ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь—декабрь 2004. Серия 1. Том 11, № 4, 68–80

УДК 519.7

О СРЕДНЕЙ МОНОТОННОЙ СЛОЖНОСТИ БУЛЕВЫХ ФУНКЦИЙ*)

А. В. Чашкин

Исследуется средняя сложность вычисления булевых функций неветвящимися монотонными программами с условной остановкой. Показано, что существуют такие n-местные булевы функции, что отношение средней монотонной сложности этих функций к обычной средней сложности равно $\Omega(\sqrt{2^n/n})$. Также показано, что средняя монотонная сложность линейной функции с точностью до постоянного множителя равна $n \log_2 n$. Приведен пример линейного оператора, средняя монотонная сложность которого с точностью до постоянного множителя равна n^2.

1. В настоящей статье рассматривается средняя сложность вычисления булевых функций монотонными неветвящимися программами с условной остановкой. Эти программы являются частным случаем неветвящихся программ с условной остановкой, рассмотренных в [1–3]. Программы в [1–3] вычисляют булевы функции, и их работу можно представить следующим образом. Вычисления выполняет процессор, снабженный памятью, состоящей из отдельных ячеек, которые будем обозначать символами x_i, y_j и z_k. Ячейки x_i содержат значения независимых переменных x_i. Ячейки y_j используются для хранения промежуточных результатов вычислений, эти ячейки будем называть *внутренними* переменными. Ячейки z_k используются для записи результатов работы программы, такие ячейки будем называть *выходными* переменными.

Процессор работает под управление программы, являющейся последовательностью вычислительных команд и команд остановки. Каждая вычислительная команда p имеет вид $a = f(b, c)$ и за единицу времени

*)Исследование выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 02-01-00985), программы поддержки ведущих научных школ РФ (проект НШ-1807.2003.1), программы «Университеты России» (проект УР.04.03.007) и программы фундаментальных исследований Отделения математических наук РАН «Алгебраические и комбинаторные методы математической кибернетики» (проект «Оптимальный синтез управляющих систем»).

© **2004 Чашкин А. В.**

вычисляет значение булевой функции f, аргументы которой b и c являются переменными x_i, y_j или z_k. Аргументы функции f называются *входами* команды p. Вычисленное значение присваивается внутренней или выходной переменной a, которая называется *выходом* команды p. Значение внутренней переменной y_j (или выходной переменной z_k) в момент времени t на наборе независимых переменных x будем обозначать через $y_j(x;t)$ (соответственно через $z_k(x;t)$). Команда остановки имеет вид $\texttt{Stop}(q)$, где аргумент q есть x_i, y_j или z_k, и может прекратить выполнение программы. Если значение аргумента равно единице, то процессор прекращает работу. Если значение аргумента равно нулю, то выполняется следующая команда программы.

Пусть $\alpha = (\alpha_1, \ldots, \alpha_n)$ — набор значений независимых переменных, $P = p_1 \ldots p_L$ — программа, $p_{i_1}, \ldots, p_{i_r}$ — команды остановки этой программы. Через $q_1(\alpha), \ldots, q_r(\alpha)$ обозначим аргументы команд остановки программы P. Будем говорить, что k-я команда остановки прекращает вычисления программы P на наборе α, если

$$q_1(\alpha) = \cdots = q_{k-1}(\alpha) = 0, \qquad q_k(\alpha) = 1.$$

Результат действия программы P на наборе α обозначим через $P(\alpha)$ и его l-ю компоненту определим следующим образом:

$$P_l(\alpha) = \begin{cases} z_l(\alpha; t_k), & \text{если } q_1(\alpha) = \cdots = q_{k-1}(\alpha) = 0, \quad q_k(\alpha) = 1, \\ z_l(\alpha; L), & \text{если } q_1(\alpha) = \cdots = q_r(\alpha) = 0, \end{cases}$$

т. е. $P_l(\alpha)$ равно значению выходной переменной в момент остановки программы. Легко видеть, что

$$\begin{aligned} P_l(\alpha) = {} & q_1(\alpha)\&z_l(\alpha; t_1) \vee \overline{q}_1(\alpha)\&q_2(\alpha)\&z_l(\alpha; t_2) \vee \ldots \\ & \ldots \vee \overline{q}_1(\alpha)\& \cdots \&\overline{q}_{k-1}(\alpha)\&q_k(\alpha)\&z_l(\alpha; t_k) \vee \ldots \\ & \ldots \vee \overline{q}_1(\alpha)\& \cdots \&\overline{q}_{r-1}(\alpha)\&q_r(\alpha)\&z_l(\alpha; t_r) \vee \overline{q}_1(\alpha)\& \cdots \\ & \ldots \&\overline{q}_r(\alpha)\&z_l(\alpha; L). \end{aligned} \tag{1}$$

