ соответствует классу поляризованных полиномов Жегалкина $Zh = \bigcup_{\tilde{\sigma}} Zh_{\tilde{\sigma}}$. Функции базиса $Zh_{\tilde{\sigma}}$ представляют собой операторные образы $a^{\tilde{\sigma},\tilde{\tau}}(x_1 \cdot ... \cdot x_n)$, где операторы $a^{\tilde{\sigma},\tilde{\tau}}$ по всем $\tilde{\tau} \in \{0,1\}^n$ составляют однородный операторный пучок $A = (d...d, ..., a_1...a_n)$, в котором

$$a_i = \begin{cases} e, \text{ если } \sigma_i = 1, \\ p, \text{ если } \sigma_i = 0. \end{cases}$$

Класс $KE = \bigcup_{A \in K_{d...d}} HE_A \subset HE$ соответствует *классу расширенных поляризованных полиномов Жегалкина* $ZhE = \bigcup_{\tilde{\sigma}} Zh_{\tilde{\sigma}}E$, где $Zh_{\tilde{\sigma}}E = \bigcup_{\tilde{\delta}}\{Zh^{\tilde{\delta}}_{\tilde{\sigma}}E\}\bigcup\{Zh_{\tilde{\sigma}}\}$ и функции базиса $Zh^{\tilde{\delta}}_{\tilde{\sigma}}E$ есть образы операторов пучка $T_{\tilde{\delta}}$ по функции $x_1 \cdot ... \cdot x_n$.

4. Алгоритм 1 минимизации функций алгебры логики в классе операторных пучков KE

В алгоритме вычисляется сложность функций алгебры логики в классе операторных пучков KE. Для работы алгоритма предварительно требуется выполнить следующие действия:

1) Сгенерировать вспомогательную библиотеку, содержащую информацию о классе операторных пучков $K_{d...d}$.

2) Построить специальную операторную форму $SOF(f)$ функции f.

Построение библиотеки класса $K_{d...d}$. Рассмотрим однородные операторные пучки $A_{\tilde{\sigma}} = (a^{\tilde{\sigma},\tilde{0}}, ..., a^{\tilde{\sigma},\tilde{\tau}}, ..., a^{\tilde{1},\tilde{\tau}})$ класса $K_{d...d}$, $\tilde{\sigma} \in \{0,1\}^n$. По построению класса $K_{d...d}$ в качестве операторов $a^{\tilde{\sigma},\tilde{\tau}}$ пучков $A_{\tilde{\sigma}}$ по всем $\tilde{\sigma}, \tilde{\tau} \in \{0,1\}^n$, $\tilde{\tau} \neq \tilde{0}$ могут быть всевозможные операторы $t^i = t_1...t_n$ длины n, компоненты которых $t_j \in \{e, d, p\}$, $1 \leq j \leq n$. Упорядочим операторы t^i по натуральному порядку, поставив в соответствие оператору e число 0, оператору d число 1, оператору p число 2. Тогда каждому оператору t^i соответствует единственное число в троичной системе счисления, а индекс i является его десятичным номером и принадлежит множеству $\{0, 1, ..., 3^n - 1\}$.

Пучкам $A_{\tilde{\sigma}}$ соответствуют операторы $c^{\tilde{\sigma}} = \bigoplus_{\tilde{\tau}} a^{\tilde{\sigma},\tilde{\tau}}$, которые по всем $\tilde{\sigma} \in \{0,1\}^n$ составят однородный операторный пучок

$$C = (e...e, \ldots, p...p).$$

Построим бинарную матрицу P, строки которой обозначены всевозможными операторами вида t^i, а столбцы операторами $c^{\tilde{\sigma}}$ следующим образом: единицы в столбце $c^{\tilde{\sigma}}$ матрицы P соответствуют тем операторам t^i, которые совпадают с операторами $a^{\tilde{\sigma},\tilde{\tau}}$ соответствующего пучка $A_{\tilde{\sigma}}$; нули стоят на остальных местах каждого столбца.

