О СЛОЖНОСТИ РЕАЛИЗАЦИИ СТЕПЕНЕЙ БУЛЕВЫХ (n, n)-ФУНКЦИЙ *)

О. Б. ЛУПАНОВ

(МОСКВА)

Как известно, любая булева функция $f(x_1, \ldots, x_n)$ может быть реализована схемой из функциональных элементов в произвольном конечном базисе со сложностью, асимптотически равной $\rho\frac{2^n}{n}$, где ρ — константа, зависящая от базиса (минимум приведенных весов элементов базиса), и почти все функции $f(x_1, \ldots, x_n)$ требуют именно такой сложности [2, 3, 4]. Возможность реализации функций схемами указанной сложности доказывается эффективно — построением соответствующего метода синтеза. Невозможность сколько-нибудь значительного упрощения схем доказывается неэффективно — «из мощностных соображений»: устанавливается, что число функций, допускающих «более простую реализацию», меньше числа всех функций.

Будем теперь рассматривать преобразования булевых наборов длины n в наборы длины n, т. е. отображения множества наборов $(\sigma_1, \ldots, \sigma_n)$ в себя. Каждому такому отображению F соответствует система n булевых функций $(f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n))$. Поэтому сложность схемы, реализующей такое отображение, не больше сложности реализации n булевых функций, т. е. асимптотически не больше $\rho 2^n$. Нижняя оценка, асимптотически равная верхней, также получается из мощностных соображений.

Усложним задачу. Для данного отображения F и набора $\tilde{\sigma}$ будем искать результат t-кратного применения к набору $\tilde{\sigma}$ отображения F:

$$\underbrace{F(\ldots F(F}_{t \text{ раз}}(\tilde{\sigma}))\ldots)$$

для произвольного t, $t \leqslant t_0$; т. е. сопоставим каждому отображению F вектор-функцию $A_F(\tilde{x}, \tilde{y})$, которая ставит в соответствие наборам $\tilde{\sigma}$ и $\tilde{\tau}$ набор

$$A_F(\tilde{\sigma}, \tilde{\tau}) = \underbrace{F(\ldots F(F}_{|\tilde{\tau}| \text{ раз}}(\tilde{\sigma}))\ldots)$$

(здесь $|\tilde{\tau}|$ — число **), двоичной записью которого является набор $\tilde{\tau}$; пусть для определенности $\tilde{y}$ — набор n переменных), и рассмотрим задачу о слож-

*) Работа 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), программы фундаментальных исследований Отделения математических наук РАН «Алгебраические и комбинаторные методы математической кибернетики» (проект «Оптимальный синтез управляющих систем»).

**) В дальнейшем число $|\tilde{\alpha}|$ будем называть *значением набора* $\tilde{\alpha}$.

ности реализации функции $A_F(\tilde{x}, \tilde{y})$. Нетрудно заметить, что $A_F(\tilde{x}, \tilde{y})$ может быть получена в виде некоторой комбинации отображений $F(\tilde{x})$, $F(F(\tilde{x})) = F^2(\tilde{x})$, $F^4(\tilde{x}), \ldots, F^{2^{n-1}}(\tilde{x})$, т. е. со сложностью, асимптотически не превосходящей $\rho n 2^n$. Оказывается, однако, что многократное (с переменной кратностью) применение отображения F осуществляется асимптотически без увеличения сложности по сравнению с однократным вычислением F, причем это верно не только для отображений, определенных на множестве всех наборов, но и для отображений, определенных на подмножествах наборов.

Сформулируем теперь результат более точно.

Пусть $\mathfrak{M} = \{\tilde{\sigma}_0, \ldots, \tilde{\sigma}_{M-1}\}$ — некоторое множество булевых наборов длины n. Обозначим через $\mathfrak{S}_{\mathfrak{M}}$ множество всех отображений множества $\mathfrak{M}$ в себя. Для каждого F из $\mathfrak{S}_{\mathfrak{M}}$ обозначим через $L(F)$ сложность простейшего доопределения отображения F. Пусть*) $m = \lceil \log M \rceil$.

Пусть $L^*(F)$ — сложность реализации функции $A_F(\tilde{x}, \tilde{y})$, где набор $\tilde{y}$ имеет длину**) m, и $L^*(n, M)$ — максимум сложностей $L^*(F)$ (по всем множествам $\mathfrak{M}$ из M наборов длины n и всем отображениям F из $\mathfrak{S}_{\mathfrak{M}}$).

Основной результат работы состоит в следующем.

Теорема. *Пусть* $\frac{M}{\log n} \to \infty$. *Тогда*

$$L^*(n, M) \sim \rho \frac{Mn}{\log(Mn)}. \tag{1}$$

Замечание 1. Построение схемы для $A_F(\tilde{x})$ асимптотически без увеличения сложности по сравнению со схемой для F (для почти всех отображений F) оказывается возможным из-за того, что «основная сложность» приходится на кодирование отображения F, а дополнительные преобразования имеют существенно меньшую сложность.

Нижняя оценка в (1) получается из мощностных соображений. Соответствующее доказательство приводится в конце статьи.

Доказательство верхней оценки основано на применении метода локального кодирования [3] и некоторого варианта теоремы Д. Улига о сложности одновременной реализации булевой функции на нескольких наборах (асимптотически без увеличения сложности) [9,11]. Из-за практической недоступности работы [11] доказательство соответствующего утверждения приводится в добавлении 1. Используются также результаты Э. И. Нечипорука [7], Л. А. Шоломова [10], А. Е. Андреева [1] и Н. П. Редькина [8] о сложности реализации не всюду определенных функций.

Основная часть статьи — описание метода синтеза, дающего верхнюю оценку. При этом существенно используется граф отображения. Как известно, этот граф состоит из циклов, в которые «врастают» деревья.

Процесс вычисления набора $A_F(\tilde{\sigma}, \tilde{\tau})$ (этот набор будем называть *результирующим*, а набор $\tilde{\sigma}$ — *исходным*) проходит в два этапа. Сначала, если набор $\tilde{\sigma}$ принадлежит некоторому дереву, происходит работа с этим деревом. Если результирующий набор оказывается в дереве, то на этом соответствующая часть вычисления заканчивается. Если же длина пути от вершины, соответствующей набору $\tilde{\sigma}$, до корня дерева меньше значения набора $\tilde{\tau}$, то наступает второй этап — работа с циклом (если вершина, соответствующая набору $\tilde{\sigma}$, находится в цикле, то подсхема для первого этапа «не работает»).

Используются две разные конструкции — для малых значений M ($M \leqslant n^2$) и для больших значений M ($M > n^2$).

---

*) Через log будем обозначать логарифм по основанию 2.

**) Это ограничение может быть значительно ослаблено.

В первом случае сначала от наборов $\tilde{\sigma}$ переходят к их номерам $\tilde{\nu}$; при этом длина набора $\tilde{\nu}$ имеет порядок логарифма от длины исходного набора. Вычисления осуществляются над номерами наборов. Находится номер результирующего набора. Эти вычисления имеют «небольшую сложность». Наконец, по номеру результирующего набора вычисляется сам результирующий набор. Это вычисление составляет основную сложность схемы.

Во втором случае (для больших значений M) множество всех наборов разбивается на подмножества, так что количество подмножеств имеет порядок $(\log M)^c$ и длина записи номера подмножества — порядок $\log\log M$. По набору $\tilde{\sigma}$ вычисляется номер подмножества. Затем по этому номеру и набору $\tilde{\tau}$ вычисляются номера нескольких подмножеств, в объединении которых находится результирующий набор *). Основная (по сложности) часть схемы в этом случае — вычисление по номерам подмножеств списков наборов в этих подмножествах. Именно здесь используется вариант конструкции Д. Улига. Наконец, по набору $\tilde{\sigma}$, спискам наборов в подмножествах и набору $\tilde{\tau}$ вычисляется результирующий набор.

Кодирование деревьев

Здесь будет применяться кодирование деревьев, отличное от «традиционного», основанного на обходе.