Число команд, выполненных программой P на наборе переменных x, назовем *временем работы* программы P на x и обозначим через $T_P(x)$. Величину

$$T(P) = 2^{-n} \sum T_P(\sigma),$$

где суммирование производится по всем двоичным наборам длины n, назовем *средним временем работы* программы P. Если для некоторой

n-местной булевой вектор-функции f и любого двоичного набора σ длины n справедливо равенство $f(\sigma) = P(\sigma)$, то будем говорить, что программа P вычисляет функцию f. Величину

$$T(f) = \min T(P),$$

где минимум берется по всем программам, вычисляющим f, назовем *средней сложностью* функции f. Программу P, вычисляющую функцию f, для которой справедливо равенство $T(P) = T(f)$, назовем *минимальной* программой. *Сложностью* $C(P)$ программы P назовем число команд этой программы. Величину

$$C(f) = \min C(P),$$

где минимум берется по всем программам, вычисляющим f, назовем *сложностью* функции f. Величина $C(f)$ характеризует время, необходимое для вычисления f в худшем случае. Поэтому $C(f)$ также будем называть *сложностью* в худшем случае.

Рассматриваемые ниже монотонные программы отличаются от произвольных программ тем, что в вычислительных командах используются только монотонные булевы функции: дизъюнкция, конъюнкция, тождественная функция и две константы 0 и 1. Нетрудно видеть, что любая булева функция может быть вычислена подходящей монотонной неветвящейся программой с условной остановкой. В качестве примера рассмотрим следующую программу:

$$z = 0;\ y = x_1 \& x_2;\ \texttt{Stop}(y);\ z = 1;\ y = x_1 \vee x_2;\ \texttt{Stop}(y);\ z = 0,$$

которая, как легко видеть, вычисляет линейную функцию $x_1 \oplus x_2$. *Средней монотонной сложностью* и *монотонной сложностью* булевой функции f назовем величины

$$T_m(f) = \min T(P), \qquad C_m(f) = \min C(P),$$

где минимумы берутся по всем монотонным программам, вычисляющим f.

2. Покажем, что монотонная сложность произвольной n-местной булевой функции по порядку величины не более чем в n раз превосходит сложность ее вычисления схемами из функциональных элементов в полном базисе. Это легко следует из результатов Вегенера о сложности слойфункций [6]. Ниже через $L_{\{\&,\vee\}}(f)$ обозначается сложность реализации монотонной булевой функции f схемами в монотонном базисе $\{\&, \vee\}$, а

через $L_{\{\&,\vee,\neg\}}(f)$ — сложность реализации булевой функции f схемами в базисе $\{\&,\vee,\neg\}$.

Теорема 1. *Для каждой n-местной булевой функции f*

$$C_m(f) \leqslant 2nL_{\{\&,\vee,\neg\}}(f) + 7n^2.$$

Доказательство. Для построения программы, вычисляющей n-местную булеву функцию f, воспользуемся ее слой-функциями. Монотонная функция f_k называется k-й слой-функцией функции f, если

$$f_k(x_1,\ldots,x_n) = \begin{cases} 1 & \text{при } \sum\limits_{i=1}^{n} x_i > k, \\ f(x_1,\ldots,x_n) & \text{при } \sum\limits_{i=1}^{n} x_i = k, \\ 0 & \text{при } \sum\limits_{i=1}^{n} x_i < k. \end{cases}$$

Если вычислены значения всех слой-функций функции f и значения всех симметрических пороговых функций $Th_k(x_1,\ldots,x_n)$, определяемых для $k = 1, 2, \ldots, n$ равенством

$$Th_k(x_1,\ldots,x_n) = \begin{cases} 1 & \text{при } \sum\limits_{i=1}^{n} x_i \geqslant k, \\ 0 & \text{при } \sum\limits_{i=1}^{n} x_i < k, \end{cases}$$

