Известия вузов. Математика
2014, №5, с. 38–52

http://old.kpfu.ru/journals/izv_vuz/
e-mail: izvuz.matem@kpfu.ru

И.С. СЕРГЕЕВ

ВЕРХНИЕ ОЦЕНКИ СЛОЖНОСТИ ФОРМУЛ ДЛЯ СИММЕТРИЧЕСКИХ БУЛЕВЫХ ФУНКЦИЙ

Аннотация. Показано, что сложность реализации оператора подсчета числа единиц в булевом наборе длины n формулами над базисом B_2 двуместных булевых функций не превосходит $n^{3.03}$, а над стандартным базисом B_0 — $n^{4.47}$. Как следствие, такие же оценки справедливы для сложности любой пороговой симметрической функции n переменных, в частности, для функции голосования. Для сложности произвольной симметрической функции n переменных получены оценки $n^{3.04}$ и $n^{4.48}$ над базисами B_2 и B_0 соответственно. Доказательство основано на применении модулярной арифметики.

Ключевые слова: сложность формул, симметрические булевы функции, функция голосования, умножение.

УДК: 519.714

1. Введение

В работе рассматривается сложность реализации симметрических булевых функций формулами над базисом B_2 всех двуместных булевых функций и над стандартным базисом $B_0 = \{\wedge, \vee, \bar{}\}$. (Функция называется симметрической, если ее значения сохраняются при любых перестановках значений аргументов; в булевом случае это определение эквивалентно тому, что значения функции зависят только от арифметической суммы аргументов.)

Напомним, что множество формул над базисом B, сложность формулы, глубина формулы и функция, реализуемая формулой, определяются индуктивно следующим образом:

0) символы переменных являются формулами сложности 1, глубины 0 и реализуют соответствующие тождественные функции;

1) выражение $G(F_1, \ldots, F_k)$, где G — символ, обозначающий k-местную функцию $g \in B$, а F_i — формула сложности L_i и глубины D_i, реализующая функцию f_i, является формулой сложности $L_1 + \cdots + L_k$, глубины $\max\{D_1, \ldots, D_k\} + 1$ и реализует функцию $g(f_1, \ldots, f_k)$.[1]

Если базис B состоит из не более чем двуместных булевых функций, то используют компактное правило записи формулы: $(F_1 \circ F_2)$, где $\circ$ обозначает двуместную операцию

Поступила 07.11.2012

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проекты №№ 11-01-00508, 11-01-00792-а) и программы фундаментальных исследований Отделения математических наук РАН “Алгебраические и комбинаторные методы математической кибернетики и информационные системы нового поколения” (проект “Задачи оптимального синтеза управляющих систем”).

[1]Сложность и глубина формул могут определяться иначе: например, сложность — как число (сумма весов) базисных функций, используемых при построении формулы, в глубине — не учитывают одноместные функции. Выводы данной работы остаются в силе при указанных вариациях определений.

базиса B, или $\overline{F}_1$ в случае одноместной операции отрицания. При этом скобки опускают в тех случаях, когда приоритет операций определен или не важен.

Сложность $L_B(f)$ реализации булевой функции f формулами над базисом B определяется как минимум сложности формул, реализующих f. Сложность $L_B(K)$ класса (множества) функций K определяется как $\max_{f\in K} L_B(f)$. Формула, реализующая булев оператор, определяется как совокупность формул, реализующих отдельные функции — компоненты оператора. Сложность булевого оператора определяется как сумма сложностей его компонент. Более подробно введенные понятия обсуждаются в [1]–[6] (там же см. понятие схемы из функциональных элементов, которое встречается ниже, но несущественно для изложения основных результатов).

Обозначим через S_n класс всех симметрических булевых функций n переменных. Пусть T_n^k обозначает пороговую симметрическую функцию n переменных с порогом k; по определению, $T_n^k(x_1,\dots,x_n) = (x_1+\dots+x_n \ge k)$. Функция $T_n^{n/2}$ также называется функцией голосования.

Рекордные на сегодня верхние оценки сложности и глубины реализации симметрических функций как формулами, так и схемами из функциональных элементов над полными базисами, опираются на эффективную реализацию булевого (n,m)-оператора $C_n(x_1,\dots,x_n) = (C_{n,m-1},\dots,C_{n,0})$ подсчета числа единиц в булевом наборе $(x_1,\dots,x_n)$, где $m = \lceil \log_2(n+1)\rceil$. Сведение к вычислению C_n используется при минимизации глубины и сложности формул для умножения двоичных чисел.

Предварительное представление о сравнительной сложности реализации функции $T_n^{n/2}$, оператора C_n и класса функций S_n дают следующие известные или легко выводимые оценки.

Теорема 1. *Для любого полного конечного базиса B справедливы соотношения*[2]

$$L_B(C_n) \preceq \log n \cdot L_B(S_n), \quad L_B(T_n^{n/2}) \preceq L_B(S_n),$$
$$L_B(T_n^{n/2}) \preceq L_B(C_{2n}), \quad L_B(S_n) \preceq \frac{n}{\log n \log\log n}\cdot L_B(C_n),$$
$$L_B(C_n) \preceq \sum_{i=0}^{n} L_B(T_n^k), \quad L_B(S_n) \preceq \sum_{i=0}^{n} L_B(T_n^k), \quad L_B(T_n^k) \preceq L_B(T_{2n}^n).$$

Доказательство. Первые три неравенства очевидны: компоненты оператора C_n и функция голосования являются симметрическими функциями, функция $T_n^{n/2}$ является подфункцией функции $C_{2n,m}$, $m = \lfloor \log_2 n \rfloor$. Следующие три соотношения легко устанавливаются в случае $B = B_0$. Первое из них вытекает из того, что произвольная симметрическая функция представляется как функция компонент оператора C_n. Эту функцию можно реализовать асимптотически оптимальным методом О.Б. Лупанова ([1], § 14; [5], § 4.1), учитывая то, что более половины компонент C_n имеют сложность не выше $L_B(C_n)/\log n$ по порядку. Пятое и шестое неравенства вытекают из простых формул, выражающих компоненты оператора C_n и функции из S_n через пороговые симметрические функции T_n^k. При этом любая из функций T_n^k является подфункцией функции T_{2n}^n.

Для перехода к произвольному базису B используется соотношение $L_B(f) \le L + O(2^D)$, где L и D — сложность и глубина некоторой формулы F, реализующей f над базисом B_0. Это соотношение следует из того, что любая из функций $\overline{x}$, $x\vee y$, xy выразима формулой в базисе B, в которой переменные x и y не повторяются. Это вытекает из свойства полных

[2]Символ $\preceq$ обозначает неравенство по порядку.

базисов (наличие немонотонной и нелинейной функций) ([3], § 8.3; [4], § I.1.6). Остается проверить, что при подстановке в формулу F вместо функций базиса B_0 упомянутых выше формул над B, реализующих их, сложность формулы F возрастает не более чем на $O(2^D)$ (рост проистекает за счет несущественных переменных, присутствующих в замещающих формулах). □

При специальном выборе базиса, например, $B = S_l$, легко получить верхние оценки для $L_B(C_n), L_B(T_n^{n/2})$ вида $n^{1+\varepsilon(l)}$, где $\varepsilon(l) \to 0$ (см. также [7]). При достаточно больших l эти оценки очевидно близки к точным: известные нижние оценки имеют вид $L_B(T_n^{n/2}) \succeq n \log n$ ([8], см. также [5], § 4.2.2).

В случае $B \in \{B_0, B_2\}$ оценки теоремы 1 характеризуют соотношения между величинами $L_B(C_n)$, $L_B(T_n^{n/2})$ и $L_B(S_n)$ с удовлетворительной точностью ввиду большого расхождения между известными нижними и верхними оценками для каждой из этих величин.

А именно, наилучшие известные нижние оценки для функции $T_n^{n/2}$ (и, как следствие, для C_n и S_n) имеют вид $L_{B_0}(T_n^{n/2}) \succeq n^2$ (см. [9], а также [2], § 8.5; [5], § 4.5.1; [6], § 6.8) и $L_{B_2}(T_n^{n/2}) \succeq n \log n$ (см. [10], а также [2], § 8.7; [5], § 4.2.3).