Будем рассматривать деревья с корнями, ребра в которых ориентированы к корню (из каждой вершины, отличной от корня, исходит одно ребро, идущее к более близкой к корню вершине). Пусть дерево имеет q ребер и $q+1$ вершин, и вершины занумерованы различными целыми числами от 0 до q. Тогда определена функция $h(x)$, задающая номер вершины, к которой ведет (единственное) ребро, исходящее из вершины с номером x (для всех x, кроме номера корня).

Лемма 1. *Существует нумерация вершин последовательными целыми неотрицательными числами, такая что соответствующая функция h обладает следующими свойствами*:

1) *функция $h(x)$ монотонна, т. е. если $a_1 > a_2$, то $h(a_1) \geqslant h(a_2)$;*

2) *для любого a выполнено неравенство $h(a) > a$.*

Доказательство. Нумерация со свойствами 1) и 2) может быть получена следующим образом. Дерево располагается на плоскости, корень — внизу. Вершины разбиваются на подмножества; в одно подмножество включаются все вершины, находящиеся на одинаковом расстоянии от корня. Сначала нумеруются вершины, находящиеся на максимальном расстоянии от корня — в одном направлении; например, слева направо; потом — в следующем подмножестве — в том же направлении и т. д. (см. рис. 1) **).

---

*) Поскольку известно только подмножество, в котором находится набор $\tilde{\sigma}$, то образ подмножества (в результате сдвига на величину $|\tilde{\tau}|$) может оказаться в объединении нескольких подмножеств.

**) Нумерацию вершин можно осуществить более формально, не прибегая к вложению дерева в плоскость и используя обычное индуктивное построение деревьев с корнями. Определим отношение «вершина a расположена левее вершины b». Для дерева, состоящего из одного ребра, такие пары вершин отсутствуют. Если дерево T' образовано из дерева T добавлением к корню нового ребра (и перемещением корня), то определенное в T отношение переносится на T'. Если дерево T' образовано из деревьев T_1 и T_2 в результате отождествления их корней, то одно из них (например, T_1) объявляется (по определению) левым, а другое (T_2) — правым. Отношение для пар вершин из T_1 и для пар вершин из T_2 переносится на все дерево T'; для каждой пары (a, b) вершин, отличных от корня дерева, где a принадлежит дереву T_1, а b — дереву T_2, по определению вершина a расположена левее вершины b. Далее, в каждом множестве вершин, расположенных на одинаковом расстоянии от корня, эти вершины нумеруются слева направо, начиная с множества самых далеких от корня вершин.

Кодирование деревьев производится фактически так же, как кодирование монотонных (n, m)-функций в [3] — кодируется функция h. Код дерева с q ребрами — это набор из нулей и единиц длины $2q$, имеющий q нулей и q единиц. Нули в коде соответствуют вершинам дерева, отличным от корня. Начало кода — массив из единиц, число которых равно $h(0)$; затем ставится 0 (соответствующий вершине с номером 0); дальше помещается массив из единиц длины $h(1) - h(0)$ (пустой, если $h(1) = h(0)$); затем ставится 0 (соответствующий вершине с номером 1) и т. д. Вообще, между нулями, соответствующими вершинам с номерами $i+1$ и i, помещается массив из единиц длины $h(i+1) - h(i)$.

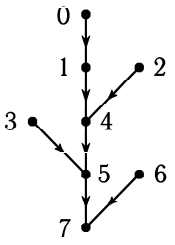


Рис. 1

Очевидно, что по коду дерево восстанавливается однозначно: вершины имеют номера $0, 1, \ldots, q$; для каждого i из вершины с номером i $(0 \leqslant i \leqslant q-1)$ проводится (единственное) ребро в вершину с номером $h(i)$.

Таблица 1

x	$h(x)$
0	1
1	4
2	4
3	5
4	5
5	7
6	7

Пример. Для дерева на рис. 1 соответствующая функция представлена в табл. 1. Код этого дерева имеет вид

$$10111001001100.$$

Для сравнения приведем код, основанный на обходе дерева:

$$00100011011101.$$

Декодирование кодов деревьев

Как уже говорилось, через $|\widetilde{\sigma}|$ будем обозначать число, двоичной записью которого является набор $\widetilde{\sigma}$. Однако в дальнейшем, в случаях, когда это не будет вызывать недоразумений, булевы наборы будут рассматриваться как двоичные записи чисел. Например, запись $\widetilde{\alpha} < \widetilde{\beta}$ будет означать $|\widetilde{\alpha}| < |\widetilde{\beta}|$; вместо выражения «$\widetilde{\gamma}$ есть такой набор, что $|\widetilde{\alpha}| + |\widetilde{\beta}| = |\widetilde{\gamma}|$» мы будем писать «$\widetilde{\gamma} = \widetilde{\alpha} + \widetilde{\beta}$» и т. д. Но иногда, если специально необходимо подчеркнуть, что $\widetilde{\alpha}$ — именно двоичная запись некоторого числа, которое имеет и другое обозначение, мы будем писать $|\widetilde{\alpha}|$.

Пусть D_q — оператор, который по трем наборам

$\widetilde{\varkappa}$ (длины $2q$) — код дерева T с q ребрами,

$\widetilde{\zeta}$ (длины $k = \lceil \log q \rceil$) — номер вершины дерева,

$\widetilde{\lambda}$ (длины k) — двоичная запись некоторого числа

вычисляет наборы

$\widetilde{\mu}$ (длины k) — двоичная запись длины (ориентированного) пути из вершины с номером $\widetilde{\zeta}$ до корня дерева T,

$\widetilde{\xi}$ (длины k) — номер вершины, в которую ведет (ориентированный) путь длины $\widetilde{\lambda}$ в дереве T из вершины с номером $\widetilde{\zeta}$ (если $\widetilde{\lambda} \leqslant \widetilde{\mu}$)

и

δ — один разряд, равный 1, если $\widetilde{\lambda} \leqslant \widetilde{\mu}$, и равный 0, если $\widetilde{\lambda} > \widetilde{\mu}$.

Лемма 2. $L(D_q) \leqslant C_1 q \log q$.

Доказательство. Идея вычисления значений $\widetilde{\mu}$, $\widetilde{\xi}$ и δ состоит в следующем. Последовательно вычисляются значения $\widetilde{\zeta}_1 = F(\widetilde{\zeta})$, $\widetilde{\zeta}_2 = F(\widetilde{\zeta}_1)$, ... до такого v, что $\widetilde{\zeta}_v$, $\widetilde{\zeta}_v = F(\widetilde{\zeta}_{v-1})$, есть корень дерева. В этом случае $|\widetilde{\mu}| = v$. На каждом шаге величина i сравнивается с $|\widetilde{\lambda}|$; если для некоторого i выполняется соотношение $i = |\widetilde{\lambda}|$, то $\widetilde{\xi} = \widetilde{\zeta}_i$. Схема, осуществляющая эти вычисления, получается из схемы для декодирования монотонных вектор-функций (см. [3, с. 65]) добавлением некоторых подсхем. Опишем алгоритм вычисления $\widetilde{\mu}$, $\widetilde{\xi}$ и δ более подробно (правда, при этом будут опускаться некоторые детали «программирования в терминах схем из функциональных элементов»).

На рис. 2 приведен «макет» схемы для дерева, изображенного на рис. 1; указана работа схемы для наборов $\widetilde{\zeta} = (0010)$, $|\widetilde{\zeta}| = 2$, и $\widetilde{\lambda} = (0010)$, $|\widetilde{\lambda}| = 2$.