то значение функции f находится следующим образом. Выходной переменной z присваивается значение $f_n(x_1,\ldots,x_n)$ и проверяется равенство единице функции $Th_n(x_1,\ldots,x_n)$. Если $Th_n(x_1,\ldots,x_n) = 1$, то вычисления прекращаются. Если $Th_n(x_1,\ldots,x_n) = 0$, то выходной переменной z присваивается значение $f_{n-1}(x_1,\ldots,x_n)$ и проверяется равенство единице функции $Th_{n-1}(x_1,\ldots,x_n)$. Повторяя, если необходимо, подобные действия в общей сложности n раз, найдем значение функции f. Нетрудно видеть, что для сложности такой монотонной программы P справедливо неравенство

$$C(P) \leqslant L_{\{\&,\vee\}}(f_1,\ldots,f_n) + L_{\{\&,\vee\}}(Th_1,\ldots,Th_n) + 2n - 1. \qquad (2)$$

Известно [6], что сложность одновременного вычисления всех слой-функций произвольной n-местной булевой функции f схемами из дизъюнкций и конъюнкций не превосходит величины $2nL_{\{\&,\vee,\neg\}}(f)+6n^2$, а сложность

одновременного вычисления всех n-местных симметрических пороговых не больше n^2. Поэтому

$$C_m(f) \leqslant 2nL_{\{\&,\vee,\neg\}}(f) + 7n^2. \tag{3}$$

Теорема 1 доказана.

Нетрудно видеть, что утверждение теоремы 1 справедливо также и для булевых операторов.

Следующую теорему приведем без доказательства. Ее нижняя оценка является тривиальным следствием результатов работы [1], а верхняя оценка легко следует из теоремы 1 и [4].

Теорема 2. *Пусть $n \to \infty$. Тогда для почти каждой n-местной булевой функции f^**

$$T_m(f) = \Omega\Big(\frac{2^n}{n}\Big)$$

и для каждой n-местной булевой функции f

$$T_m(f) = \mathcal{O}\Big(\frac{2^n}{n}\Big).$$

Далее рассмотрим вопрос о том, насколько сильно могут различаться средняя монотонная сложность и средняя сложность булевой функции. Сравнение результатов работы [1] и теоремы 2 показывает, что при $n \to \infty$ средняя монотонная сложность и средняя сложность почти каждой n-местной булевой функции отличаются не более чем в постоянное число раз. В приводимой ниже теореме 3 оценивается максимально возможное значение этого отношения. При доказательстве этой теоремы потребуется следующее простое, но эффективное вспомогательное утверждение.

Лемма 1. *Пусть P — монотонная программа, α и β — булевы наборы одинаковой длины. Тогда если $\alpha \preccurlyeq \beta$, то $T_P(\alpha) \geqslant T_P(\beta)$.*

Доказательство. Пусть $q_1, \ldots, q_r$ — аргументы всех команд остановки программы P. Из монотонности функций $q_i(x)$ легко следует, что если $q_i(\alpha) = 1$, то $q_i(\beta) = 1$. Поэтому время работы программы P на наборе α не меньше времени работы на наборе β. Лемма 1 доказана.

Теорема 3. *Существует такая n-местная булева функция f, что*

$$T_m(f)\Big/T(f) = \Omega\Bigg(\sqrt{\frac{2^n}{n}}\Bigg).$$

$^*f(n) = \Omega(g(n))$ означает, что существует такая константа c, что $f(n) \geqslant c \cdot g(n)$.

Доказательство. Пусть $h(x_1,\ldots,x_k)$ — функция, средняя монотонная сложность которой максимальна среди всех k-местных булевых функций. Из теоремы 2 следует, что $T_m(h) \geqslant \dfrac{c\cdot 2^k}{k}$, где c — некоторая постоянная. Рассмотрим функцию

$$f(x_1,\ldots,x_n) = h(x_1,\ldots,x_k)x_{k+1}\&\cdots\&x_n.$$

Из леммы 1 следует, что для любой монотонной программы P, вычисляющей f, и любых двоичных наборов $\alpha=(\alpha_1,\ldots,\alpha_k)$ и $\beta=(\beta_{k+1},\ldots,\beta_n)$ выполняется неравенство $T_P(\alpha,\beta)\geqslant T_P(\alpha,\mathbf{1})$, где $\mathbf{1}$ — двоичный набор длины $n-k$, все компоненты которого равны единице. Пусть P^* — минимальная монотонная программа, вычисляющая функцию h. Если в программе P вместо последних $n-k$ переменных подставить единицы, то получившаяся программа будет вычислять функцию h, причем $T_P(\alpha,\mathbf{1})\geqslant T_{P^*}(\alpha)$ для любого α из $\{0,1\}^k$. Поэтому