Наилучшие верхние оценки для оператора C_n (и, как следствие, для $T_n^{n/2}$) были получены в работе [11] и имеют вид $L_{B_2}(C_n) \preceq n^{3.06}$, $L_{B_0}(C_n) \preceq n^{4.54}$. Они уточняют известные до этого оценки $L_{B_2}(C_n) \preceq n^{3.13}$, $L_{B_0}(C_n) \preceq n^{4.57}$ ([12]), $L_{B_2}(C_n) \preceq n^{3.32}$ ([13]), $L_{B_0}(C_n) \preceq n^{4.62}$ ([14]) (о более ранних результатах см. в [15]).

Начиная с работы [14], для вывода верхних оценок сложности симметрических функций вместо грубого соотношения теоремы 1 используются следующие утверждения.

Утверждение 1. *Пусть K — класс булевых функций n переменных, булев (m, n)-оператор $\xi(x_1, \ldots, x_n) = (\xi_1, \ldots, \xi_m)$ таков, что любая функция $f \in K$ представляется в виде $f = \varphi(\xi_1(x_1, \ldots, x_n), \ldots, \xi_m(x_1, \ldots, x_n))$. Тогда для любого полного конечного базиса B справедливо*

$$L_B(K) \preceq \sum_{i=1}^{m} 2^{m-i} L_B(\xi_i).$$

Доказательство. Функция φ как функция компонент оператора ξ реализуется методом разложения по переменным (метод каскадов) (см., например, [1], § 5; [2], § 4.3; [14], [4], § V.2.3). □

Следствие 1. Пусть $m = \lceil \log_2(n+1) \rceil$. Для любого полного конечного базиса B справедливо

$$L_B(S_n) \preceq \sum_{i=0}^{m-1} 2^{m-i} L_B(C_{n,i}).$$

С использованием специальных методов синтеза формул для оператора C_n (в котором его компоненты реализуются формулами различной сложности) получаются оценки $L_{B_2}(S_n) \preceq n^{3.23}$, $L_{B_0}(S_n) \preceq n^{4.82}$ ([11]), $L_{B_2}(S_n) \preceq n^{3.30}$, $L_{B_0}(S_n) \preceq n^{4.85}$ ([16])[3], $L_{B_2}(S_n) \preceq n^{3.37}$ ([13]), $L_{B_0}(S_n) \preceq n^{4.93}$ ([14]).

Аналогичный метод позволяет получать более точные, чем следуют из теоремы 1, оценки сложности пороговых симметрических функций, опираясь на следующее утверждение.

[3]Несколько лучшие оценки вытекают из [12].

Утверждение 2. *Пусть* $m = \lceil \log_2 k \rceil$. *Для любого полного конечного базиса* B *справедливо*

$$L_B(T_n^k) \preceq \sum_{i=0}^{m-1} L_B(C_{n,i}) + L_B(T_n^{2^m}).$$

Доказательство. Функция T_n^k реализуется как дизъюнкция функции $T_n^{2^m}$ и функции сравнения m-разрядного числа $[C_{n,m-1}, \ldots, C_{n,0}]$ с числом k. Последняя реализуется формулой линейной сложности и глубины $O(\log m)$ (см., например, [3], § 10.1; [5], § 2.5). □

Сведéние к реализации функции $T_n^{2^m}$ объясняется тем, что обычно эту функцию легко вычислить попутно с вычислением C_n. Методы [14], [16] позволяют одновременно получать оценки $L_B(S_n) \preceq n^{1+\alpha}$ и $L_B(T_n^k) \preceq kn^\alpha$ (при небольших значениях k из метода [16] извлекаются даже лучшие оценки сложности T_n^k).

Одним из наиболее популярных приложений оператора C_n является реализация умножения и многократного сложения чисел, хотя обычно речь идет о реализации схемами или программами, а не формулами. Обозначим через M_n булев $(2n, 2n-1)$-оператор умножения двоичных n-разрядных чисел, а через $\Sigma_{n,k}$ булев $(kn, k+m)$-оператор сложения n штук k-разрядных чисел, $m = \lceil \log_2 n \rceil$.

Утверждение 3. *Пусть* $m = \lceil \log_2 n \rceil$. *Для любого полного конечного базиса* B *справедливо*

$$L_B(\Sigma_{n,k}) \preceq (k+m) \log^{O(1)} n \cdot L_B(C_n), \quad L_B(M_n) \preceq L_B(\Sigma_{n,2n}).$$

Доказательство. Первое соотношение вытекает из неравенств

$$L_B(\Sigma'_{n,h}) \le L_B(C_n) \cdot L_B(\Sigma'_{m,h}), \quad L_B(C_n) \preceq n^{O(1)}, \quad L_B(\Sigma'_{n,h}) \preceq n^{O(1)},$$

где оператор $\Sigma'_{n,h}$ вычисляет первые h разрядов суммы n чисел. Первое из этих неравенств является интерпретацией "школьного" метода сложения чисел, второе доказано в [14], третье является следствием первого и второго. Второе соотношение утверждения вытекает из школьного метода умножения. □

Рассмотрим вопрос о реализации оператора C_n. Вероятно, все известные эффективные по сложности и глубине формулы и схемы для C_n строятся из компрессоров (западный термин для компрессора — CSA (carry save adder)).[4] Двоичный (k,l)-компрессор ширины 1 — это формула (или схема), реализующая булев оператор $(x_1, \ldots, x_k) \to (y_1, \ldots, y_l)$, определяемый условием $\sum 2^{a_i} x_i = \sum 2^{b_j} y_j$, где $k > l$, $a_i, b_j \in \mathbb{Z}$. (k,l)-компрессор произвольной ширины строится из параллельных копий компрессоров ширины 1 и позволяет сводить сложение k чисел к сложению l чисел.[5] Действительно, k чисел $x^i = [x^i_{m-1}, \ldots, x^i_0]$, $1 \le i \le k$, параллельными преобразованиями $(x^1_{j+a_1}, \ldots, x^k_{j+a_k}) \to (y^1_{j+b_1}, \ldots, y^l_{j+b_l})$, $j \in \mathbb{Z}$, переводятся в l чисел $y^i = [y^i_{m+h}, \ldots, y^i_0]$, $1 \le i \le l$, где $h < \log_2 k$, с сохранением суммы (все не определенные разряды в приведенных формулах полагаются равными нулю).

Вывод верхних оценок формульной или схемной сложности или глубины оператора C_n обычно состоит из двух этапов:

1) конструирование подходящего (k,l)-компрессора (при малых k и l),

2) синтез формулы (схемы) из компрессоров и вспомогательных формул (схем).

---

[4] Впрочем, для построения коротких формул в базисе B_0 пригодна простая конструкция, описанная в ([1], § 22; [4], § V.2.8). С ее помощью несложно вывести оценки $L_{B_0}(C_{2^m,i}) \le \left(C^i_{m+i-1} - C^{i-2}_{m+i-1}\right) \cdot 8^m$ и, как следствие, $L_{B_0}(C_n) \preceq n^5$, $L_{B_0}(S_n) \preceq n^{5.17}$.

[5] Поэтому дальнейшее рассмотрение можно ограничить компрессорами ширины 1.

Второй этап в случае оценки схемной сложности не представляет затруднений, в случае оценки сложности и глубины формул он в значительной степени унифицирован в работах [12], [15], [16]. Поэтому в любом случае задача фактически сводится к построению элементарных компрессоров.

Простейшим компрессором является $(3,2)$-компрессор (для него принято обозначение FA_3 (full adder)). Он реализует сумму трех битов по правилу $x_1 + x_2 + x_3 = 2y_2 + y_1$. Самое раннее установленное упоминание об этом компрессоре (и о компрессорах вообще) в контексте быстрого вычисления сумм приводится в [17]. Впоследствии этот компрессор и принцип его применения переоткрывался в работах многих авторов (перечень этих работ приводится, например, в [15]).

Рекордные на сегодняшний день оценки сложности и глубины (как для формул, так и для схем) получены при помощи сложно устроенных компрессоров. В частности, используется идея использования дополнительных способов кодировки двоичных наборов [12], [18] в конструкциях компрессоров. Примеры эффективных компрессоров с несколькими типами кодировки входов и выходов построены в [18] (сложность схем, вычисляющих сумму по модулю 4, над базисом B_2), [12] (глубина формул над базисом B_2), [19] (сложность схем над базисом B_2), [11] (сложность формул над базисами B_0 и B_2).