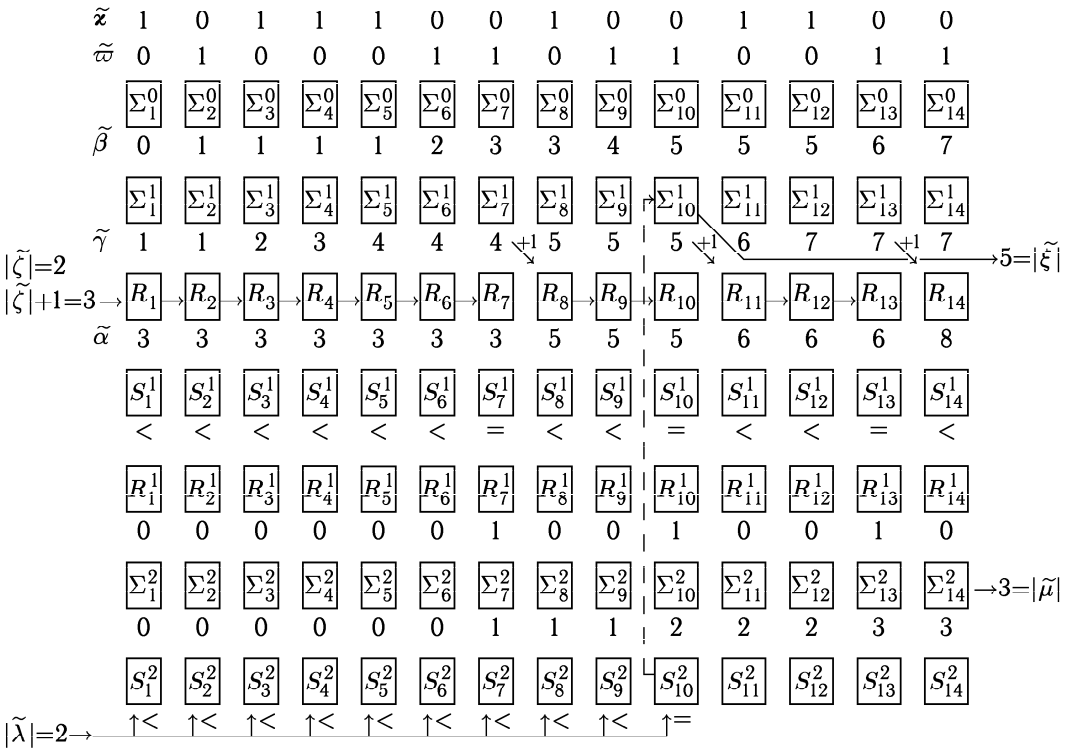


Рис. 2

Образуется инверсия набора $\widetilde{x}$ — набор $\widetilde{\varpi}$. Для этого требуется схема сложности порядка q. Подсчитывается число единиц во всех начальных отрезках наборов $\widetilde{x}$ и $\widetilde{\varpi}$. Это осуществляют две системы k-разрядных сумматоров.

Сумматоры $\Sigma^0_1, \Sigma^0_2, \ldots, \Sigma^0_{2q}$ подсчитывают число единиц во всех начальных отрезках набора $\widetilde{\varpi}$, т. е. число нулей в начальных отрезках набора $\widetilde{x}$. Сумматор Σ^0_i прибавляет i-й разряд набора $\widetilde{\varpi}$ к результату сумматора Σ^0_{i-1}. На его выходе образуется число нулей в первых i разрядах набора $\widetilde{\varpi}$ (сумматор Σ^0_1 на выходе имеет первый разряд набора $\widetilde{\varpi}$). Общая сложность этих схем имеет порядок kq.

Сумматоры $\Sigma^1_1, \Sigma^1_2, \ldots, \Sigma^1_{2q}$ подсчитывают число единиц в начальных отрезках набора $\widetilde{x}$. Сумматор Σ^1_i прибавляет i-й разряд набора $\widetilde{x}$ к ре-

зультату сумматора Σ^1_{i-1}. На его выходе реализуется число единиц в первых i разрядах набора $\tilde{\alpha}$. Соответствующая сложность также имеет порядок*) kq.

Следующая часть схемы позволяет по результатам работы сумматоров Σ^0_i и Σ^1_i последовательно вычислять значения $\tilde{\zeta}_1 = h(\tilde{\zeta})$, $\tilde{\zeta}_2 = h(\tilde{\zeta}_1)$ и т. д. Это осуществляет система $2q$ штук k-разрядных «регистров» (т. е. схем, имеющих на своих выходах определенные наборы) $R_1, \ldots, R_{2q}$, $2q$ штук k-разрядных схем сравнения $S^1_1, \ldots, S^1_{2q}$, $2q$ штук 1-разрядных регистров $R^1_1, \ldots, R^1_{2q}$, $2q$ штук k-разрядных сумматоров $\Sigma^2_1, \ldots, \Sigma^2_{2q}$, $2q$ штук k-разрядных схем сравнения $S^2_1, \ldots, S^2_{2q}$. Их работа происходит следующим образом. В регистр R_1 заносится набор $\tilde{\zeta}'$, такой что $|\tilde{\zeta}'| = |\tilde{\zeta}| + 1$ (нумерация вершин в дереве начинается с 0; поэтому вершине с номером $|\tilde{\zeta}|$ соответствует в коде нуль с номером $|\tilde{\zeta}| + 1$). Схема S_1 сравнивает содержимое $\tilde{\alpha}$ регистра R_1 (в данном случае $\tilde{\alpha} = \tilde{\zeta}$) с результатом $\tilde{\beta}$ сумматора Σ^0_1. Если $\tilde{\beta} < \tilde{\alpha}$, то в регистр R_2 заносится $\tilde{\alpha}$, а в регистр R^1_1 заносится 0 (т. е. «значение функции h еще не вычислено»). Если $\tilde{\alpha} = \tilde{\beta}$, то в регистр R_2 заносится число $|\tilde{\gamma}| + 1$, где $|\tilde{\gamma}|$ — результат сумматора Σ^1_1 (т. е. значение $h(\tilde{\alpha}) + 1$), а в регистр R^1_1 заносится 1 («вычисление осуществлено», «сделан один шаг в дереве»). В общем случае схема S_i сравнивает содержимое $\tilde{\alpha}$ регистра R_i с результатом $\tilde{\beta}$ сумматора Σ^0_i. Если $\tilde{\beta} < \tilde{\alpha}$, то в регистр R_{i+1} заносится $\tilde{\alpha}$, а в регистр R^1_i заносится 0. Если $\tilde{\alpha} = \tilde{\beta}$, то в регистр R_{i+1} заносится результат число $|\tilde{\gamma}| + 1$, где $\tilde{\gamma}$ — результат сумматора Σ^1_i (т. е. значение $h(\tilde{\alpha}) + 1$), а в регистр R^1_i заносится 1 (т. е. «осуществлен один шаг в дереве»). Сумматоры Σ^2_i, $i = 1, \ldots, 2q$, подсчитывают число единиц в первых i регистрах R^1_i. Значение в Σ^2_{2q} равно $\tilde{\mu}$. Разряд δ вычисляет одна схема k-разрядного сравнения: если $\tilde{\lambda} \leqslant \tilde{\mu}$, то $\delta = 1$, если $\tilde{\lambda} > \tilde{\mu}$, то $\delta = 0$. Наконец, для вычисления $\tilde{\xi}$ производится сравнение набора $\tilde{\lambda}$ с результатами сумматоров $\Sigma^2_1, \ldots, \Sigma^2_{2q}$ (схемы сравнения $S^2_1, \ldots, S^2_{2q}$). Находится сумматор Σ^2_j с минимальным номером, имеющий результат $\tilde{\lambda}$. Тогда сумматор Σ^1_j содержит набор $\tilde{\xi}$.

Очевидно, что суммарная сложность всех схем имеет порядок kq, т. е. $q \log q$.

Лемма доказана.

Следствие. *Для любого дерева T с q ребрами существует оператор DT_q, который по двум наборам $\tilde{\zeta}$ и $\tilde{\lambda}$ вычисляет наборы $\tilde{\mu}$, $\tilde{\xi}$ и разряд δ, причем*

$$L(DT_q) \leqslant C_2 q \log q.$$

Оператор DT_q получается из оператора D_q в результате подстановки констант вместо переменных для кода дерева, а именно, кода $\tilde{\varkappa}$ дерева T.

---

*) Система сумматоров Σ^0_i и Σ^1_i позволяет находить значение функции h на всех наборах. Именно, ее значение на произвольном наборе $\tilde{\zeta}$ определяется следующим образом. Надо найти наименьший номер i сумматора Σ^0_i, на выходе которого имеется значение $\tilde{\zeta}$; тогда значение на выходе сумматора Σ^1_i и есть $h(\tilde{\zeta})$.