Построение $SOF(f)$. По пучку $C = (e...e, \ldots, p...p)$ построим операторную форму $O(f) = c^1 g(\tilde{x}) \oplus ... \oplus c^k g(\tilde{x})$ функции $f(\tilde{x})$. На матрице P отметим те столбцы $c^{\tilde{\sigma}}$, которые совпадают с операторами c^j, $j \in \{1, 2, ...k\}$. Тогда для того чтобы получить специальную операторную форму $SOF(f)$ функции $f(\tilde{x})$, достаточно по каждой строке t^i, $i \in \{0, 1, ..., 3^n - 1\}$, осуществить сложение по модулю 2 элементов матрицы

по столбцам c^j. Если сумма равна 1, то t^i находится среди элементов множества операторов $M(SOF(f))$, образы которых составляют $SOF(f)$.

Пусть $M(SOF(f)) = \{b^1, ..., b^N\}$. Отметим на матрице P те строки t^i, которые соответствуют операторам b^m, $m \in \{1, 2, ...N\}$. Тогда сложность $L(O_{\tilde{\sigma}}(f))$ операторной фурмы $O_{\tilde{\sigma}}(f)$ функции $f(\tilde{x})$ по пучку $A_{\tilde{\sigma}}$ вычисляется так: *в столбце* $c^{\bar{\bar{\sigma}}}$ *сложить те элементы матрицы* P, *которые по строкам совпадают с операторами* b^m *из* $M(SOF(f))$ [7].

При $n = 1$ матрица P имеет вид:

$$P_1 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

Если сохранить порядок операторов e, d, p при построении операторов длины n, то матрица $P = P_n$ есть кронекерово произведение n матриц P_1 :

$$P_n = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \otimes ... \otimes \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

Матрица P имеет фрактальную структуру. Если числу 1 поставить в соответствие пиксель черного цвета, а числу 0 – пиксель белого цвета и повернуть изображение на 90 градусов, то вид матрицы будет таким, как показано на рис. 1.

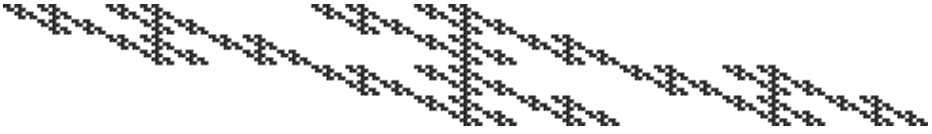

Рис. 1. Матрица P при $n = 5$, повернутая на 90 градусов

Основные шаги алгоритма 1:

1) Сгенерировать матрицу P.

2) Сгенерировать специальную операторную форму $SOF(f)$ функции $f(\tilde{x})$.

3) По всем $\tilde{\sigma} \in \{0, 1\}^n$:

   а) получить операторную форму $O_{\tilde{\sigma}}(f)$ функции $f(\tilde{x})$, используя специальную операторную форму $SOF(f)$ и матрицу P,

   б) вычислить сложности $L(O_{\tilde{\sigma}}(f))$, $L(O^+_{\tilde{\sigma}}(f)) = 2^n - L(O_{\tilde{\sigma}}(f)) + 1$ и выбрать минимальное значение

4) Найти $L_{KE}(f) = \min_{O_{\tilde{\sigma}}(f)}(\min\{2^n - L(O_{\tilde{\sigma}}(f)) + 1,\ L(O_{\tilde{\sigma}}(f))\})$.

Алгоритм 1. *integer* $MinBoolKE(f)$ {
vars
s, P : *integer array*;
$Cost$, Q, Ex : *integer*;
i, j : *integer*; // вспомогательные переменные;
$Cost \leftarrow J$; $gen(P)$; $s \leftarrow gen\_sof(P, f)$;
for $j \leftarrow 0$ *to* J { // по столбцам матрицы P, $\tilde{\sigma} \in \{0,1\}^n$;
$Q \leftarrow 0$;
for $i \leftarrow 0$ *to* I { // по строкам матрицы P;
if$(s_i \& P_{ij} = 1)$ $Q \leftarrow Q + 1$; }
$Ex \leftarrow J - Q + 1$;
if$(Q < Ex)$ { *if*$(Q < Cost)$ $Cost \leftarrow Q$; }
else { *if*$(Ex < Cost)$ $Cost \leftarrow Ex$; } }
return $Cost$; }