Результаты данной работы основаны на простом применении модулярной арифметики.[6] Вместо прямого вычисления арифметической суммы σ булевых переменных $x_1, \ldots, x_n$ вычисляем числа $(\sigma \bmod 2^k)$ и $(\sigma \bmod 3^l)$, где $2^k \cdot 3^l > n$, причем для вычисления $(\sigma \bmod 3^l)$ используется троичная система счисления. Искомое число σ может быть затем определено при помощи китайской теоремы об остатках.

При помощи этого приема получены оценки $L_{B_0}(C_n) \preceq n^{4.47}$, $L_{B_0}(S_n) \preceq n^{4.48}$, $L_{B_2}(C_n) \preceq n^{3.03}$, $L_{B_2}(S_n) \preceq n^{3.04}$, которые доказываются далее в теореме 2 (метод позволил понизить оценки сложности класса симметрических функций существенно). Предварительно, в разделе 2 приводятся необходимые технические утверждения, а в разделах 3, 4 описаны подходящие конструкции компрессоров.

Отметим, что метод может также применяться для вывода оценок глубины формул.

2. Описание метода

Обозначим через $C_n^{(3)}(x_1, \ldots, x_n) = \big(C_{n,m-1}^{(3)}, \ldots, C_{n,0}^{(3)}\big)$, $m = \lceil \log_3(n+1) \rceil$, булев $(n, 2m)$-оператор вычисления арифметической суммы булевых переменных $x_1, \ldots, x_n$ в троичной системе счисления. Компонента $C_{n,i}^{(3)}$ является 2-битным кодом соответствующей цифры из троичной записи числа.

Утверждение 4. *Пусть $2^k \cdot 3^l > n$. Для любого полного конечного базиса B справедливо*

$$L_B(C_n) \preceq 2^{(\log\log n)^{O(1)}} \left(\sum_{i=0}^{k-1} L_B(C_{n,i}) + \sum_{i=0}^{l-1} L_B\big(C_{n,i}^{(3)}\big) \right).$$

Доказательство. Для вычисления C_n перепишем число $\big[C_{n,l-1}^{(3)}, \ldots, C_{n,0}^{(3)}\big]$ в двоичной системе счисления и выполним восстановление результата из остатков по модулям 2^k и 3^l. Перезапись $O(\log n)$-разрядного числа из одной системы счисления в другую может быть реализована методом "деления пополам" А. Шёнхаге ([21], гл. 14). Следующий шаг реализуется

[6] Отметим, что модулярная арифметика уже используется при реализации симметрических функций контактными схемами (см., например, [20]).

при помощи сложения, вычитания, умножения и деления с остатком $O(\log n)$-разрядных чисел согласно тождеству

$$(\sigma \bmod 2^k \cdot 3^l) = (\sigma \bmod 2^k) + 2^k\big(\tau\big((\sigma \bmod 3^l) - (\sigma \bmod 2^k)\big) \bmod 3^l\big),$$

где константа τ определяется сравнением $\tau 2^k \equiv 1 \mod 3^l$. Все перечисленные арифметические операции реализуются с глубиной $(\log\log n)^{O(1)}$,[7] поэтому увеличивают сложность формул, реализующих входы, в $2^{(\log\log n)^{O(1)}}$ раз. □

Введем вспомогательные целочисленные операторы. Обозначим через $\mathrm{sort}_n(x_1,\ldots,x_n) = (\mathrm{sort}_{n,n-1},\ldots,\mathrm{sort}_{n,0})$ оператор упорядочивания набора из n чисел в порядке убывания: $\mathrm{sort}_{n,n-1} \geq \ldots \geq \mathrm{sort}_{n,0}$. Через $\mathrm{ext}_n^{\beta,\gamma}(x_1,\ldots,x_n) = (\mathrm{ext}_{n,m-1}^{\beta,\gamma},\ldots,\mathrm{ext}_{n,0}^{\beta,\gamma})$ обозначим (n,m)-оператор сглаживания-растяжения с параметрами $\beta,\gamma \in \mathbb{R}$, где $m = \lceil \gamma n \rceil$. Его компоненты определяются следующим образом: $\mathrm{ext}_{n,j} = \max\{x_i \mid |i-(j+1)/\gamma| \leq \beta\}$. Например,

$$\mathrm{ext}_5^{1,\sqrt{2}}(1,3,4,0,5) = (5,5,5,4,4,4,3,1).$$

Утверждение 5. *Пусть* $2^k \cdot 3^l > n$, $\beta \in \mathbb{N}$, $\beta = \Theta(\log n/\log\log n)$, $m = k + \lceil \log_2 3 \cdot l \rceil$. *Пусть* B *— полный конечный базис. Положим*

$$(L_{m-1},\ldots,L_0) = \mathrm{sort}_m\left(L_B(C_{n,k-1}),\ldots,L_B(C_{n,0}),\mathrm{ext}_l^{\beta,\log_2 3}\left(L_B\big(C_{n,l-1}^{(3)}\big),\ldots,L_B\big(C_{n,0}^{(3)}\big)\right)\right).$$

Тогда

$$L_B(S_n) \preceq 2^{(\log\log n)^{O(1)}}\left(\sum_{i=0}^{m-1} 2^{m-i}L_i\right).$$

Доказательство. Формула строится так же, как в методе утверждения 1 с той разницей, что симметрическая функция рассматривается как функция разрядов чисел $(\sigma \bmod 2^k)$ и $(\sigma \bmod 3^l)$, где σ — арифметическая сумма аргументов, а второе из чисел задано в системе счисления с основанием 3^β (цифры в этой системе записываются своим двоичным представлением).

Предварительно выполняется преобразование числа $(\sigma \bmod 3^l)$ из троичной системы счисления в систему с основанием 3^β, сложность этого преобразования учитывается так же, как в предыдущем утверждении. В результате число $(\sigma \bmod 3^l)$ записывается кодом длины $\lceil \log_2 3 \cdot \beta \rceil \cdot \lceil l/\beta \rceil = \log_2 3 \cdot l + O(\log\log n)$. По построению, каждый бит этого кода зависит от β соседних цифр исходной троичной записи числа $(\sigma \bmod 3^l)$. Поэтому в силу определения оператора ext вектор $\mathrm{ext}_l^{\beta,\log_2 3}\left(L_B(C_{n,l-1}^{(3)}),\ldots,L_B(C_{n,0}^{(3)})\right)$ из условия утверждения с точностью до множителя вида $2^{(\log\log n)^{O(1)}}$ мажорирует сложность вычисления основной части, а именно, $\lceil \log_2 3 \cdot l \rceil$ битов, этого кода. Оставшиеся $O(\log\log n)$ битов учитываются множителем вида $2^{O(\log\log n)}$ в оценке сложности.[8] □

Утверждение 5 подходит для компьютерных расчетов, однако не очень удобно для аналитических. Приведем ослабленный вариант этого утверждения, которым и будем пользоваться далее.

[7] Глубина преобразования N-разрядных чисел из одной системы счисления в другую по порядку не более чем в $\log N$ раз превосходит глубину умножения. Деление с остатком сводится к двум умножениям и нескольким сложениям (см., например, [21], гл. 16), поэтому имеет глубину по порядку не большую, чем глубина умножения и сложения. Сложение и вычитание имеют глубину $O(\log N)$ (см., например, [3], § 10.1, [5], § 2.5). Умножение N-разрядных чисел выполняется с глубиной $O(\log N)$, например, методом компрессоров (см. также [3], § 10.3).

[8] Без ограничения общности можем считать, что эти "лишние" биты соответствуют самым внешним переменным из разложения симметрической функции.

Утверждение 6. *Пусть* $l = \lceil \alpha \log_3 n \rceil$, $k = \lceil (1-\alpha) \log_2 n \rceil$. *Пусть выполнены соотношения*

$$L_B(C_{n,i}) \preceq 2^{\tau_1 \cdot i} n^{\omega_1}, \qquad L_B\big(C^{(3)}_{n,i}\big) \preceq 2^{\tau_2 \cdot i} n^{\omega_2},$$

а также $\tau_1(\eta - \alpha) > \eta$ *и* $\tau_2 \alpha \log_3 2 > \eta$ *при некотором* η, *где* $\alpha \le \eta \le 1$. *Тогда*

$$L_B(S_n) \preceq n^{\max\{\tau_1(1-\alpha)+\omega_1,\, \tau_2 \alpha \log_3 2 + \omega_2\} + o(1)}.$$

Доказательство. Представляем симметрическую функцию так же, как в утверждении 5. Далее используем формулу разложения симметрической функции по переменным, в качестве которых подставляем последовательно чередуемые блоки из компонент оператора C_n и блоки из битов двоичного кода числа $(\sigma \bmod 3^l)$ в порядке убывания номера разряда, где длины блоков соотносятся как $(\eta - \alpha)/\alpha$, до исчерпания кода числа $(\sigma \bmod 3^l)$ (код имеет длину $(\alpha + o(1)) \log_2 n$). Оставшиеся $(1 - \eta - o(1)) \log_2 n$ младших разрядов оператора C_n подставляются вместо самых внутренних переменных в разложении.