Пусть UD_q — оператор, который по четырем наборам

$\tilde{\pi}$ (длины $2q$), имеющему для некоторого s ($s \leqslant q$) единицы в первых $2s$ разрядах и нули в остальных разрядах,

$\tilde{x}$ (длины $2q$), имеющему в первых $2s$ разрядах код некоторого дерева T с s ребрами (остальные $2q-2s$ разрядов — произвольные),

$\tilde{\zeta}$, $\tilde{\lambda}$ — как перед формулировкой леммы 2 (но для дерева с s ребрами) вычисляет наборы $\tilde{\mu}$, $\tilde{\xi}$ и разряд δ — как перед формулировкой леммы 2 (но для дерева с s ребрами).

Лемма 2′. $L(UD_q) \leqslant C_3 q \log q$.

Доказательство. Схема для UD_q получается в результате небольшой переделки схемы для D_q (детальное программирование в терминах схем из функциональных элементов здесь опускается).

1) По набору $\tilde{\pi}$ образуется набор $\tilde{\pi}'$, имеющий единственную единицу в $2s$-м разряде.

2) Для каждого i ($1 \leqslant i \leqslant 2q$) результаты работы i-го блока передаются в $(i+1)$-й блок, если $\pi_i = 1$, и полагаются равными 0 или набору $(0, \ldots, 0)$, если $\pi_i = 0$.

3) Наконец, на выход схемы отправляются результаты работы блока с номером $2s$ (с использованием набора $\tilde{\pi}'$).

Граф отображения и нумерация наборов

Как уже говорилось, отображению F поставим в соответствие ориентированный граф G_F, имеющий M вершин и M ребер. Вершины графа G_F соответствуют наборам из $\mathfrak{M}$. Из вершины, соответствующей набору $\tilde{\sigma}$, исходит одно ребро, направленное в вершину, соответствующую набору $F(\tilde{\sigma})$. Таким образом, в графе G_F имеется взаимно однозначное соответствие между вершинами и ребрами: вершине соответствует ребро, исходящее из этой вершины; ребру — вершина, из которой это ребро исходит. В дальнейшем будем называть эту вершину *вершиной, соответствующей ребру.*

Очевидно, что граф G_F состоит из некоторого количества циклов, в которые, быть может, «врастают» деревья с корнями. Корни располагаются на циклах.

Занумеруем наборы из множества $\mathfrak{M}$ следующим образом. Сначала занумеруем наборы, находящиеся в циклах (так же, как это делалось в [5]): $\tilde{\sigma}_1$ — произвольный набор из какого-либо цикла (например, набор, имеющий наименьшее значение); $\tilde{\sigma}_2 = F(\tilde{\sigma}_1)$ (если $\tilde{\sigma}_2 \neq \tilde{\sigma}_1$); $\tilde{\sigma}_3 = F(\tilde{\sigma}_2)$ (если $\tilde{\sigma}_3 \neq \tilde{\sigma}_1$); …; если в этом процессе окажется, что $\tilde{\sigma}_2 = F(\tilde{\sigma}_1) \neq \tilde{\sigma}_1$, $\tilde{\sigma}_3 = F(\tilde{\sigma}_2) \neq \tilde{\sigma}_1$, …, $\tilde{\sigma}_{l_1} = F(\tilde{\sigma}_{l_1-1}) \neq \tilde{\sigma}_1$, но $F(\tilde{\sigma}_{l_1}) = \tilde{\sigma}_1$, то нумерация наборов в этом цикле (длины l_1) заканчивается; выбираем некоторый набор $\tilde{\sigma}_{l_1+1}$ из другого цикла (например, набор из остальных циклов, имеющий наименьшее значение) и продолжаем аналогичным образом процесс дальше, пока не будут исчерпаны все наборы, входящие в циклы.

Нумерация наборов, не входящих в циклы, производится следующим образом. Сначала занумеруем деревья в порядке номеров наборов из циклов, являющихся корнями этих деревьев. Занумеруем теперь очередными номерами (после номеров наборов из циклов) наборы из 1-го дерева, отличные от корня, в соответствии с нумерацией вершин дерева (лемма 1); затем нумеруются вершины из 2-го дерева и т. д. Такую нумерацию всех наборов из $\mathfrak{M}$ будем называть *стандартной.*

Пример. Табл. 2 задает некоторое отображение F_0 множества $\mathfrak{M}_0$ из 11 наборов в себя (в левой части таблицы наборы помещены в порядке задаваемых ими чисел, т. е. в лексикографическом порядке). На рис. 3 приведен граф этого отображения, на рис. 4 — граф части отображения, задаваемой деревьями; на рис. 5 — нумерация вершин большего из деревьев (в соответствии с процедурой леммы 1). Табл. 3 содержит наборы из $\mathfrak{M}_0$, выписанные в порядке стандартной нумерации.

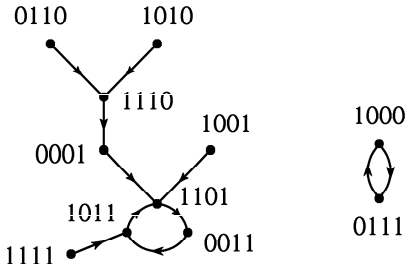


Рис. 3

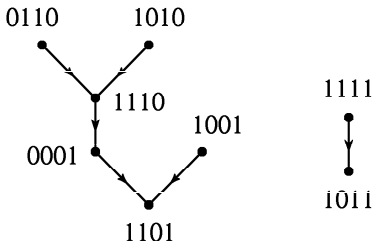


Рис. 4

Таблица 2

$\tilde{\sigma}$	$F_0(\tilde{\sigma})$
0 0 0 1	1 1 0 1
0 0 1 1	1 0 1 1
0 1 1 0	1 1 1 0
0 1 1 1	1 0 0 0
1 0 0 0	0 1 1 1
1 0 0 1	1 1 0 1
1 0 1 0	1 1 1 0
1 0 1 1	1 1 0 1
1 1 0 1	0 0 1 1
1 1 1 0	0 0 0 1
1 1 1 1	1 0 1 1

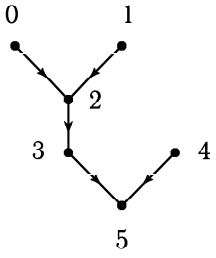


Рис. 5

Таблица 3

0 0 1 1
1 0 1 1
1 1 0 1
0 1 1 1
1 0 0 0
1 1 1 1
0 1 1 0
1 0 1 0
1 1 1 0
0 0 0 1
1 0 0 1

Если набор $\tilde{\sigma}''$ находится на (ориентированном) пути, идущем от вершины $\tilde{\sigma}'$, то будем обозначать через $\varrho(\tilde{\sigma}', \tilde{\sigma}'')$ число ребер в пути от $\tilde{\sigma}'$ до $\tilde{\sigma}''$ (длину пути).

Метод синтеза в случае малых M

Пусть выполнено условие $M \leqslant n^2$.

При получении верхней оценки в случае малых M используются существенно более слабые средства, чем в случае больших M: не требуется теорема Д. Улига; применяются грубые оценки сложности реализации не всюду определенных функций. Поэтому некоторые конструкции, связанные с деревьями, будут изложены позже, при рассмотрении случая больших M.

Опишем схему, вычисляющую результирующий набор.

Сначала по набору $\tilde{\sigma}$ из $\mathfrak{M}$ вычисляется его номер (двоичная запись номера — набор $\tilde{\nu}$ длины $m = \lceil \log M \rceil$) и один разряд ε ($\varepsilon = 0, 1$): $\varepsilon = 0$, если $\tilde{\sigma}$ принадлежит циклу, и $\varepsilon = 1$, если $\tilde{\sigma}$ принадлежит дереву (и не является его корнем). Это осуществляет оператор A_1. Он вычисляет $(n, m+1)$-функцию, определенную на некоторых M наборах длины n.

Как известно (Н. П. Редькин [8]) *), один разряд этой вектор-функции может быть вычислен со сложностью порядка M. Поэтому все $m+1$ разрядов вычисляются со сложностью порядка Mm. Таким образом,

$$L(A_1) \leqslant C_4 M \log M.$$

Дальше вычисления проводятся по-разному для наборов из циклов и для наборов из деревьев.