$$\begin{aligned} T_m(P) &= \frac{1}{2^n}\sum_{\alpha\in\{0,1\}^k,\beta\in\{0,1\}^{n-k}} T_P(\alpha,\beta) \geqslant \frac{1}{2^n}\sum_{\alpha\in\{0,1\}^k} T_P(\alpha,\mathbf{1})2^{n-k} = \\ &= \frac{1}{2^k}\sum_{\alpha\in\{0,1\}^k} T_P(\alpha,\mathbf{1}) \geqslant \frac{1}{2^k}\sum_{\alpha\in\{0,1\}^k} T_{P^*}(\alpha) = T_m(h) \geqslant \frac{c\cdot 2^k}{k}. \end{aligned} \tag{4}$$

С другой стороны, нетрудно видеть, что программа P, начинающаяся с $2n-2k+1$ команд

$$z=0;\ y=\overline{x}_{k+1};\ \texttt{Stop}(y),\ \ldots\ y=\overline{x}_n;\ \texttt{Stop}(y);$$

после которых стоят команды программы P^*, вычисляет функцию f. Учитывая верхнюю оценку средней сложности произвольной n-местной функции [1], видим, что для среднего времени работы программы P справедлива следующая оценка:

$$T(P) \sim \frac{1}{2^n}\Bigl(\sum_{j=1}^{n-k}(1+2j)\,2^{n-j} + \frac{2^{k-1}}{k}\cdot 2^k\Bigr) \leqslant 5 + \frac{2^{2k-n-1}}{k}. \tag{5}$$

Положим $k = \left\lfloor \dfrac{1}{2}\bigl(n+\log_2 n\bigr)\right\rfloor$. Тогда из неравенств (4) и (5) следует, что $T_m(f) = \Omega\left(\sqrt{\dfrac{2^n}{n}}\right)$ и $T(f)=\mathcal{O}(1)$. Следовательно,

$$T_m(f)\Big/T(f) \sim \Omega\left(\sqrt{\frac{2^n}{n}}\right).$$

Теорема 3 доказана.

В [1] показано, что $L(f)/T(f) = \mathcal{O}(\sqrt{2^n/n})$ для любой n-местной булевой функции f. Поэтому

$$\frac{T_m(f)}{T(f)} \leqslant \frac{C_m(f)}{T(f)} \leqslant \frac{2n(L_{\{\&,\vee,\neg\}}(f)+4n)}{T(f)} = \mathcal{O}\Big(\sqrt{n2^n}\Big).$$

3. Теперь рассмотрим среднюю монотонную сложность трех простейших булевых функций — дизъюнкции, конъюнкции и линейной функции от n переменных. В [3] было показано, что $T(x_1 \oplus \cdots \oplus x_n) = n-1$ и при $n \to \infty$ справедливы асимптотические равенства $T(x_1 \vee \cdots \vee x_n) \sim \frac{8}{3}$ и $T(x_1\& \ldots \& x_n) \sim \frac{11}{3}$. Средние монотонные сложности дизъюнкции и конъюнкции от n переменных вычисляются легко. Средняя монотонная сложность дизъюнкции равна ее средней сложности, а для конъюнкции имеет место равенство $T_m(x_1\& \ldots \& x_n) = n-1$. Оценки средней монотонной сложности линейной функции имеют следующий вид.

Теорема 4. *При* $n \to \infty$

$$T_m(x_1 \oplus \cdots \oplus x_n) \asymp n\log_2 n, \qquad C_m(x_1 \oplus \cdots \oplus x_n) \asymp n\log_2 n.$$

Перед доказательством теоремы докажем две леммы.

Набор $\alpha \in \{0,1\}^n$ называется *нижней единицей* n-местной монотонной булевой функции f, если $f(\alpha) = 1$ и $f(\beta) = 0$ для любого $\beta \prec \alpha$.

Лемма 2. *Пусть монотонные n-местные булевы функции $f_1, \ldots, f_m$ таковы, что для любого ненулевого набора α из $\{0,1\}^n$ среди этих функций найдется такая функция f_j, что α будет нижней единицей этой функции. Тогда* $L_{\{\&,\vee\}}(f_1, \ldots, f_m) \geqslant n\log_2 n - 2n$.