Длина блока выбирается медленно растущей функцией от n. Неравенства $\tau_1(\eta - \alpha) \ge \eta$ и $\tau_2 \alpha \log_3 2 \ge \eta$ означают, что вклад компонент соответственно из двоичной и троичной частей кода в сложность формулы при движении от старших разрядов к младшим убывает: сложность компоненты кода убывает быстрее, чем растет число ее вхождений в формулу. Поэтому сложность построенной формулы с точностью до множителя величины $n^{o(1)}$ определяется сложностью вычисления старших компонент кода: $C_{n,(1-\alpha)\log_2 n}$ и $C^{(3)}_{n,\alpha \log_3 n}$. □

Эффективный метод, позволяющий вычислять различные разряды суммы с различной формульной сложностью, предложен в [16]. Рассмотрим некоторый компрессор над базисом B. Через $x_{s,i}$ и $y_{s,i}$ обозначим соответственно входы и выходы, относящиеся к s-му разряду, $s \ge 0$. Через $\Phi(x)$ обозначим размер формулы, реализующей бит x (позволим $\Phi(x)$ принимать произвольное положительное вещественное значение). При любом s положим

$$a_s(p) = \sum_i \Phi(x_{s,i})^p - \sum_i \Phi(y_{s,i})^p, \tag{1}$$

где суммы по пустому множеству индексов считаются равными нулю. Из [16] извлекается

Лемма 1. *Пусть при некоторых* $p \ge 1$, $\Phi(x_{s,i}) > 0$ *и* $\nu \ge 1$ *выполнены неравенства*

$$a_0(p) > 0, \quad \sum_s a_s(p) \nu^{-s} > 0. \tag{2}$$

Тогда $L_B(C_{n,l}) \preceq (\nu^l n)^{1/p+o(1)}$ (*или* $L_B\big(C^{(3)}_{n,l}\big) \preceq (\nu^l n)^{1/p+o(1)}$ *в троичном случае*).

Для полноты изложения доказательство леммы приведено далее в разделе 6.

Приведем наглядную иллюстрацию применения изложенного технического приема на примере базиса B_2. Для реализации младших компонент операторов C_n и $C^{(3)}_n$ используем лемму 1 и самые простые конструкции компрессоров: стандартный $(3,2)$-компрессор[9] и троичный $(4,2)$-компрессор, который будет описан ниже в разделе 3.

Слева на рис. 1 изображен график показателя $\omega(\mu) = \inf_{p,\nu}((\mu \log_2 \nu + 1)/p)$ в оценке сложности n^ω вычисления разряда $C_{n,\mu \log n}$ при помощи двоичного компрессора методом леммы 1; нижняя грань берется по тем значениям p и ν, для которых выполнены условия леммы. Пунктиром изображен график функции $\omega(\mu) + 1 - \mu$, верхняя точка графика определяет сложность реализации симметрических функций методом п. б) утверждения 1.[10]

[9] Оптимальные формулы для C_n из таких компрессоров имеют сложность $O(n^{3.21})$ [15].

[10] Значения функций на графиках несколько завышены вблизи нуля.

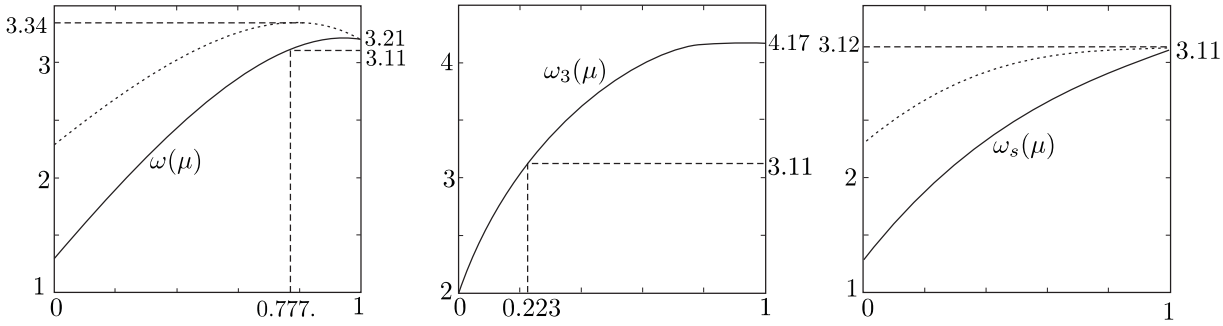

Рис. 1. Графики функций $\omega(\mu)$, $\omega_3(\mu)$, $\omega_s(\mu)$

В центре рис. 1 изображен график показателя $\omega_3(\mu) = \inf_{p,\nu}((\mu \log_3 \nu + 1)/p)$ в оценке сложности n^{ω_3} вычисления разряда $C^{(3)}_{n,\mu \log_3 n}$ при помощи троичного компрессора.

Справа изображен график показателя $\omega_s(\mu)$ в оценке сложности $L_{\mu \log_2 n} = n^{\omega_s}$ из утверждения 5 при выборе $k \approx 0.777 \log_2 n$ и $l \approx 0.223 \log_3 n$.[11] Сложность оператора C_n оценивается как $n^{\omega_s(1)+o(1)}$. Пунктиром показан график функции $\omega_s(\mu) + 1 - \mu$, максимум которой определяет сложность реализации симметрических функций методом утверждения 5.

Метод дает близкие оценки для сложности оператора C_n и класса симметрических функций: $L_{B_2}(C_n) \preceq n^{3,11}$ и $L_{B_2}(S_n) \preceq n^{3,12}$. Первая оценка уступает только оценке [11], вторая лучше любой из опубликованных. Для понижения оценок достаточно воспользоваться более эффективным двоичным компрессором.

Метод очевидно обобщается на случай комбинации более чем двух систем счисления, однако трудно ожидать существенного понижения оценок сложности за счет рассмотрения, скажем, 5-ричных или 7-ричных компрессоров. При этом, вероятно, можно добиться совпадения показателей степени n в оценках сложности для C_n и S_n.

Ввиду того, что сложность реализации компонент оператора C_n в предлагаемом методе примерно одинакова, метод не позволяет получать лучшие оценки сложности пороговых функций T^k_n, чем в утверждении 2.

3. Троичный компрессор

В этом разделе опишем реализацию троичного $(4, 2)$-компрессора (т. е. простейшего полного компрессора). Компрессор вычисляет (троичную) сумму $[U_1, U_0]$ четырех чисел $X_1, X_2, X_3, X_4 \in \{0, 1, 2\}$, т. е. $3U_1 + U_0 = X_1 + X_2 + X_3 + X_4$.

3.1. **Реализация в базисе B_0.** При вычислениях в стандартном базисе используем монотонную кодировку. Троичная цифра $X \in \{0, 1, 2\}$ кодируется упорядоченной парой двоичных битов $(x^\wedge, x^\vee)$, $x^\wedge \le x^\vee$, арифметическая сумма которых равна X (т. е. цифры 0, 1, 2 имеют соответственно коды 00, 01, 11).

Обозначим коды входов X_k через $(x^\wedge_k, x^\vee_k)$, а коды выходов U_j — через $(u^\wedge_j, u^\vee_j)$.

Предварительно заметим, что биты $u^\wedge_j$ и $u^\vee_j$ выражаются через входы двойственными формулами: формула, реализующая $u^\wedge_j$, превращается в формулу для $u^\vee_j$ при замене $x^\wedge_k \leftrightarrow x^\vee_k$, $\vee \leftrightarrow \wedge$. Это можно проверить, если

1) рассмотреть x°_k, u°_j, $\circ \in \{\vee, \wedge\}$ как функции трехзначной логики от переменных X_i;

[11] График ω_s получается объединением участков графиков $\omega([0; 0.777])$ и $\omega_3([0; 0.223])$ и последующей сортировкой.