Наборы из деревьев. Оператор A_2 по набору $\tilde{\nu}$ (номеру набора $\tilde{\sigma}$) при условии, что набор $\tilde{\sigma}$ не принадлежит циклу, и по набору $\tilde{\tau}$

— либо вычисляет номер $\tilde{\nu}$ результирующего набора (если этот набор находится в том же дереве, что и набор $\tilde{\sigma}$),

— либо находит номер $\tilde{\nu}'$ набора в цикле, являющегося корнем дерева, в котором содержится набор $\tilde{\sigma}$, и вычисляет длину «оставшейся части» пути в графе G_F.

Это осуществляется следующим образом. Пусть число деревьев равно t и пусть i-е дерево $T^{(i)}$ имеет q_i ребер $(0 \leqslant i \leqslant t-1)$.

Оператор $A_{2,1}$ по набору $\tilde{\nu}$ вычисляет три набора длины m:

набор $\tilde{\alpha}$ — номер дерева, в котором находится набор $\tilde{\sigma}$ (имеющий номер $\tilde{\nu}$),

набор $\tilde{\beta}$ — наименьший номер набора в этом дереве (т. е. с номером 0 при нумерации вершин внутри дерева),

набор $\tilde{\gamma}$ — номер набора, соответствующего корню этого дерева (этот набор принадлежит циклу; он имеет наибольший номер при нумерации вершин внутри дерева).

Таким образом, оператор $A_{2,1}$ вычисляет некоторую $(m, 3m)$-функцию. Очевидно, что

$$L(A_{2,1}) \leqslant C_5 2^m \leqslant C_6 M.$$

Оператор $A_{2,2}$ по набору $\tilde{\alpha}$ вычисляет набор $\tilde{\zeta}$ длины t с одной единицей в разряде с номером $\tilde{\alpha}$ и нулями на остальных местах («дешифратор»). Это — система конъюнкций длины m (переменных и их отрицаний). Очевидно, что

$$L(A_{2,2}) \leqslant C_7 M.$$

Оператор $A_{2,3}$ по набору $\tilde{\nu}$ вычисляет номер соответствующей вершины в дереве: $\tilde{\zeta} = \tilde{\nu} - \tilde{\beta}$. Очевидно, что

$$L(A_{2,3}) \leqslant C_8 m.$$

---

*) Пусть $\mathfrak{N}$ — некоторое множество наборов длины n и f — функция, определенная на $\mathfrak{N}$. Обозначим через $\tilde{L}(f)$ сложность простейшего доопределения функции f. Пусть

$$\tilde{L}(\mathfrak{N}) = \max \tilde{L}(f)$$

(максимум берется по всем функциям, определенным на $\mathfrak{N}$),

$$\tilde{L}(N, n) = \max \tilde{L}(\mathfrak{N})$$

(максимум берется по всем множествам $\mathfrak{N}$, содержащим N наборов длины n).

Н. П. Редькин установил [8], что для базиса $\{\&, \vee, {}^-\}$ при $N \geqslant 2$

$$\tilde{L}(N, n) \leqslant \frac{3}{2} N - 2.$$

В добавлении 3 приводится более простое доказательство более грубой оценки (для базиса $\{\&, \vee, {}^-\}$):

$$\tilde{L}(N, n) \leqslant 5n.$$

Оператор $A_{2,4}$ состоит из операторов $A_{2,4,i}$ $(0 \leqslant i \leqslant t-1)$. Оператор $A_{2,4,i}$ — это оператор $DT_{q_i}^{(i)}$ для дерева $T^{(i)}$. На входы каждого из этих операторов подаются наборы $\tilde{\zeta}$ и $\tilde{\tau}$. В силу следствия из леммы 2 имеем (поскольку $q_i \leqslant M$ для каждого i и $\sum q_i \leqslant M$)

$$L(A_{2,4}) \leqslant \sum_{i=0}^{t-1} L(A_{2,4,i}) = \sum_{i=0}^{t-1} L(DT_{q_i}^{(i)}) \leqslant \sum_{i=0}^{t-1} C_9 q_i \log q_i \leqslant$$
$$\leqslant \sum_{i=0}^{t-1} C_{10} q_i \log M = C_{10} \log M \sum_{i=0}^{t-1} q_i \leqslant C_{10} M \log M.$$

При этом оператор $A_{2,4,|\tilde{\alpha}|}$ выдает нужную информацию, а остальные — некоторую информацию, которая несущественна для дальнейшего.

Оператор $A_{2,5}$ выбирает из результатов работы операторов $A_{2,4,i}$ нужные значения: выходные наборы оператора $A_{2,4,i}$ умножаются (конъюнктивно) на i-й разряд набора $\tilde{\varphi}$, выдаваемого оператором $A_{2,2}$, и результаты поразрядно (дизъюнктивно) складываются. Очевидно, что

$$L(A_{2,5}) \leqslant C_{11}(2m+1)t \leqslant C_{12} mM \leqslant C_{13} M \log M.$$

В результате получаются два набора

$\tilde{\mu}$ — двоичная запись длины пути от $\tilde{\sigma}$ до корня дерева,

$\tilde{\xi}$ — двоичная запись номера вершины (внутри дерева), в которую ведет путь длины $|\tilde{\tau}|$ из вершины, соответствующей $\tilde{\sigma}$ (если такая есть)

и один разряд δ (равный 1, если $\tilde{\tau} \leqslant \tilde{\mu}$ и равный 0, если $\tilde{\tau} > \tilde{\mu}$).

Оператор $A_{2,6}$ вычисляет номер результирующего набора в случае $\delta = 1$. Этот номер равен $\tilde{\xi} + \tilde{\beta}$. Очевидно, что

$$L(A_{2,6}) = C_{14} m.$$

Оператор $A_{2,7}$ вычисляет информацию для дальнейшей работы с циклической частью (если $\delta = 0$): $\tilde{\sigma}' = \tilde{\gamma}$, $\tilde{\tau}' = \tilde{\tau} - \tilde{\mu}$. Очевидно, что

$$L(A_{2,7}) \leqslant C_{15} m.$$

Наборы из циклов. Здесь, фактически, воспроизводится построение из [5].

Оператор A_3 определяет, с каким набором из цикла и с какой кратностью отображения требуется дальше работать. Если $\varepsilon = 0$, то это $\tilde{\nu}$ и $\tilde{\tau}$; если $\varepsilon = 1$ и $\delta = 0$, то это $\tilde{\nu}' = \tilde{\gamma}$ и $\tilde{\tau}' = \tilde{\tau} - \tilde{\mu}$. Очевидно, что

$$L(A_3) \leqslant C_{16} m.$$

Оператор A_4 по номеру $\tilde{\nu}$ набора из цикла и набору $\tilde{\tau}$ вычисляет номер результирующего набора. Этот оператор состоит из нескольких операторов.

Оператор $A_{4,1}$ по набору $\tilde{\nu}$ вычисляет два набора длины m:

набор $\tilde{\lambda}$ — двоичная запись длины цикла, в котором находится набор $\tilde{\sigma}$,

набор $\tilde{\omega}$ — двоичная запись номера последнего набора в этом цикле.