Доказательство. Легко видеть, что если ненулевой набор $(\alpha_1, \ldots, \alpha_{n-1}, 0)$ является нижней единицей n-местной булевой функции $f(x_1, \ldots, x_n)$, то набор $(\alpha_1, \ldots, \alpha_{n-1})$ будет нижней единицей $(n-1)$-местной булевой функции $f(x_1, \ldots, x_{n-1}, 0)$. Поэтому справедливо следующее свойство: *если n-местные булевы функции $f_i(x_1, \ldots, x_n)$ удовлетворяют условиям леммы, то и $(n-1)$-местные булевы функции $f_i(x_1, \ldots, x_{n-1}, 0)$ также удовлетворяют условиям леммы.* Очевидно, что аналогичное свойство функций f_i справедливо и в случае подстановки нуля вместо любой из n переменных, а не только вместо x_n. Воспользуемся этим свойством.

Пусть S_n — оптимальная схема в базисе $\{\&, \vee\}$, которая вычисляет систему монотонных n-местных функций $\{f_1, \ldots, f_m\}$, удовлетворяющих условиям леммы. Среди этих функций обязательно присутствует конъюнкция $x_1\&x_2\& \ldots \& x_n$, так как она является единственной монотонной

функцией, для которой единичный набор будет нижней единицей. Очевидно, что глубина конъюнкции $x_1\&x_2\&\ldots\&x_n$ не меньше $\lceil\log_2 n\rceil$. Следовательно, в схеме S_n найдется такой вход, например, соответствующий переменной x_i, что любая цепь, связывающая этот вход и выход, на котором вычисляется конъюнкция $x_1\&x_2\&\ldots\&x_n$, проходит не менее чем через $\lceil\log_2 n\rceil$ элементов. После подстановки нуля вместо переменной x_i эта конъюнкция превратится в тождественный нуль. Далее воспользуемся следующим простым свойством монотонных функций: *конъюнкция и дизъюнкция любых двух ненулевых монотонных булевых функций не равны тождественному нулю.* Из этого свойства следует, что если в монотонной схеме некоторый элемент вычисляет тождественный нуль, то, по крайней мере, один из его двух предков также вычисляет тождественный нуль. Поэтому между выходом схемы, на котором после подстановки $x_i \equiv 0$ вычисляется тождественный нуль, и i-м входом — единственным, равным тождественному нулю, можно найти соединяющую их цепь l, каждый элемент которой вычисляет тождественный нуль и, следовательно, может быть удален из схемы. Таким образом, если схема S_{n-1} вычисляет систему функций, получившуюся из системы функций $\{f_1,\ldots,f_m\}$ подстановкой нуля вместо переменной x_i, то она содержит по крайней мере на $\lceil\log_2 n\rceil$ элементов меньше, чем схема S_n. Поэтому $L_{\{\vee,\&\}}(S_n) \geqslant L_{\{\vee,\&\}}(S_{n-1}) + \lceil\log_2 n\rceil$. Повторяя подобные рассуждения в общей сложности $n-1$ раз, нетрудно убедиться в справедливости следующих неравенств:

$$\begin{aligned} L_{\{\vee,\&\}}(S_n) &\geqslant \lceil\log_2 n\rceil + L_{\{\vee,\&\}}(S_{n-1}) \\ &\geqslant \lceil\log_2 n\rceil + \lceil\log_2(n-1)\rceil + L_{\{\vee,\&\}}(S_{n-2}) \geqslant \ldots \\ &\geqslant \sum_{k=1}^{n}\lceil\log_2 k\rceil \geqslant \sum_{k=1}^{n}\log_2 k = \log_2 n! \geqslant \log_2\Big(\frac{n}{4}\Big)^n = n\log_2 n - 2n. \end{aligned}$$

Лемма 2 доказана.

Лемма 3. *Пусть* $\alpha \in \{0,1\}^n$. *Тогда*

$$C_m(x_1 \oplus \cdots \oplus x_n \oplus \alpha) \geqslant n\log_2 n - 2n.$$

Доказательство. Пусть монотонная программа P вычисляет функцию $x_1 \oplus \cdots \oplus x_n \oplus \alpha$ и $t_1,\ldots,t_r$ — номера команд остановки этой программы, причем $t_1 < \cdots < t_r$. Будем полагать, что $z(x;t_i) \neq z(x;t_{i+1})$ при $i = 1,2,\ldots,r-1$. Нетрудно видеть, что любую программу можно преобразовать в программу с данным свойством без увеличения ее

сложности. Введем функции $h_1, \ldots, h_{r+1}$, определив их равенствами

$$h_i(x) = \begin{cases} q_i(x)\&z(x;t_i), & \text{если } i = 1,2,\ldots,r, \\ z(x,L), & \text{если } i = r+1. \end{cases}$$

Легко видеть, что $h_i(x) \leqslant q_i(x)$ для каждого $i \in \{1,\ldots,r\}$.