2) в таблице значений функций x_k°, u_j° заменить 1 на 2, вместо функций базиса B_0 подставить в формулу аналогичные функции, определенные на наборах из нулей и двоек и принимающие значения из $\{0,2\}$;

3) применить троичный вариант принципа двойственности ([4], § I.1.3), считая константы k и $(2-k)$ двойственными, $k \in \{0,1,2\}$.

Таким образом, достаточно построить формулы для битов $u_0^\wedge$ и $u_1^\vee$. Построение выполним стандартным способом выражения через пороговые функции. Обозначим $\chi_1^t = (X_1+X_2 \ge t)$ и $\chi_2^t = (X_3+X_4 \ge t)$. Это пороговые функции троичных переменных, заданных двоичными кодами. Тогда

$$\begin{gathered} u_1^\vee = \chi_1^3 \vee \chi_1^2\chi_2^1 \vee \chi_1^1\chi_2^2 \vee \chi_2^3, \\ u_0^\wedge = (\chi_1^1\overline{\chi_1^2} \vee \chi_1^4)(\chi_2^1\overline{\chi_2^2} \vee \chi_2^4) \vee \chi_1^2\overline{\chi_1^3}(\chi_2^3\overline{\chi_2^4} \vee \overline{\chi_2^1}) \vee \chi_2^2\overline{\chi_2^3}(\chi_1^3\overline{\chi_1^4} \vee \overline{\chi_1^1}). \end{gathered} \tag{3}$$

Функции χ_i^1 и χ_i^2 реализуются формулами

$$\chi_i^1 = x_{2i-1}^\vee \vee x_{2i}^\vee, \quad \chi_i^2 = x_{2i-1}^\vee x_{2i}^\vee \vee x_{2i-1}^\wedge \vee x_{2i}^\wedge, \tag{4}$$

а функции χ_i^4 и χ_i^3 — двойственными к ним в указанном выше смысле.

Если обозначить через $\Phi(X_i)$ и $\Phi(U_j)$ сложность формул, реализующих любой из битов кода X_i и U_j соответственно, то из (3) и (4) получаем соотношения

$$\begin{aligned} &\Phi(U_0) \le 12(\Phi(X_1)+\Phi(X_2)+\Phi(X_3)+\Phi(X_4)), \\ &\Phi(U_1) \le 5(\Phi(X_1)+\Phi(X_2)+\Phi(X_3)+\Phi(X_4)). \end{aligned} \tag{5}$$

3.2. **Реализация в базисе B_2.** При вычислениях в базисе B_2 используем кодировку троичных цифр X тройками $(x^\oplus, x^\wedge, x^\vee)$, где $x^\wedge$, $x^\vee$ определяются как выше, а $x^\oplus$ — как $x^\wedge \oplus x^\vee$, т. е. $x^\oplus$ — младший разряд обычной двоичной записи числа X. Любые два из трех битов составляют неизбыточную кодировку.

Из соображений двойственности (см. выше) достаточно реализовать только биты $u_i^\wedge$ и $u_i^\oplus$. При построении двойственных формул над B_2 дополнительно применяются правила $x_k^\oplus \leftrightarrow \overline{x_k^\oplus}$ и $\oplus \leftrightarrow\sim$, где $\sim$ — операция эквивалентности.

Для упрощения восприятия и проверки формул определим троичные числа $[E_1,E_0] = X_1+X_2$ и $[H_1,H_0] = X_3+X_4$, и введем естественные обозначения e_i°, h_i° для битов кода. Тогда

$$\begin{aligned} &u_0^\wedge = (e_0^\wedge \oplus h_0^\vee)(e_0^\oplus \sim h_0^\oplus), && u_0^\oplus = (e_0^\wedge \sim h_0^\wedge)(e_0^\oplus \oplus h_0^\vee), \\ &u_1^\wedge = e_1^\vee h_1^\vee \vee e_0^\wedge x_3^\wedge x_4^\wedge \vee h_0^\wedge x_1^\wedge x_2^\wedge, && u_1^\oplus = u_0^\oplus \oplus x_1^\oplus \oplus x_2^\oplus \oplus x_3^\oplus \oplus x_4^\oplus. \end{aligned} \tag{6}$$

Вспомогательные функции реализуются формулами

$$\begin{aligned} &e_0^\wedge = (x_1^\wedge \oplus x_2^\vee)(x_1^\oplus \sim x_2^\oplus), && e_0^\oplus = (x_1^\wedge \sim x_2^\wedge)(x_1^\oplus \oplus x_2^\vee), \\ &e_0^\vee = (x_1^\wedge \oplus x_2^\oplus) \vee (x_1^\oplus \oplus x_2^\wedge), && e_1^\vee = x_1^\wedge x_2^\vee \oplus x_1^\oplus x_2^\wedge, \end{aligned} \tag{7}$$

аналогично реализуются h_i°. Отметим, что формулы для u_0°, e_0° получаются из стандартной реализации оператора сложения по модулю 3 (см., например, [5], § 4.4).

Обозначая через $\Phi(X_i)$ и $\Phi(U_j)$ сложность формул, реализующих любой из битов кода X_i и U_j соответственно, из (6) и (7) выводим соотношения

$$\begin{aligned} &\Phi(U_0) \le 4(\Phi(X_1)+\Phi(X_2)+\Phi(X_3)+\Phi(X_4)), \\ &\Phi(U_1) \le 5(\Phi(X_1)+\Phi(X_2)+\Phi(X_3)+\Phi(X_4)). \end{aligned} \tag{8}$$

4. Двоичные компрессоры

Для вывода основных результатов воспользуемся двоичными компрессорами из [11].

4.1. **Конструкция для базиса B_0.** В реализации компрессора над базисом B_0 некоторые пары битов v_1, v_2 кодируются тройками $\widehat{v} = (v^\wedge, v^\vee, v^\oplus)$, где $v^\wedge = v_1v_2$, $v^\vee = v_1 \vee v_2$, $v^\oplus = v^\vee\overline{v^\wedge}$, а некоторые тройки битов v_1, v_2, v_3 кодируются четверками $\widetilde{v} = (v', v'', v''', v^\oplus)$:

$$\begin{aligned} &v' = T_3^1(v_1, v_2, v_3) = v_1 \vee v_2 \vee v_3, \quad v'' = T_3^2(v_1, v_2, v_3) = v_1(v_2 \vee v_3) \vee v_2v_3, \\ &v''' = T_3^3(v_1, v_2, v_3) = v_1v_2v_3, \qquad v^\oplus = v_1 \oplus v_2 \oplus v_3 = v_1(v_2v_3 \vee \overline{v_2}\,\overline{v_3}) \vee \overline{v_1}(v_2\overline{v_3} \vee \overline{v_2}v_3). \end{aligned} \tag{9}$$

Пара битов $[v'', v^\oplus]$ является двоичной записью суммы $v_1 + v_2 + v_3$.

На рис. 2 изображен $(17, 6)$-компрессор, функционирующий по правилу

$$x_1 + \cdots + x_{17} = y_1 + y_2 + y_3 + 2y_4 + 4q_2^\oplus + 8q_2''.$$

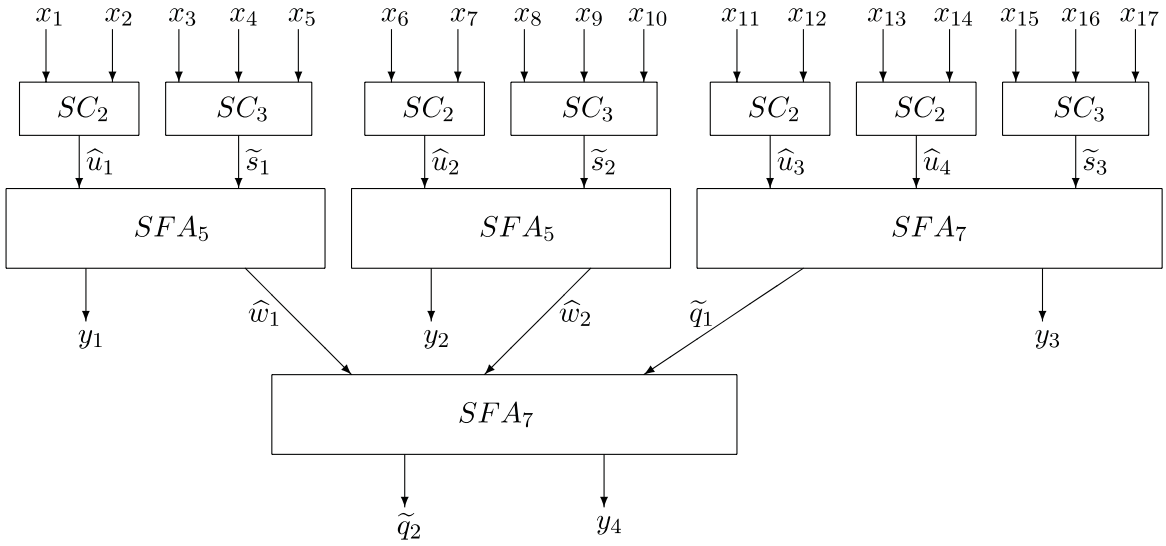

Рис. 2. Двоичный компрессор для базиса B_0

Блоки SC_2 и SC_3 выполняют перекодирование пар и троек входов соответственно в тройки типа $\widehat{v}$ и четверки типа $\widetilde{v}$ согласно приведенным выше формулам. Компрессор SFA_5 выполняет преобразование $(\widehat{u}, \widetilde{s}) \to (\widehat{w}, y)$ по правилу $u^\wedge + u^\vee + s' + s'' + s''' = y + 2(w^\wedge + w^\vee)$ согласно формулам