Это $(m, 2m)$-оператор. Очевидно, что

$$L(A_{4,1}) \leqslant C_{17} M.$$

Оператор $A_{4,2}$ вычисляет остаток от деления $\widetilde{\tau}$ на $\widetilde{\lambda}$ (набор $\widetilde{\pi}$) двоичную запись величины сдвига по циклу. Очевидно, что

$$L(A_{4,2}) \leqslant C_{18} m^2.$$

Оператор $A_{4,3}$ образует набор $\widetilde{\theta} = \widetilde{\nu} + \widetilde{\pi}$. Очевидно, что

$$L(A_{4,3}) \leqslant C_{19} m.$$

Оператор $A_{4,4}$ сравнивает наборы $\widetilde{\theta}$ и $\widetilde{\omega}$, т. е. выясняет, выполнено ли условие $\widetilde{\theta} \leqslant \widetilde{\omega}$. Очевидно, что

$$L(A_{4,4}) \leqslant C_{20} m.$$

Оператор $A_{4,5}$ образует разность $\widetilde{\theta} - \widetilde{\lambda}$. Очевидно, что

$$L(A_{4,5}) \leqslant C_{21} m.$$

Оператор $A_{4,6}$ вычисляет набор $\widetilde{\iota}$ — номер результирующего набора:

$$\widetilde{\iota} = \begin{cases} \widetilde{\theta}, & \text{если } \widetilde{\theta} \leqslant \widetilde{\omega}, \\ \widetilde{\theta} - \widetilde{\lambda}, & \text{если } \widetilde{\theta} > \widetilde{\omega}. \end{cases}$$

Очевидно, что

$$L(A_{4,6}) \leqslant C_{22} m.$$

Таким образом,

$$L(A_4) \leqslant C_{23} m^2.$$

Наконец, оператор A_5 по номеру $\widetilde{\iota}$ результирующего набора вычисляет этот набор. Это осуществляет*) оператор из $\mathfrak{F}^{M,n}$. Легко проверить, что условия теоремы $(*)$ выполнены. Поэтому

$$L(A_5) \lesssim \frac{Mn}{\log(Mn)}.$$

Очевидно, что для случая $M \leqslant n^2$

$$M \log M = o\left(\frac{Mn}{\log(Mn)}\right), \qquad m^2 = o\left(\frac{Mn}{\log(Mn)}\right).$$

Поэтому для этого случая имеем

$$L^*(M, n) \lesssim \frac{Mn}{\log(Mn)}.$$

---

*) Пусть $q = \lceil \log r \rceil$ и $\mathfrak{F}^{r,t}$ — класс (q, t)-функций F, удовлетворяющих условию: если $|\widetilde{\sigma}| \geqslant r$, то $|F(\widetilde{\sigma})| = 0$.

Для сложности реализации вектор-функций из $\mathfrak{F}^{r,t}$ справедливо следующее утверждение.

Теорема $(*)$. *Пусть*

$$r_i \to \infty, \qquad \frac{\log t_i}{r_i} \to 0.$$

Тогда

$$L(\mathfrak{F}^{r_i, t_i}) \sim \frac{r_i t_i}{\log(r_i t_i)}.$$

Доказательство см., например, в [3, с. 49, теорема Д.8] или в [4, с. 82, теорема13].

Случай больших M ($M \geqslant n^2$)

В этом случае будет применяться метод локального кодирования.

Наиболее громоздкой является процедура работы с деревьями.

При построении кодов отображений будут использоваться разбиения деревьев с корнями *). Будем называть *весом дерева* число его ребер; вес дерева T будем обозначать через $|T|$. Справедливо следующее утверждение.

Лемма 3. *Для любого натурального числа W любое ориентированное к корню дерево можно разбить на не пересекающиеся по ребрам поддеревья $T_1, T_2, \ldots$, веса которых удовлетворяют условиям*

$$W \leqslant |T_i| \leqslant 3W.$$

Такие разбиения в дальнейшем будем называть *W-разбиениями*.

Утверждение леммы 3 хорошо известно. Его доказательство приводится в добавлении 2 «для полноты».

Разбиение деревьев на поддеревья, наборов — на подмножества. Пусть Q_1, Q_2, Q_3 — три параметра, удовлетворяющие условиям

$$Q_1 = o(Q_2), \qquad Q_2 = o(Q_3); \tag{2}$$

их значения будут выбраны позже. Разобьем теперь все деревья в G_F на четыре подмножества:

1) деревья *первого размера* — это деревья, веса w которых удовлетворяют условию $w \leqslant Q_1$;

2) деревья *второго размера* — такие что $Q_1 < w \leqslant Q_2$;

3) деревья *третьего размера* — такие что $Q_2 < w \leqslant Q_3$;

4) деревья *четвертого размера* — такие что $Q_3 < w$.

Параметр Q_2 является в некотором смысле основным. Поэтому в дальнейшем индекс 2 при Q иногда будем опускать, т. е. положим

$$Q = Q_2.$$

Разобьем теперь все наборы из $\mathfrak{M}$ на подмножества следующим образом. Подмножества образуют четыре семейства.

Первое семейство. Сначала разобьем на подмножества наборы из циклов. Пусть M_1 — число наборов в циклах и $N_1 = \left\lceil \frac{M_1}{Q} \right\rceil$. Множество $\mathfrak{M}'_0$ содержит первые Q наборов из циклов, $\mathfrak{M}'_1$ — следующие Q наборов и т. д. В общем случае множество $\mathfrak{M}'_j$ ($0 \leqslant j < N_1 - 2$) содержит наборы с номерами $jQ, jQ+1, \ldots, (j+1)Q-1$; множество $\mathfrak{M}'_{N_1-1}$ может содержать меньшее число наборов (из циклов). Далее, добавим к каждому множеству $\mathfrak{M}'_j$ все наборы из деревьев первого размера, корни которых принадлежат $\mathfrak{M}'_j$. В результате образуются множества $\mathfrak{M}_j$, каждое из которых (кроме, быть может, последнего) содержит от Q до QQ_1 наборов.

Второе семейство. Пусть M_2 — число вершин, отличных от корней, в деревьях второго размера.

Разобьем деревья 2-го размера на два класса. В первый класс включим деревья, веса w которых удовлетворяют условию

$$w < \frac{Q}{2}.$$

Во второй класс включим остальные деревья, т. е. деревья, веса w которых удовлетворяют условию

$$\frac{Q}{2} \leqslant w \leqslant Q.$$

*) Сходное, но существенно более простое разбиение графа отображения использовалось для решения другой задачи в работе Ф. Мюллера [6].

Разобьем теперь вершины деревьев из первого класса (отличные от корней) на подмножества так, чтобы выполнялось следующее условие:

(*) Корни всех деревьев, содержащих вершины одного подмножества, принадлежат одному множеству $\mathfrak{M}'_j$.

Будем включать в каждое подмножество (2-го семейства) вершины из наибольшего возможного числа очередных (в порядке их номеров) деревьев, так чтобы число вершин было не более Q. Тогда для каждого фиксированного j может быть образовано несколько (может быть, ни одного) «полных» подмножеств, содержащих от $\frac{1}{2}Q$ до Q вершин, и, может быть, одно «неполное» подмножество, содержащее вершины оставшихся деревьев (2-го размера).

Вершины каждого дерева (отличные от корней) второго класса также образуют множество 2-го семейства.

Таким образом, число «неполных» подмножеств 2-го семейства не превосходит N_1, а число «полных» подмножеств и подмножеств, образованных из деревьев второго класса, не превосходит $\frac{2M_2}{Q}$. Поэтому число N_2 всех подмножеств второго семейства удовлетворяет условию

$$N_2 \leqslant N_1 + \frac{2M_2}{Q}.$$

Заметим, что каждое подмножество из второго семейства содержит не более Q вершин.

Третье семейство. В соответствии с леммой 3 для каждого дерева третьего размера образуем $\lceil \frac{1}{3}Q \rceil$-разбиение. Образовавшиеся поддеревья будем называть *малыми*. Для каждого малого поддерева рассмотрим множество вершин, соответствующих ребрам этого поддерева. Каждое из таких множеств образует подмножество из третьего семейства. Пусть M_3 — общее число вершин в подмножествах третьего семейства. Тогда число N_3 подмножеств, на которые разобьются вершины, удовлетворяет условию

$$N_3 \leqslant \frac{3M_3}{Q}.$$

Четвертое семейство. В соответствии с леммой 3 для каждого дерева четвертого размера образуем $\lceil Q_3 \rceil$-разбиение. Получившиеся поддеревья назовем *средними* поддеревьями. Для каждого среднего поддерева образуем $\lceil \frac{1}{3}Q \rceil$-разбиение (аналогично тому, как это делалось для деревьев третьего размера). Эти поддеревья будем называть *малыми* (так же, как в случае деревьев третьего размера). Разбиение вершин (отличных от корней) в деревьях четвертого размера на подмножества происходит аналогично тому, как это делалось для деревьев третьего размера. Пусть M_4 — число вершин, отличных от корней, в деревьях четвертого размера. Тогда число N_4 подмножеств, на которые разобьются вершины, удовлетворяет условию

$$N_4 \leqslant \frac{3M_4}{Q}.$$

Таким образом, поскольку $M = M_1 + M_2 + M_3 + M_4$, общее число N подмножеств, на которые разбиваются все вершины, оценивается следующим образом:

$$N = N_1 + N_2 + N_3 + N_4 \leqslant 2\left(\frac{M_1}{Q} + 1\right) + \frac{2M_2}{Q} + \frac{3(M_3 + M_4)}{Q} \leqslant \frac{3M}{Q} + 2.$$

Деревья 3-го размера и средние поддеревья деревьев 4-го размера нумеруются различными числами, следующими за номерами подмножеств.