Замечания по работе алгоритма:

1. В алгоритме используются следующие программные функции и обозначения:

1) функция $gen(P)$ генерирует бинарную матрицу P, содержащую информацию о классе $K_{d\ldots d}$;

2) функция $gen\_sof(P, f)$; выводит в массив s данные о специальной операторной форме $SOF(f)$ функции $f(\tilde{x})$:

$$s_i = \begin{cases} 1, & \text{если } t^i \in M(SOF(f)), \\ 0, & \text{иначе.} \end{cases}$$

2. Для небольших значений n (зависит от возможности загрузки библиотеки в оперативную память) матрица P строится предварительно через кронекерово произведение матриц меньшей размерности. Свойства фрактальной структуры матрицы позволяют динамически строить необходимые столбцы матрицы, если невозможно загрузить всю библиотеку в оперативную память.

3. Бинарные массивы s и P представлены целочисленными массивами, меньшей размерности. Все операции над элементами этих массивов сводятся к использованию различных масок, логических операций и операций побитового сдвига.

5. Алгоритм 2 минимизации функций алгебры логики в классе обратимых схем RS

Пусть функция алгебры логики $f(\tilde{x})$ по некоторому пучку $A_{\tilde{\sigma}} \in K_{d...d}$ имеет следующую операторную форму:

$$O_{\tilde{\sigma}}(f) = \bigoplus_{\tilde{\tau}} \alpha_{\tilde{\tau}} a^{\tilde{\sigma},\tilde{\tau}}(x_1 \cdot ... \cdot x_n) = k_1 \oplus ... \oplus k_m, \tag{5.1}$$

m — число ненулевых коэффициентов $\alpha_{\tilde{\tau}}$.

Обратимая функция $F(\tilde{x}_0)$ представляет функцию $f(\tilde{x})$ по определению (2.1). Построим обратимую схему S_1, которая реализует функцию $F(\tilde{x}_0)$, в соответствии с операторной формой (5.1). Обратимая схема S_1 состоит из следующих элементов Тоффоли (рис. 2):

1) $N_{i_1}, ..., N_{i_l}$ расположены на входах $x_{i_1}, ..., x_{i_l}$, реализуют функции $T_0^{n+1}(x_{i_1}), ..., T_0^{n+1}(x_{i_l})$, соответствуют компонентам $\sigma_{i_1}, ..., \sigma_{i_l}$ вектора поляризации $\tilde{\sigma}$, равным 0;

2) $B_{n+1}^{k_1}, ..., B_{n+1}^{k_m}$ реализуют функции вида T_r^{n+1}, $r > 0$, которые представляют слагаемые $k_1, ..., k_m$;

3) $N_{i_1}, ..., N_{i_l}$ реализуют по определению (2.1) условия $f_i(\tilde{x}_0) = x_i$, $1 \le i \le n$.

Среди элементов $B_{n+1}^{k_1}, ..., B_{n+1}^{k_m}$ может быть элемент N_0, реализующий функцию $T_0^{n+1}(x_0)$, которая представляет слагаемое $d...d(x_1 \cdot ... \cdot x_1) = 1$.

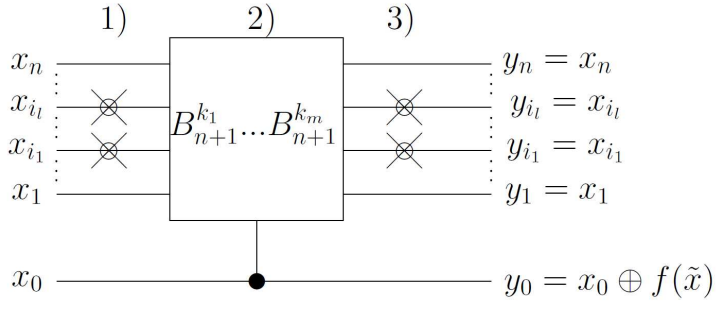

Рис. 2. Обратимая схема S_1 (функция $f(\tilde{x})$ имеет форму $O_{\tilde{\sigma}}(f)$)