Преобразуем программу P в новую программу P', поместив перед i-й командой остановки s_i $(i = 1,2,\ldots,r)$ команду, которая вычисляет функцию h_i и присваивает вычисленное значение переменной z. Так как $h_{r+1}(x) = z(x;L)$ и для каждого $k \in \{1,\ldots,r\}$

$$\overline{q}_1(x)\cdots\overline{q}_{k-1}(x)q_k(x)z(x;t_k) = \overline{q}_1(x)\cdots\overline{q}_{k-1}(x)q_k(x)h_k(x),$$

то из (1) следует, что программы P и P' вычисляют одну и ту же функцию.

Теперь покажем, что для любого ненулевого набора α из $\{0,1\}^n$ среди функций $q_1,\ldots,q_r$ и $h_1,\ldots,h_{r+1}$ найдется такая, что α будет нижней единицей этой функции.

Допустим, что это не так, и некоторый набор α не является нижней единицей ни одной из функций $q_1,\ldots,q_r$ и $h_1,\ldots,h_{r+1}$. Без ограничения общности будем полагать, что работу программы P' на наборе α прекращает ее k-я команда остановки. Тогда*)

$$q_1(\alpha) = \cdots = q_{k-1}(\alpha) = 0, \quad q_k(\alpha) = 1. \tag{6}$$

Рассмотрим два случая: $h_k(\alpha) = 0$ и $h_k(\alpha) = 1$.

Первый случай. Так как $q_k(\alpha) = 1$ и α не является нижней единицей функции q_k, то найдется хотя бы один такой набор β, меньший α и отличающийся от α ровно в одном разряде, что $q_k(\beta) = 1$. Из монотонности функций q_i и равенств (6) следует, что $q_1(\beta) = \cdots = q_{k-1}(\beta) = 0$, т. е. значение программы P' на наборе β равно $h_k(\beta)$. С другой стороны, из равенства $h_k(\alpha) = 0$ и монотонности функции h_k следует, что $h_k(\beta) = 0$, т. е. вычисляемая программой P' функция не меняет своего знака при изменении значения одного из аргументов. Следовательно, эта функция не является линейной.

Теперь рассмотрим случай $h_k(\alpha) = 1$. Так как α не является нижней единицей функции h_k, то найдется хотя бы один такой набор γ, меньший α, отличающийся от α ровно в одном разряде и $h_k(\gamma) = 1$. Из неравенства $h_k(x) \leqslant q_k(x)$ следует, что $q_k(\gamma) = 1$, а из монотонности функций

*) Случай, когда все $q_i(\alpha)$ равны нулю, сводится к рассматриваемому добавлением в конце программы P' двух команд: $y = 1$ и $\texttt{Stop}(y)$.

q_i и равенств (6) следует, что $q_1(\gamma) = \cdots = q_{k-1}(\gamma) = 0$, т. е. значение программы P' на наборе γ равно $h_k(\gamma)$. Следовательно, как и в первом случае вычисляемая программой P' функция не меняет своего знака при изменении значения одного из аргументов и поэтому не является линейной.

Таким образом, сделанное выше предположение о существовании ненулевого набора, который не является нижней единицей ни одной из функций $q_1, \ldots, q_r$ и $h_1, \ldots, h_{r+1}$ ложно. Следовательно, каждый ненулевой набор будет нижней единицей хотя бы одной из этих функций.

Так как программа P' содержит ровно r команд остановки, то из леммы 2 следует, что $C(P') - r$ — число вычислительных команд программы P' — не меньше $n \log_2 n - 2n$. С другой стороны, число вычислительных команд программы P' равно числу команд программы P, т. е. $C(P') - r = C(P)$. Следовательно,

$$C(P) = C(P') - r \geqslant n \log_2 n - 2n.$$

Лемма 3 доказана.