$$y = u^\oplus\overline{s^\oplus} \vee \overline{u^\oplus}s^\oplus, \quad w^\vee = u^\wedge \vee s'' \vee u^\vee s', \quad w^\wedge = u^\wedge s'' \vee u^\vee s'''. \tag{10}$$

Компрессор SFA_7 выполняет преобразование $(\widehat{u}, \widehat{w}, \widetilde{s}) \to (\widehat{q}, y)$ по правилу

$$u^\wedge + u^\vee + w^\wedge + w^\vee + s' + s'' + s''' = y + 2(q' + q'' + q''')$$

согласно формулам

$$\begin{aligned} &y = s^\oplus\big(u^\oplus w^\oplus \vee \overline{u^\oplus}\,\overline{w^\oplus}\big) \vee \overline{s^\oplus}\big(u^\oplus\overline{w^\oplus} \vee \overline{u^\oplus}w^\oplus\big), && q' = s'T_4^1(\widehat{u}, \widehat{w}) \vee T_4^2(\widehat{u}, \widehat{w}) \vee s'', \\ &q'' = s'''T_4^1(\widehat{u}, \widehat{w}) \vee s''T_4^2(\widehat{u}, \widehat{w}) \vee s'T_4^3(\widehat{u}, \widehat{w}) \vee T_4^4(\widehat{u}, \widehat{w}), && q''' = s'''T_4^3(\widehat{u}, \widehat{w}) \vee s''T_4^4(\widehat{u}, \widehat{w}), \end{aligned}$$

$$q^{\oplus} = \overline{s_2'}\,T_4^2(\widehat{u},\widehat{w})\overline{T_4^4(\widehat{u},\widehat{w})} \vee s_2'\overline{s_2''}\,T_4^1(\widehat{u},\widehat{w})\overline{T_4^3(\widehat{u},\widehat{w})} \vee \\ \vee\, s_2''\overline{s_2'''}\big(T_4^4(\widehat{u},\widehat{w}) \vee \overline{T_4^2(\widehat{u},\widehat{w})}\big) \vee s_2'''\big(T_4^3(\widehat{u},\widehat{w}) \vee \overline{T_4^1(\widehat{u},\widehat{w})}\big), \quad (11)$$

где

$$T_4^1(\widehat{u},\widehat{w}) = u^{\vee} \vee w^{\vee},\ \ T_4^2(\widehat{u},\widehat{w}) = u^{\vee}w^{\vee} \vee u^{\wedge} \vee w^{\wedge},\ \ T_4^3(\widehat{u},\widehat{w}) = u^{\wedge}w^{\vee} \vee u^{\vee}w^{\wedge},\ \ T_4^4(\widehat{u},\widehat{w}) = u^{\wedge}w^{\wedge}.$$

Обозначая через $\Phi(x)$ сложность формулы, реализующей бит x, и полагая для удобства

$$\Phi_1 = \Phi(x_1) = \Phi(x_2) = \Phi(x_3) = \Phi(x_6) = \Phi(x_7) = \Phi(x_8),$$
$$\Phi_2 = \Phi(x_4) = \Phi(x_5) = \Phi(x_9) = \Phi(x_{10}), \qquad \Phi_3 = \Phi(x_{11}) = \Phi(x_{12}) = \Phi(x_{13}) = \Phi(x_{14}),$$

$\Phi_4 = \Phi(x_{15})$, $\Phi_5 = \Phi(x_{16}) = \Phi(x_{17})$, из (9), (10), (11) получаем соотношение

$$\begin{pmatrix} \Phi(y_1)\\ \Phi(y_2)\\ \Phi(y_3)\\ \Phi(y_4)\\ \Phi(q_2^{\oplus})\\ \Phi(q_2'') \end{pmatrix} \le \begin{pmatrix} 12 & 16 & 0 & 0 & 0\\ 12 & 16 & 0 & 0 & 0\\ 0 & 0 & 32 & 4 & 16\\ 96 & 96 & 96 & 12 & 32\\ 144 & 144 & 96 & 14 & 40\\ 72 & 72 & 48 & 7 & 20 \end{pmatrix} \begin{pmatrix} \Phi_1\\ \Phi_2\\ \Phi_3\\ \Phi_4\\ \Phi_5 \end{pmatrix}. \quad (12)$$

4.2. **Конструкция для базиса B_2.** В конструкции компрессора над базисом B_2 некоторые пары битов v_1, v_2 кодируются как $\breve{v} = (v^0, v^{\oplus})$, где $v^0 = v_1$, $v^{\oplus} = v_1 \oplus v_2$.[12] Используем $(15,6)$-компрессор, изображенный на рис. 3, который получается обобщением конструкции из [12]. Он функционирует по правилу

$$16x_1 + 8(x_2 + x_3 + x_4) + 4(x_5 + x_6 + x_7) + 2(x_8 + x_9 + x_{10}) + x_{11} + \cdots + x_{15} = \sum_{i=1}^{6} 2^{i-1} y_i.$$

Блок Z выполняет кодирование $(v_1, v_2) \to \breve{v}$. Блок MDFA является $(5,3)$-компрессором из работы [19], выполняющим преобразование $(x, \breve{u}_1, \breve{u}_2) \to (y, \breve{z})$ согласно условию

$$x + u_1^0 + (u_1^0 \oplus u_1^{\oplus}) + u_2^0 + (u_2^0 \oplus u_2^{\oplus}) = 2(z^0 + (z^0 \oplus z^{\oplus})) + y$$

по формулам

$$y = x \oplus u_1^{\oplus} \oplus u_2^{\oplus}, \quad z^0 = (x \oplus u_1^0)\, u_1^{\oplus} \oplus u_1^0, \quad z^{\oplus} = ((x \oplus u_1^0) \vee u_1^{\oplus}) \oplus (x \oplus u_1^{\oplus} \oplus u_2^0)\,\overline{u_2^{\oplus}}. \quad (13)$$

Компрессор FA_3' совпадает с FA_3 с точностью до кодировки пары входов. Он выполняет преобразование $(x_1, \breve{z}_9) \to (y_5, y_6)$ по формулам

$$y_5 = x_1 \oplus z_9^{\oplus}, \quad y_6 = (x_1 \oplus z_9^0)\, z_9^{\oplus} \oplus z_9^0. \quad (14)$$

Обозначая $\vec{\Phi} = (\Phi(x_1), \ldots, \Phi(x_{15}))^T$, из (13) и (14) получаем соотношение

$$\begin{pmatrix} \Phi(y_1)\\ \Phi(y_2)\\ \Phi(y_3)\\ \Phi(y_4)\\ \Phi(y_5)\\ \Phi(y_6) \end{pmatrix} \le \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 3 & 1 & 2\\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 6 & 1 & 2\\ 0 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 6 & 3 & 3 & 6 & 1 & 2\\ 1 & 2 & 2 & 3 & 3 & 3 & 6 & 3 & 3 & 6 & 3 & 3 & 6 & 1 & 2\\ 1 & 4 & 4 & 9 & 3 & 3 & 6 & 3 & 3 & 6 & 3 & 3 & 6 & 1 & 2 \end{pmatrix} \cdot \vec{\Phi}. \quad (15)$$

[12] Такой способ кодирования применялся в [18].