Пусть функция алгебры логики $f(\tilde{x})$ по некоторому пучку $T_{\tilde{\delta}} \in KE$ имеет операторную форму:

$$O_{\tilde{\sigma}}^{+}(f) = \bigoplus_{\tilde{\tau}} \bar{\alpha}_{\tilde{\tau}} a^{\tilde{\sigma},\tilde{\tau}}(x_1 \cdot ... \cdot x_n) \oplus c(x_1 \cdot ... \cdot x_n) = \\ = \bar{k}_1 \oplus ... \oplus \bar{k}_{s-1} \oplus k_s^*, \ (s = 2^n - m + 1). \tag{5.2}$$

Построим обратимую схему S_2, которая реализует функцию $F(\tilde{x}_0)$, в соответствии с операторной формой (5.2). Обратимая схема S_2 состоит из следующих элементов Тоффоли (рис. 3):

элементы 1), 2), 3) реализуют слагаемые $\bar{k}_1, ..., \bar{k}_{s-1}$ аналогично обратимой схеме S_1.

4) $N_{j_1}, ..., N_{j_{n-l}}$ ($\{j_1, ..., j_{n-l}\} = \{1, 2, ..., n\} \backslash \{i_1, ..., i_l\}$) расположены на входах $x_{j_1}, ..., x_{j_{n-l}}$, реализуют функции $T_0^{n+1}(x_{j_1}), ..., T_0^{n+1}(x_{j_{n-l}})$, необходимые для представления слагаемого k_s^*, соответствуют компонентам $\sigma_{i_1}, ..., \sigma_{i_l}$ вектора поляризации $\tilde{\sigma}$, равным 1;

5) $B_{n+1}^{k_s^*}$ реализует функцию вида $T_n^{n+1}(x_1, ..., x_n, x_0)$, которая представляют слагаемое k_s^*;

6) $N_{j_1}, ..., N_{j_{n-l}}$ реализуют условия $f_i(\tilde{x}_0) = x_i$, $1 \le i \le n$.

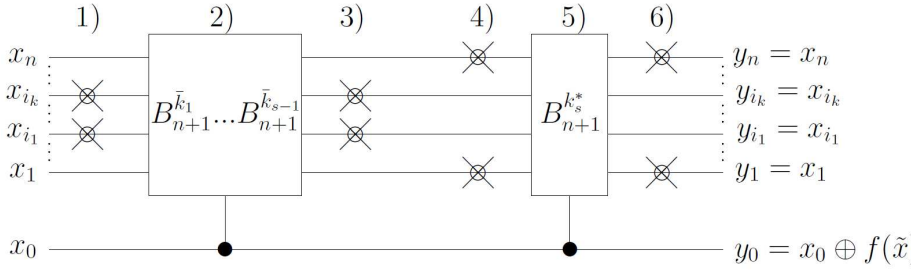

Рис. 3. Обратимая схема S_2 (функция $f(\tilde{x})$ имеет форму $O_{\tilde{\sigma}}^{+}(f)$)

Основные шаги алгоритма 2

1) Выполнить шаги 1) и 2) алгоритма 1.

2) По всем $\tilde{\sigma} \in \{0, 1\}^n$:

- вычислить сложность $L(O_{\tilde{\sigma}}(f))$ операторной формы $O_{\tilde{\sigma}}(f)$, используя специальную операторную форму $SOF(f)$ функции $f(\tilde{x})$ и матрицу P,
- вычислить сложность обратимой схемы S_1 : $L(S_1) = L(O_{\tilde{\sigma}}(f)) + 2 \cdot \omega(\tilde{\sigma})$, где $\omega(\tilde{\sigma})$ равно разности n и веса вектора поляризации $\tilde{\sigma}$;
- вычислить сложность обратимой схемы S_2 : $L(S_2) = 2^n - L(O_{\tilde{\sigma}}(f)) + 1 + 2 \cdot n$ [3];
- выбрать минимальное из $L(S_1)$, $L(S_2)$.

3) Найти $L_{RS}(f) = \min_{O_{\tilde{\sigma}}(f)}(\min\{L(S_1), L(S_2)\})$.