Замечание 2. Малые поддеревья и деревья второго размера обладают следующими свойствами. В каждом малом поддереве и в каждом дереве второго размера

(1) число вершин, отличных от корня, не превосходит Q,

(2) длина пути от любой вершины до его корня не превосходит Q.

Построение частей кода. Части кода — это матрицы ширины $n+7$ и различной высоты. Каждая строка этой матрицы — это

— либо набор из $\mathfrak{M}$ и некоторая дополнительная (конечная) информация,

— либо двоичная запись некоторого числа (такие строки будут встречаться редко).

Будут образованы части кода трех уровней (предварительно будут построены вспомогательные коды циклов и вспомогательные коды деревьев):

— малые части (в дальнейшем они будут называться *кусками кода*) — соответствующие подмножествам, на которые разбиты наборы;

— средние части — соответствующие деревьям 3-го размера и средним поддеревьям деревьев 4-го размера (в случае циклов и деревьев 1-го и 2-го размера они не образуются);

— основные части, имеющие приблизительно одинаковую длину, асимптотически равную новому параметру Q_4.

Образуем сначала вспомогательные коды циклов и вспомогательные коды деревьев.

(1) Вспомогательный код циклов (т. е. фрагментов отображения F, задаваемых циклами в графе G_F).

Кодирование циклов производится так же как в [4]. Пусть $l_1, \ldots, l_k$ — длины циклов и $l = l_1 + \ldots + l_k$ — число наборов, содержащихся в циклах. Вспомогательный код циклов — это матрица, имеющая l строк и $n+1$ столбцов. В первых n столбцах выписаны наборы в порядке описанной выше нумерации наборов в циклах. В последнем столбце стоят единицы в тех и только тех строках, которые содержат последние наборы каждого цикла.

Замечание 3. Дополнительный $(n+1)$-й столбец вводит некоторую избыточность. Относительная избыточность здесь равна $\frac{1}{n}$. Однако это несущественно для получения основного результата — асимптотически точной оценки функции $L^*(n, M)$.

Таблица 4

$\tilde{\sigma}$	$F_0(\tilde{\sigma})$
0 0 1 1	1 0 1 1
1 0 1 1	1 1 0 1
1 1 0 1	0 0 1 1
0 1 1 1	1 0 0 0
1 0 0 0	0 1 1 1

Таблица 5

0 0 1 1	0
1 0 1 1	0
1 1 0 1	1
0 1 1 1	0
1 0 0 0	1

Пример. Табл. 4 задает часть отображения F_0, соответствующую циклам (см. табл. 2), а табл. 5 — вспомогательный код этой части.

(2) Вспомогательный код дерева.

Вспомогательный код фрагмента отображения, задаваемого деревом с q ребрами — это матрица, имеющая $q+1$ строк и $n+2$ столбцов. Образуется нумерация вершин дерева в соответствии с леммой 1. В первых n столбцах матрицы располагаются наборы, соответствующие вершинам дерева, в порядке этой нумерации. В двух следующих столбцах помещается код дерева — набор $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{2q})$ длины $2q$ в некотором определенном порядке (так, как показано в табл. 6).

Таблица 6

ε_1	ε_2
ε_3	ε_4
$\vdots$	$\vdots$
ε_{2q-1}	ε_{2q}
0	0

Замечание 4. Если дерево имеет q ребер, то это кодирование имеет относительную избыточность порядка $\frac{1}{q}$ (так как в код включается набор, соответствующий корню дерева; этот же

набор входит в часть кода, описывающую циклы). Именно для того, чтобы эта избыточность не влияла на асимптотическую оптимальность кодирования в случае малых q, специально выделяются деревья 1-го размера, которые кодируются вместе с наборами из циклов — см. ниже. Кроме того, как и в случае кодирования циклов, здесь имеется еще избыточность порядка $\frac{1}{n}$ — за счет кода дерева.

Таблица 7

0 1 1 0	1 1
1 0 1 0	0 0
1 1 1 0	1 0
0 0 0 1	1 1
1 0 0 1	0 0
1 1 0 1	0 0

Пример. Код большего дерева отображения F_0 в соответствии с принятой нумерацией вершин (рис. 5) имеет вид 1100101100. Табл. 7 представляет вспомогательный код этой части отображения.

Опишем теперь полностью части кода для разных ситуаций.

Кодирование циклов с деревьями 1-го размера. Куски кода строятся в несколько этапов.

(1) Сначала образуется матрица из наборов, входящих в циклы — в том порядке, в каком они входили во вспомогательный код наборов из циклов.

(2) Затем каждый набор, являющийся корнем дерева 1-го размера заменяется списком наборов, относящихся к дереву (1-го размера) в том же порядке, в каком они входили во вспомогательный код дерева.

(3) Образовавшаяся матрица (из наборов из $\mathfrak{M}$) разрезается на части, содержащие (кроме, может быть, последней) Q наборов из циклов; при этом наборы, относящиеся к некоторому дереву (1-го размера), полностью входят в соответствующую часть (содержащую корень этого дерева). Поэтому каждая часть содержит все наборы некоторого множества из 1-го семейства.

(4) Перед каждой частью дополнительно помещаются два набора:

— двоичная запись номера соответствующего подмножества;

— номер 0-го набора из этого подмножества.

(5) Дополнительные столбцы формируются следующим образом (см. рис. 6; в некоторых дополнительных столбцах нули не поставлены):

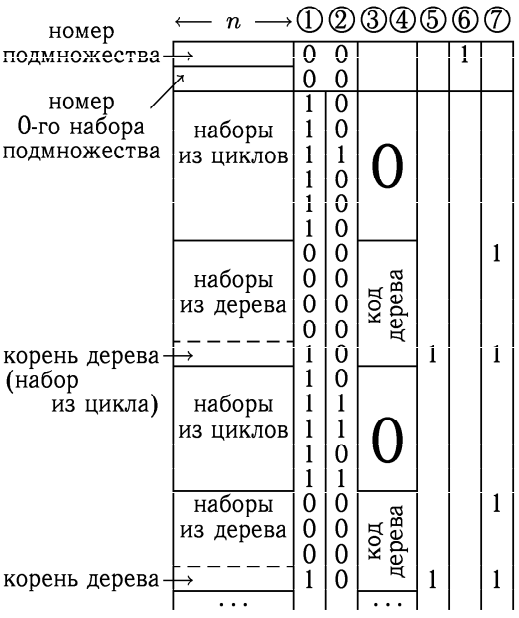


Рис. 6

— 1-й дополнительный столбец содержит единицы в строках, соответствующих наборам из циклов;

— 2-й дополнительный столбец содержит единицы в строках, содержащих последние наборы из циклов (т. е. он определяется дополнительным столбцом вспомогательного кода наборов из циклов);

— 3-й и 4-й дополнительные столбцы в строках, соответствующих наборам из деревьев, содержат коды этих деревьев;

— 5-й дополнительный столбец имеет единицы в строках, содержащих наборы, являющиеся корнями деревьев (в данном случае эта информация избыточная; такая информация будет необходимой для кодирования деревьев 2-го и 3-го размеров, а также средних поддеревьев деревьев 4-го размера);

— 6-й дополнительный столбец имеет единицу в строках, содержащих номера кусков кода;

— 7-й дополнительный столбец имеет единицы в первой и последней строках каждого кода дерева.

Пример. Табл. 8 содержит код отображения F_0 в предположении, что оба дерева — первого размера и $Q = 4$. Код состоит из двух подмножеств. Первое подмножество содержит 4 набора из циклов и 6 наборов из деревьев, второе подмножество — один набор из циклов.