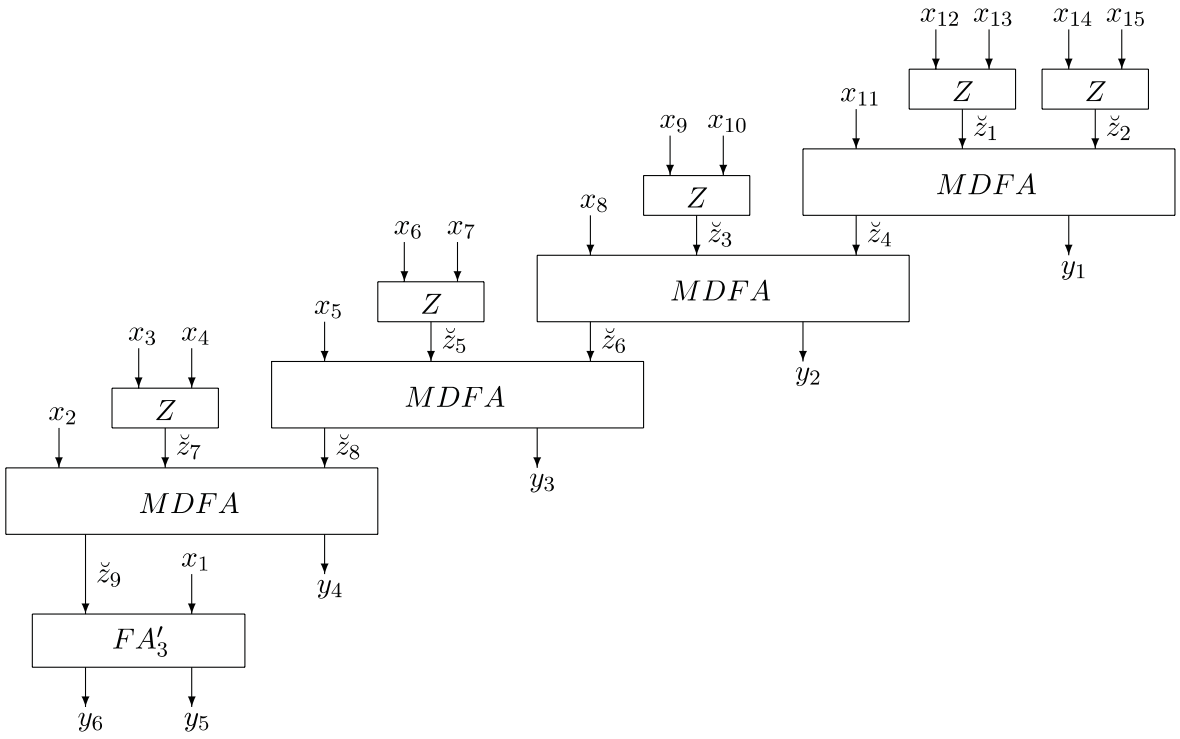

Рис. 3. Двоичный компрессор для базиса B_2

5. Результаты

Теорема 2. *Справедливы соотношения*

$$L_{B_0}(C_n) \preceq n^{4.47}, \quad L_{B_0}(S_n) \preceq n^{4.48}, \quad L_{B_2}(C_n) \preceq n^{3.03}, \quad L_{B_2}(S_n) \preceq n^{3.04}.$$

Доказательство. Докажем оценки для базиса B_0. Из (5) и леммы 1 при выборе $\Phi(x_1) = \cdots = \Phi(x_4)$, $p = 0.2812$ и $\nu = 2.255$ следует $L_{B_0}(C^{(3)}_{n,l}) \preceq 2^{4.172 \cdot l} n^{3.5562}$.

Из (12) и леммы 1 при выборе $\Phi_1 = 0.45$, $\Phi_2 = 0.21$, $\Phi_3 = 0.4$, $\Phi_4 = 1$, $\Phi_5 = 0.5$, $p = 0.271$ и $\nu = 1.2511$ следует $L_{B_0}(C_{n,l}) \preceq 2^{1.193 \cdot l} n^{3.6901}$.

Из двух приведенных оценок и утверждения 4 при выборе $k = \lceil 0.6531 \log_2 n \rceil$ и $l = \lceil 0.3469 \log_3 n \rceil$ вытекает $L_{B_0}(C_n) \preceq n^{4.47}$.

Для реализации симметрических функций используем соотношения

$$L_{B_0}\big(C^{(3)}_{n,l}\big) \preceq 2^{4.776 \cdot l} n^{3.4341}, \quad L_{B_0}(C_{n,l}) \preceq 2^{1.5703 \cdot l} n^{3.4543}.$$

Первое вытекает из (5) и леммы 1 при подстановке $\Phi(x_1) = \cdots = \Phi(x_4)$, $p = 0.2912$ и $\nu = 2.622$. Второе вытекает из (12) и леммы 1 при подстановке $\Phi_1 = 0.45$, $\Phi_2 = 0.2$, $\Phi_3 = 0.4$, $\Phi_4 = 1$, $\Phi_5 = 0.5$, $p = 0.2895$ и $\nu = 1.3704$. Далее применяется утверждение 6 с параметрами $\alpha = 0.3469$ и $\eta = 1$.

Перейдем к базису B_2. Из (8) и леммы 1 при выборе $\Phi(x_1) = \cdots = \Phi(x_4)$, $p = 0.4325$ и $\nu = 5.3513$ следует $L_{B_2}\big(C^{(3)}_{n,l}\big) \preceq 2^{5.5951 \cdot l} n^{2.3122}$.

Из (15) и леммы 1 при выборе $\Phi(x_1) = 10$, $\Phi(x_2) = \Phi(x_3) = 1.27$, $\Phi(x_4) = 0.55$, $\Phi(x_5) = \Phi(x_6) = 0.53$, $\Phi(x_7) = 0.26$, $\Phi(x_8) = \Phi(x_9) = 0.19$, $\Phi(x_{10}) = 0.09$, $\Phi(x_{11}) = \Phi(x_{12}) = 0.066$,

$\Phi(x_{13}) = 0.033$, $\Phi(x_{14}) = 0.15$, $\Phi(x_{15}) = 0.08$, $p = 0.444$ и $\nu = 1.3479$ следует $L_{B_2}(C_{n,l}) \preceq 2^{0.9701\cdot l} n^{2.2523}$.

Из указанных двух соотношений и утверждения 4 при выборе $k = \lceil 0.7978 \log_2 n \rceil$ и $l = \lceil 0.2022 \log_3 n \rceil$ вытекает $L_{B_2}(C_n) \preceq n^{3.03}$.

Для реализации симметрических функций воспользуемся соотношениями

$$L_{B_2}\big(C^{(3)}_{n,l}\big) \preceq 2^{6.3776\cdot l} n^{2.22718}, \quad L_{B_2}(C_{n,l}) \preceq 2^{1.33293\cdot l} n^{1{,}9763}.$$

Первое вытекает из (8) и леммы 1 при подстановке $\Phi(x_1) = \cdots = \Phi(x_4)$, $p = 0.449$ и $\nu = 7.278$. Второе вытекает из (15) и леммы 1 при подстановке $\Phi(x_1) = 10$, $\Phi(x_2) = \Phi(x_3) = 1.27$, $\Phi(x_4) = \Phi(x_5) = \Phi(x_6) = 0.53$, $\Phi(x_7) = 0.26$, $\Phi(x_8) = \Phi(x_9) = 0.19$, $\Phi(x_{10}) = 0.09$, $\Phi(x_{11}) = \Phi(x_{12}) = 0.068$, $\Phi(x_{13}) = 0.033$, $\Phi(x_{14}) = 0.16$, $\Phi(x_{15}) = 0.08$, $p = 0.506$ и $\nu = 1.596$. Применяем утверждение 6 с параметрами $\alpha = 0.202$ и $\eta = 0.8128$. □

Из теоремы 2 и утверждения 3 извлекаем

Следствие 2. Справедливы соотношения $L_{B_0}(M_n) \preceq n^{5.47}$, $L_{B_2}(M_n) \preceq n^{4.03}$.

6. Приложение. Доказательство леммы 1

По построению функция размера $\Phi(y_{s,j})$ формулы выхода компрессора является линейной комбинацией размеров формул входов с неотрицательными целочисленными коэффициентами. Поэтому неравенство (2) остается в силе при пропорциональном изменении всех $\Phi(x_{s,i})$. Без ограничения общности можем считать, что $\min\{\Phi(x_{s,i})\} = 1$.