Алгоритм 2. *integer MinBoolRS*(f) {
vars
s, P : *integer array*;

$Cost$, T, N, Ex : $integer$;
a, i, j : $integer$; // вспомогательные переменные;
$Cost \leftarrow 2^n + 2n + 1$; $gen(P)$; $s \leftarrow gen\_sof(P, f)$; $a \leftarrow fict(f)$;
$for\ j \leftarrow 0\ to\ 2^n$ { // по столбцам матрицы P, $\tilde{\sigma} \in \{0,1\}^n$;
$T \leftarrow 0$;// подсчет элементов T_r^{n+1}, $r > 0$;
$for\ i \leftarrow 0\ to\ 3^n$ { // по строкам матрицы P;
$if(s_i \& P_{ij} = 1)\ T \leftarrow T + 1$; }
$N \leftarrow 2 \cdot (w(i) - w(i\&a))$; // подсчет элементов T_0^{n+1};
$Ex \leftarrow 2^n - T + 1 + 2 \cdot n$; // сложность схемы S_2;
$if((T + N) < Ex)$ { $if((T + N) < Cost)\ Cost \leftarrow T + N$;}
$else$ { $if(Ex < Cost)\ Cost \leftarrow Ex$; } }
$return\ Cost$; } [6]

Замечания:

1) функция $fict(f)$ выводит число, в двоичной записи которого на позиции i стоит 1, если переменная x_i в функции $f(\tilde{x})$ фиктивна и 0 – иначе;

2) функция $w(i)$ вычисляет количество единиц в бинарной записи целого положительного числа i.

6. Вычислительные результаты алгоритма 2

Выполнение вычисления сложности для всех функций алгебры логики осуществлено при $n = 3, 4, 5$.

При $n = 6, 7, 8, 9, 10$ алгоритм выполнялся на выборке случайно сгенерированных функций, которая была получена с помощью линейного конгруэнтного метода с коэффициентами a =6364136223846793005, c =1442695040888963407 и $m = 2^{64} - 1$ [8].

Вычислительные результаты представлены в графической форме в виде диаграмм, которые изображены на рисунках 4, 5 6.

С помощью алгоритма 1 была найдена последовательность множеств $M_n = \{p_n, q_n, t_n\}$ функций алгебры логики:

$$p_3 = (00011011),\ q_3 = (11010001),\ t_3 = (11001010),$$
$$p_n(x_1, ..., x_n) = x_n q_{n-1}(x_1, ..., x_{n-1}) \oplus \bar{x}_n p_{n-1}(x_1, ..., x_{n-1}),$$
$$q_n(x_1, ..., x_n) = x_n t_{n-1}(x_1, ..., x_{n-1}) \oplus \bar{x}_n q_{n-1}(x_1, ..., x_{n-1}),$$
$$t_n(x_1, ..., x_n) = x_n p_{n-1}(x_1, ..., x_{n-1}) \oplus \bar{x}_n t_{n-1}(x_1, ..., x_{n-1}).$$

Для этих функций удалось доказать, что они являются самыми сложными в классе ZhE [3]. Значения сложности представлений функций $f_i \in M_n$ до 10 переменных в классе обратимых схем RS представлены в таблице 1.

Все расчеты выполнены на оборудовании центра коллективного пользования «Иркутский суперкомпьютерный центр СО РАН» [5].

Таблица 1

Сложности функций $f_i \in M_n$ в классе обратимых схем RS

n	3	4	5	6	7	8	9	10
p_n	4	8	16	32	64	128	256	512
q_n	5	7	17	31	65	127	257	511
t_n	3	9	15	33	63	129	255	513