Доказательство теоремы 4. Сначала покажем, что средняя монотонная сложность линейной n-местной булевой функции асимптотически не меньше $\frac{1}{2} n \log_2 n$. Пусть монотонная программа P вычисляет функцию $x_1 \oplus \cdots \oplus x_n$. Положим $k = \lceil n/2 + \sqrt{n} \log_2 n \rceil$ и рассмотрим произвольный набор $\alpha = (\alpha_1, \ldots, \alpha_n)$ веса k. Из леммы 1 следует, что если взять первые $T_P(\alpha)$ команд программы P и при любом i таком, что $\alpha_i = 1$, вместо переменной x_i подставить единицу, то получится программа, которая вычисляет линейную функцию от оставшихся $n - k$ переменных. Поэтому в силу леммы 3 при $n \to \infty$ имеем

$$T_P(\alpha) \geqslant (n-k) \log_2 (n-k) - 2(n-k) \gtrsim \frac{1}{2} n \log_2 n.$$

Из этой оценки и асимптотического равенства $\sum\limits_{i=0}^{k} \binom{n}{i} \sim 2^n$ при $n \to \infty$ имеем

$$T_m(P) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} T_P(x) \geqslant \frac{1}{2^n} \sum_{\|x\| \leqslant k} T_P(x) \gtrsim$$

$$\gtrsim \frac{1}{2^n} \sum_{\|x\| \leqslant k} \frac{n \log_2 n}{2} = \frac{n \log_2 n}{2^{n+1}} \sum_{i=0}^{k} \binom{n}{i} \sim \frac{1}{2} n \log_2 n.$$

Теперь покажем, что монотонная сложность функции $x_1 \oplus \cdots \oplus x_n$ есть $\mathcal{O}(n \log_2 n)$. Без ограничения общности будем полагать, что n четно. Известно [5], что сложность вычисления всех n-местных симметрических пороговых монотонных функций Th_k схемами в базисе $\{\&, \vee\}$ по порядку не превосходит величины $n \log_2 n$. Используем этот результат для построения монотонной программы, вычисляющей линейную функцию. Сначала используя $\mathcal{O}(n \log_2 n)$ команд и полагая, что значение переменной y_j равно значению функции Th_j, вычислим все функции Th_k. Затем, начиная с $j = \dfrac{n}{2}$ и заканчивая $j = 1$, добавим $\dfrac{n}{2}$ четверок команд вида:

$$z = 0;\ \texttt{Stop}(y_{2j});\ z = 1;\ \texttt{Stop}(y_{2j-1}).$$

Закончим программу командой $z = 0$. Нетрудно видеть, что такая программа вычисляет функцию $x_1 \oplus \cdots \oplus x_n$ и состоит из $\mathcal{O}(n \log_2 n)$ команд. Теорема 4 доказана.

В следующей теореме находятся порядки монотонной и средней монотонной сложности линейного булева оператора. При этом доказываются квадратичные (относительно числа переменных и числа компонент) нижние оценки монотонной и средней монотонной сложности. Отметим, что в силу теоремы 1 из более чем квадратичных нижних оценок монотонной сложности следуют нелинейные нижние оценки сложности реализации булевых функций и операторов схемами в полном базисе.

Теорема 5. *Пусть $f_n : \{0,1\}^n \to \{0,1\}^n$ — линейный булев оператор, в матрице которого все элементы, кроме стоящих на главной диагонали, равны единице. Тогда при $n \to \infty$*

$$T_m(f_n) \asymp n^2, \qquad C_m(f_n) \asymp n^2.$$

Доказательство. Верхняя оценка монотонной сложности, и, следовательно, средней монотонной сложности легко следует из теоремы 1. Докажем нижние оценки. Рассмотрим монотонную программу P, вычисляющую систему линейных функций f_n. Покажем, что в этой программе для каждого $i = 1, 2 \ldots, n$ найдется не менее $\lfloor (n-1)/2 \rfloor$ вычислительных команд, выходом которых является переменная z_i. Если это не так и некоторая переменная, например, z_1, является выходом менее $\lfloor (n-1)/2 \rfloor$ вычислительных команд, то из леммы 1 следует, что среди n наборов

$$(0, \ldots, 0, 0) \prec (0, \ldots, 0, 1) \prec (0, \ldots, 1, 1) \prec \cdots \prec (0, 1, \ldots, 1)$$

найдется не менее трех последовательных наборов, на которых значения первой компоненты программы P вычисляются одной и той же коман-