В силу непрерывной зависимости $\Phi(y_{s,j})$ от $\Phi(x_{s,i})$ найдется такое $\delta > 0$, что неравенства (2) остаются справедливыми при подстановке в (1) параметров $\Phi'_{s,i} \in [\Phi(x_{s,i}) - \delta, \Phi(x_{s,i})]$ и $\Psi'_{s,i} \in [\Phi(y_{s,i}), \Phi(y_{s,i}) + \delta]$ вместо соответствующих $\Phi(x_{s,i})$ и $\Phi(y_{s,i})$. Тогда найдется (достаточно малое) $\lambda > 1$, для которого существуют $d^x_{s,i}, d^y_{s,i} \in \mathbb{Z}$ такие, что $\lambda^{d^x_{s,i}/p} \in [\Phi(x_{s,i}) - \delta, \Phi(x_{s,i})]$ и $\lambda^{d^y_{s,i}/p} \in [\Phi(y_{s,i}), \Phi(y_{s,i}) + \delta]$ для всех s, i. Назовем число $d^x_{s,i}$ (соответственно $d^y_{s,i}$) уровнем входа $x_{s,i}$ (выхода $y_{s,i}$). Можно считать, что $\min\{d^x_{s,i}\} = 0$. Обозначим $d = \max\{d^y_{s,i}\}$.

Формулу, реализующую оператор C_n (аналогично $C^{(3)}_n$), построим по следующему шаблону. Формула состоит из компрессоров, расположенных на различных уровнях и относящихся к различным разрядам: входами компрессоров могут быть входы формулы, выходы компрессоров, расположенных на уровнях с меньшими номерами, а также тождественно нулевые формулы.

Пусть компрессор разряда l и уровня k принимает входы разряда $s+l$ на уровнях $d^x_{s,i}+k$ и производит выходы разряда $s+l$ на уровнях $d^y_{s,i}+k$. Рассмотрим формулу, в которой при любом l, $0 \le l \le \log_2 n + 1$, на уровне k, $0 \le k \le \log_\lambda(\nu^l n)$, расположено $\lceil c\nu^l n \lambda^{-k} \rceil$ компрессоров разряда l, где c — некоторая константа, которая будет определена позднее. Ненулевые входы формулы располагаются в нулевом разряде на уровнях с номерами не меньше d.

Оценим число входов и выходов формулы, относящихся к фиксированному разряду l, в зависимости от уровня k. По построению все выходы формулы на уровнях меньше d являются нулевыми. Суммарное число входов (все они нулевые) на тех же уровнях есть $O(n)$. Если $d \le k \le \log_\lambda(\nu^l n)$, то разность между числом входов и числом выходов есть

$$\sum_{s,i} \lceil c\nu^{l-s} n \lambda^{d^x_{s,i}-k} \rceil - \sum_{s,i} \lceil c\nu^{l-s} n \lambda^{d^y_{s,i}-k} \rceil = c\nu^l n \lambda^{-k} \sum_s a_s(p) \nu^{-s} \pm O(1) = \Theta(\nu^l n \lambda^{-k}) \pm O(1),$$

следовательно, формула имеет не более $O(1)$ выходов на уровне d. На уровнях выше $\log_\lambda(\nu^l n)$ формула суммарно принимает и производит $O(1)$ входов и выходов.

Таким образом, формула в каждом разряде производит $O(\log n)$ выходов. Подходящий выбор константы c обеспечивает не менее n входов в нулевом разряде.

Согласно выбору λ, размер формул на уровне k оценивается сверху как $\lambda^{k/p}$. Поэтому формулы, реализующие выходы, относящиеся к разряду l, имеют сложность не выше $\lambda^{(\log_\lambda(\nu^l n)+O(1))/p} = O((\nu^l n)^{1/p})$.

Заключительное сложение $O(\log n)$ чисел (составленных из выходов формулы в разрядах не старше l-го) можно реализовать произвольной формулой полиномиальной сложности, поэтому окончательно сложность вычисления l-го разряда оператора C_n оценивается как $O((\nu^l n)^{1/p} \log^{O(1)} n)$. Лемма 1 доказана.

Литература

[1] Лупанов О.Б. *Асимптотические оценки сложности управляющих систем* (Изд-во МГУ, М., 1984).

[2] Нигматуллин Р.Г. *Сложность булевых функций* (Наука, М., 1991).

[3] Чашкин А.В. *Дискретная математика* (Академия, М., 2012).

[4] Яблонский С.В. *Введение в дискретную математику* (Наука, М., 1986).

[5] Dunne P.E. *The complexity of Boolean networks* (Academic Press, San Diego, 1988).

[6] Jukna S. *Boolean function complexity* (Springer-Verlag, Berlin–Heidelberg, 2012).

[7] Храпченко В.М. *О сложности реализации симметрических функций алгебры логики формулами в конечных базисах*, в сб. "Проблемы кибернетики" (Наука, М., 1976), вып. 31, с. 231–234.

[8] Черухин Д.Ю. *Нижние оценки формульной сложности симметрических булевых функций*, Дискр. анализ и исслед. опер. Сер. 1 **7** (3), 86–98 (2000).

[9] Храпченко В.М. *Об одном методе получения нижних оценок сложности π-схем*, Матем. заметки **10** (1), 83–92 (1971).

[10] Fischer M.J., Meyer A.R., Paterson M.S. $\Omega(n \log n)$ *lower bounds on length of Boolean formulas*, SIAM J. Comput. **11** (3), 416–427 (1982).

[11] Sergeev I.S. *Upper bounds for the formula size of the majority function*, http://arxiv.org/abs/1208.3874 (2012). [там же русский перевод]

[12] Paterson M., Zwick U. *Shallow circuits and concise formulae for multiple addition and multiplication*, Comput. Complexity **3** (3), 262–291 (1993).

[13] Peterson G.L. *An upper bound on the size of formulae for symmetric Boolean function*, Tech. Report 78-03-01 (Univ. Washington, 1978).

[14] Храпченко В.М. *О сложности реализации симметрических функций формулами*, Матем. заметки **11** (1), 109–120 (1972).

[15] Paterson M.S., Pippenger N., Zwick U. *Optimal carry save networks*, LMS Lecture Notes Series. Boolean function complexity (Cambridge University Press, 1992), vol. 169, p. 174–201.

[16] Paterson M.S., Pippenger N., Zwick U. *Faster circuits and shorter formulae for multiple addition, multiplication and symmetric Boolean functions*, Proc. 31st IEEE Symp. Found. Comput. Sci., 642–650 (1990).

[17] Столяров Г.К. *Способ параллельного умножения в цифровых вычислительных машинах и устройство для осуществления способа*, Авт. свид-во кл. 42 т 14, № 126668 (1960).

[18] Stockmeyer L.J. *On the combinational complexity of certain symmetric Boolean functions*, Math. Syst. Theory **10** (4), 323–336 (1977).

[19] Demenkov E., Kojevnikov A., Kulikov A., Yaroslavtsev G. *New upper bounds on the Boolean circuit complexity of symmetric functions*, Inf. Proc. Letters **110** (7), 264–267 (2010).

[20] Лупанов О.Б. *К вопросу о реализации симметрических функций алгебры логики контактными схемами*, в сб. "Проблемы кибернетики" (Наука, М., 1965), вып. 15, с. 85–99.

[21] Гашков С.Б. *Занимательная компьютерная арифметика. Быстрые алгоритмы операций с числами и многочленами* (Либроком, М., 2012).

И.С. Сергеев

старший научный сотрудник, кафедра дискретной математики,
Московский государственный университет,
ГСП-1, Ленинские горы, г. Москва, 119991, Россия,

e-mail: isserg@gmail.com

I.S. Sergeev

Upper bounds on the formula size of symmetric Boolean functions

Abstract. It is proved that complexity of implementation of the counting function of n Boolean variables with binary formulae is at most $n^{3.03}$ and is at most $n^{4.47}$ with respect to DeMorgan formulae. Hence, the same bounds hold for the formula size of any threshold symmetric function of n variables, particularly, for majority function. The following bounds are proved for the formula size of any symmetric Boolean function of n variables: $n^{3.04}$ with respect to binary formulae and $n^{4.48}$ with respect to DeMorgan formulae. A proof is based on modular arithmetic.

Keywords: formula size, symmetric Boolean functions, majority function, multiplication.

I.S. Sergeev

Senior Researcher, Chair of Discrete Mathematics,
Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991 Russia,

e-mail: isserg@gmail.com