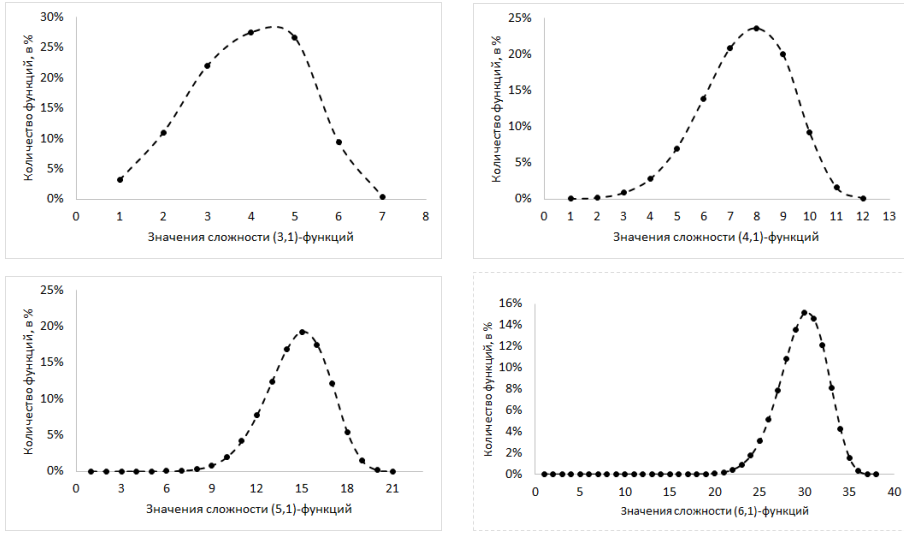

Рис. 4. Диаграммы распределения значений сложности функций алгебры логики при $n = 3, 4, 5, 6$

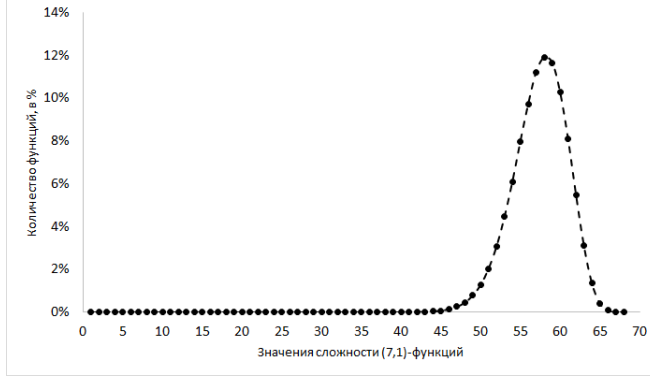

Рис. 5. Диаграмма распределения значений сложности функций алгебры логики при $n = 7$

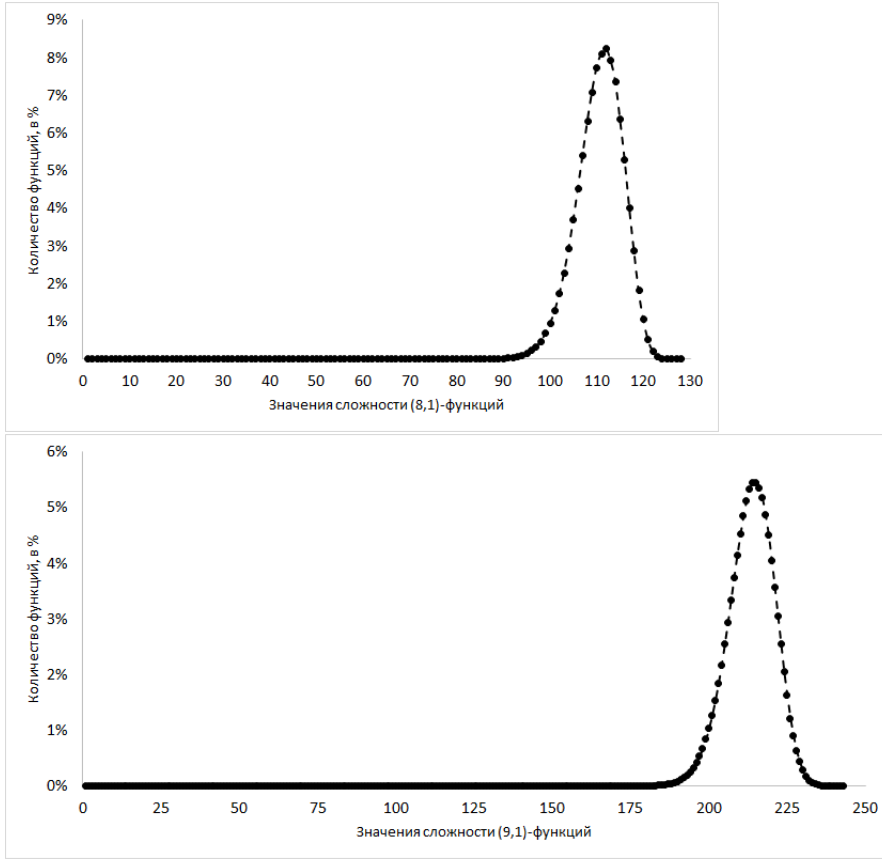

Рис. 6. Диаграммы распределения значений сложности функций алгебры логики при $n = 8, 9$