дой. Из монотонности этой команды следует, что на двух соседних наборах из этих трех вычисленные значения совпадут, и после прекращения работы программы P на ме́ньшем из этих наборов возникнет равенство

$$z_1(\underbrace{0,\dots,0}_{k},\underbrace{1,\dots,1}_{n-k}) = z_1(\underbrace{0,\dots,0}_{k+1},\underbrace{1,\dots,1}_{n-k-1}),$$

которое, очевидно, противоречит неравенству

$$\underbrace{0\oplus\dots\oplus 0}_{k}\oplus\underbrace{1\oplus\dots\oplus 1}_{n-k} \neq \underbrace{0\oplus\dots\oplus 0}_{k+1}\oplus\underbrace{1\oplus\dots\oplus 1}_{n-k-1}.$$

Таким образом, в программе P каждая переменная z_i $(i = 1, 2\dots, n)$ является выходом не менее $\lfloor (n-1)/2 \rfloor$ вычислительных команд. Следовательно, число вычислительных команд в P не меньше $n\lfloor (n-1)/2 \rfloor$ и при $n \to \infty$

$$C_m(f) \gtrsim \frac{1}{2}n^2. \tag{7}$$

Пусть k — максимальное четное целое, не превосходящее $\lceil n/2 + \sqrt{n}\log_2 n \rceil$. Рассмотрим произвольный набор $\alpha = (\alpha_1, \dots, \alpha_n)$ веса k. Из леммы 1 следует, что $T_P(\beta) \leqslant T_P(\alpha)$ для любого $\beta \succ \alpha$. Поэтому первые $T_P(\alpha)$ команд программы P можно использовать для вычисления системы линейных функций f_{n-k}, если для всех i таких, что $\alpha_i = 1$, вместо переменных x_i подставить единицы. Поэтому в силу (7) при $n \to \infty$ имеем

$$T_P(\alpha) \gtrsim \frac{1}{2}(n-k)^2 \sim \frac{1}{8}n^2. \tag{8}$$

Теперь из леммы 1 и (8) для любого набора γ, содержащего не более k единиц, при $n \to \infty$ имеем

$$T_P(\gamma) \geqslant T_P(\alpha) \gtrsim \frac{1}{8}n^2.$$

Следовательно, из этой оценки и соотношения $\sum\limits_{i=0}^{k}\binom{n}{i} \sim 2^n$ при $n \to \infty$ имеем

$$T(P) = \frac{1}{2^n}\sum_{\alpha\in\{0,1\}^n} T_P(\alpha) \geqslant \frac{1}{2^n}\sum_{\|\alpha\|\leqslant k} T_P(\alpha) \gtrsim$$

$$\gtrsim \frac{1}{2^n}\frac{n^2}{8}\sum_{\|\alpha\|\leqslant k} 1 \sim \frac{1}{2^n}\frac{n^2}{8}\sum_{i=0}^{k}\binom{n}{i} \sim \frac{1}{2^n}\frac{n^2}{8}2^n \sim \frac{n^2}{8}.$$

Теорема 5 доказана.

ЛИТЕРАТУРА

1. **Чашкин А. В.** О среднем времени вычисления значений булевых функций // Дискрет. анализ и исслед. операций. Сер. 1. 1997. Т. 4, № 1. С. 60–78.
2. **Чашкин А. В.** О среднем времени вычисления булевых операторов // Дискрет. анализ и исслед. операций. Сер. 1. 1998. Т. 5, № 1. С. 88–103.
3. **Чашкин А. В.** Среднее время вычисления значений элементарных булевых функций // Дискретная математика. 2000. Т. 12, вып. 4. С. 109–120.
4. **Шоломов Л. А.** О реализации недоопределенных булевых функций схемами из функциональных элементов // Проблемы кибернетики. Вып. 21. М.: Наука, 1969. С. 215–226.
5. **Ajtai M., Komlos Ja., Szemeredi E.** Sorting in $O(n \log n)$ parallel steps // Combinatorica. 1983. V. 3, N. 1. P. 1–19.
6. **Wegener I.** The complexity of Boolean functions. Chechester: John Wiley&Sons; Stuttgart: B.G. Teubner, 1987.

Адрес автора:
МГУ, мех.-мат. факультет,
Воробьевы горы,
119992 Москва, Россия.
E-mail: chash@online.ru

Статья поступила
15 апреля 2004 г.