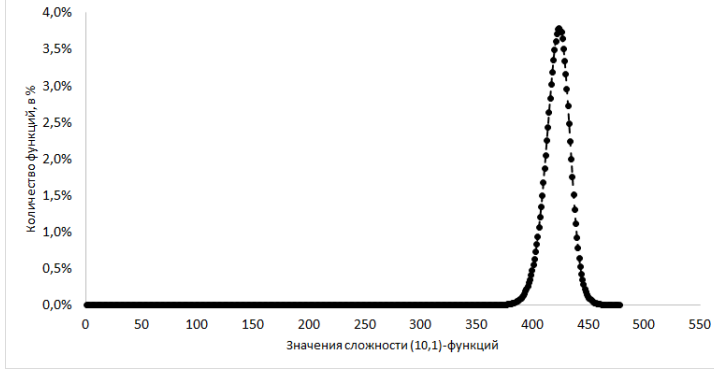

Рис. 7. Диаграмма распределения значений сложности функций алгебры логики при $n = 10$

Список литературы

1. Винокуров С. Ф., Казимиров А. С. Перечисление операторных классов булевых функций // Изв. Иркут. гос. ун-та. Сер. Математика. 2009. Т. 2, № 2. С. 40–55.
2. Винокуров С. Ф., Францева А. С. Приближенный алгоритм вычисления сложности обратимой функции в базисе Тоффоли // Изв. Иркут. гос. ун-та. Сер. Математика. 2011. Т. 4, № 4. С. 12–26.
3. Винокуров С. Ф., Францева А. С. Сложность представлений многовыходных функций алгебры логики // Изв. Иркут. гос. ун-та. Сер. Математика. 2016. Т. 16. С. 30–42.
4. Избранные вопросы теории булевых функций : монография / под ред. С. Ф. Винокурова, Н. А. Перязева. М. : Физматлит, 2001. 192 с.
5. Иркутский суперкомпьютерный центр СО РАН [Электронный ресурс] : сайт. Иркутск : ИДСТУ СО РАН. URL: http://hpc.icc.ru (дата обращения: 05.05.2018).
6. Пат. № 2017619310 Российская Федерация. Программа построения минимального представления многовыходных булевых функций в классе обратимых схем / С. Ф. Винокуров, Л. В. Рябец, А. С. Францева ; опубл. 22.08.17. Бюл. № 9.
7. Рябец Л. В., Винокуров С. Ф. Алгоритм точной минимизации булевых функций в классе кронекеровых форм // Алгебра и теория моделей 4. Новосибирск, 2003. С. 148–159.
8. Knuth D. E. MMIXware: A RISC Computer for the Third Millennium // Springer Heidelberg New York Dordrecht London. 2003. LNCS Sublibrary: SL 2 – Programming and Software Engineering. 550 p. DOI: 10.1007/3-540-46611-8.
9. Fredkin E., Toffoli T. Conservative Logic // International Journal of Theoretical Physics. 1982. Vol. 21, Iss. 3. P. 219–253. DOI: 10.1007/BF01857727.
10. Toffoli T. Reversible Computing // Automata, Languages and Programming (Series: Lecture Notes in Computer Science). 1980. Vol. 85. P. 632–644. DOI: 10.1007/3-540-10003-2_104.

Анастасия Сергеевна Францева, Педагогический институт, Иркутский государственный университет, Российская Федерация, 664003, г. Иркутск, ул. К. Маркса, 1, тел.: (3952)200739
(e-mail: a.s.frantseva@gmail.com)

Поступила в редакцию 10.08.18

An Algorithm for Minimization of Boolean Functions in the Class of Toffoli Reversible Logic Circuits

A. S. Frantseva

Irkutsk State University, Irkutsk, Russian Federation

Abstract. In this paper, the problem of Boolean function's representation by the reversible circuits constructed of the Toffoli gates is considered. Interest in this problem is connected with actual studies of the possibility for realization of "cold" computations. It means that when performing such computations, there is no heat dissipation.

In general, reversible circuits realize reversible functions. Therefore, Toffoli-Fredkin's method for representation of the Boolean function by the reversible function is used.

In work, an algorithm for finding the minimal representation of the Boolean function in a class of the reversible circuits, which are constructed from Toffoli elements is described. The algorithm uses the polynomial normal forms or exclusive-or sum-of-products expressions (ESOPs) of the Boolean function in the operator representation and the problem of finding the minimal representation of the Boolean function in the certain class of operator bundles. The chosen class of operator bundles corresponds to a class of the extended polarized Zhegalkin polynomials, which includes a well-known class of the polarized Zhegalkin polynomials or Reed-Muller forms.

In conclusion, the computational results of the algorithm for minimizing the Boolean functions in the class of reversible circuits are given.

Keywords: reversible circuit, Toffoli functions, Boolean functions, polarized Zhegalkin polynomials or Reed-Muller forms.

References

1. Vinokurov S.F., Kazimirov A.S. Enumeration of operator classes of Boolean functions. *The Bulletin of Irkutsk state University. Series Mathematics*, 2009, vol. 2, no. 2, pp. 40-55. (in Russian)
2. Vinokurov S.F., Frantseva A.S. An approximate algorithm for computing the complexity of reversible functions in the basis of Toffoli. *The Bulletin of Irkutsk state University. Series Mathematics*, 2011, vol. 4, no. 4, pp. 12-26. (in Russian)
3. Vinokurov S.F., Frantseva A.S. The Complexity of the Representation of Multiple-Output Boolean Functions. *The Bulletin of Irkutsk state University. Series Mathematics*, 2016, vol. 16, pp. 30–42. (in Russian)
4. *Izbrannye voprosy teorii bulevykh funktsii* [Selected problems of the theory of Boolean functions]. Edited by Vinokurov S.F., Peryazev N.A. Moscow, FIZMATLIT Publ., 2001, 192 p.
5. *Irkutskii superkompiuternyi tsentr SO RAN* [The Irkutsk Supercomputer Center of Siberian Branch of the Russian Academy of Science]. Available at: http://hpc.icc.ru (date of access: 05.05.2018).
6. Vinokurov S.F., Frantseva A.S., Ryabets L.V. *Programma postroeniia minimalnogo predstavleniia mnogovykhodnykh bulevykh funktsii v klasse obratimykh skhem* [Program for Constructing Minimal Representations of Multiple-output Boolean Functions in The Reversible Logic Circuits]. Patent RF, no. 2017619310, 2017. (in Russian)
7. Ryabets L.V., Vinokurov S.F. Algoritm tochnoi minimizatsii bulevykh funktsii v klasse kronekerovykh form [An Algorithm of Exact Minimization of Boolean Functions in the Class of Kronecker Forms]. *Algebra i teoriya modeley* [Algebra and theory of models], Novosibirsk, 2003, vol. 4, pp. 148–159. (in Russian)
8. Knuth D.E. MMIXware: A RISC Computer for the Third Millennium. *Springer Heidelberg New York Dordrecht London, LNCS Sublibrary: SL 2 – Programming and Software Engineering*, 2003, 550 p. https://doi.org/ 10.1007/3-540-46611-8
9. Fredkin E., Toffoli T. Conservative Logic. *International Journal of Theoretical Physics*, 1982, vol. 21, iss. 3, pp. 219-253. https://doi.org/ 10.1007/BF01857727
10. Toffoli T. Reversible Computing. *Automata Languages and Programming (Series: Lecture Notes in Computer Science)*, 1980, vol. 85, pp. 632-644. https://doi.org/ 10.1007/3-540-10003-2_104

Anastasiya Frantseva, Pedagogical Institute, Irkutsk State University, 1, K. Marx st., Irkutsk, 664003, Russian Federation, tel.: (3952)200739 (e-mail: a.s.frantseva@gmail.com)

Received 10.08.18