Таблица 8

0 0 0 0	0 0 0 0 0 1 0
0 0 0 0	0 0 0 0 0 0 0
0 0 1 1	1 0 0 0 0 0 0
1 1 1 1	0 0 1 0 0 0 1
1 0 1 1	1 0 0 0 1 0 1
0 1 1 0	0 0 1 1 0 0 1
1 0 1 0	0 0 0 0 0 0 0
1 1 1 0	0 0 1 0 0 0 0
0 0 0 1	0 0 1 1 0 0 0
1 0 0 1	0 0 0 0 0 0 0
1 1 0 1	1 1 0 0 1 0 1
0 1 1 1	1 0 0 0 0 0 0
0 0 0 1	0 0 0 0 0 1 0
0 1 0 0	0 0 0 0 0 0 0
1 0 0 0	1 1 0 0 0 0 0

Деревья 2-го размера. Для каждого дерева наборы выписываются в порядке нумерации соответствующих вершин в дереве; затем (дополнительно) помещается набор, являющийся корнем дерева. Наборы из разных деревьев упорядочиваются в соответствии с номерами корней этих деревьев. В начале множества наборов каждого дерева помещается номер 0-го набора этого дерева. В начале всего куска кода помещается его номер.

Дополнительные столбцы формируются следующим образом (рис. 7):

— 1-й и 2-й столбцы состоят сплошь из нулей;

— 3-й и 4-й столбцы содержат (в соответствующих строках) коды деревьев;

— 5-й столбец имеет единицы в строках, содержащих наборы, являющиеся корнями деревьев;

— 6-й столбец имеет единицу в строке, содержащей номер куска кода;

— 7-й столбец имеет единицы в первой и последней строках кода каждого дерева.

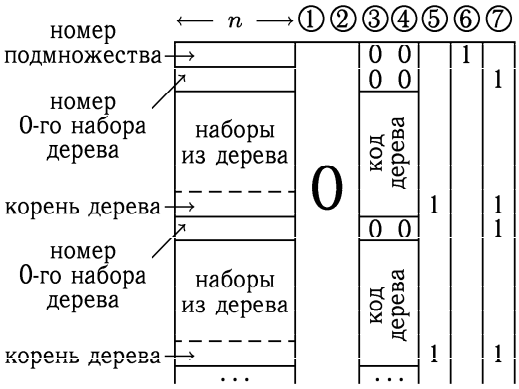


Рис. 7

Деревья 3-го размера. Деревья 3-го размера нумеруются числами, следующими за номерами подмножеств. Наборы, входящие в дерево, располагаются в соответствии с их номерами в дереве (без разбиения на малые поддеревья). В начале помещается номер дерева и номер 0-го набора этого дерева.

Дополнительные столбцы формируются следующим образом (рис. 8):

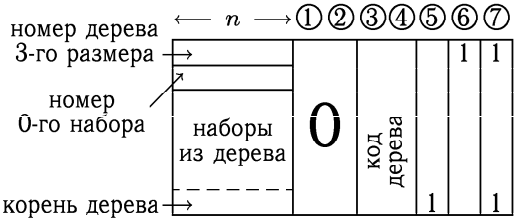


Рис. 8

— 1-й и 2-й столбцы состоят сплошь из нулей;

— 3-й и 4-й столбцы содержат код дерева;

— 5-й столбец имеет единицу в строке, соответствующей набору, являющемуся корнем дерева;

— 6-й столбец имеет единицу в строке, содержащей номер дерева;

— 7-й столбец имеет единицы в строках, соответствующих началу и концу кода дерева.

Деревья 4-го размера. Средние поддеревья нумеруются числами, следующими за номерами деревьев 3-го размера.

Для каждого среднего поддерева дерева 4-го размера T образуем дерево $[T]$ следующим образом. Пусть $\tilde{\beta}$ — корень дерева T. Добавим к T цепочку длины $2Q$ из наборов

$$\tilde{\beta}_1 = F(\tilde{\beta}), \quad \tilde{\beta}_{i+1} = F(\tilde{\beta}_i), \qquad 1 \leqslant i \leqslant 2Q - 1$$

(с соответствующими ребрами).

Это дерево будем называть *удлиненным**). Очевидно, что $|[T]|=|T|+2Q$, и в силу условия (2) $|[T]| \sim |T|$. Код для дерева 4-го размера образуется следующим образом. Для каждого среднего поддерева T образуется удлиненное дерево $[T]$. Для каждого удлиненного дерева входящие в него наборы располагаются в соответствии с их номерами в (удлиненном) дереве (без разбиения на малые поддеревья). В начале списка (для дерева $[T]$) помещается номер дерева $[T]$ и номер 0-го набора поддерева T (в порядке нумерации всех наборов).

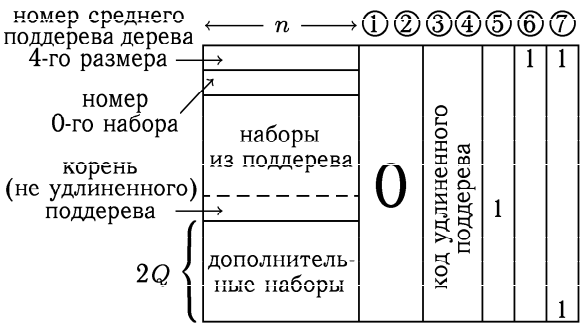


Рис. 9

Дополнительные столбцы формируются следующим образом (см. рис. 9):

— 1-й и 2-й столбцы состоят сплошь из нулей;

— 3-й и 4-й столбцы содержат код удлиненного дерева;

— 5-й столбец имеет единицу в строке, соответствующей набору, являющемуся корнем основного (не удлиненного!) поддерева;

— 6-й столбец имеет единицу в строке, содержащей номер поддерева;

— 7-й столбец имеет единицы в строках, соответствующих началу и концу кода удлиненного дерева.

Построение основных частей кода. Выберем теперь параметр Q_4 так, чтобы выполнялись условия

$$Q_3 = o(Q_4), \qquad QQ_1 = o(Q_4). \tag{3}$$

Каждая основная часть кода имеет длину, асимптотически равную Q_4. При этом вводится некоторая избыточность, не влияющая на асимптотическую оптимальность кодирования. Основные части кода строятся последовательно. Каждая основная часть кода содержит не более Q_4 строк. Сначала берутся куски кода для циклов с деревьями первого размера. При построении очередной основной части кода берется максимально возможное число очередных кусков кода для наборов из циклов и деревьев 1-го размера, так чтобы суммарное количество строк в них не превосходило Q_4. Число наборов (из циклов и деревьев первого размера) в каждой очередной части, кроме, может быть, одной, не менее $Q_4 - QQ_1$. Затем помещаются коды деревьев 2-го размера (при этом в каждую основную часть кода помещаются все наборы из очередного подмножества). Затем помещаются коды деревьев 3-го размера. Наконец, аналогичным образом помещаются коды удлиненных средних поддеревьев деревьев 4-го размера.

Из условий (3) и (2) следует, что общее число R основных частей кода удовлетворяет условию

$$R \sim \frac{M}{Q_4}.$$

*) В цепочке вершины могут повторяться (если цепочка содержит вершины из цикла, и длина цикла меньше $2Q$). Удлиненные деревья вводятся для удобства декодирования. Если результирующий набор для набора с минимальным номером из куска кода, содержащего набор $\tilde{\sigma}$, находится в дереве T, то результирующие наборы для всех наборов из этого куска кода будут находиться в удлиненном дереве $[T]$ (см. замечание 2).

Некоторые обозначения и понятия

Если набор $\tilde{\sigma}$ принадлежит дереву, то обозначим через $r(\tilde{\sigma})$ набор, расположенный на цикле и являющийся корнем дерева, которому принадлежит набор $\tilde{\sigma}$. В дальнейшем для краткости будем обозначать через $[\tilde{\sigma}]$ набор $\tilde{\sigma}$ или $r(\tilde{\sigma})$. Для любого набора $\tilde{\sigma}$ из $\mathfrak{M}$ будем называть *циклом набора* $\tilde{\sigma}$ тот цикл, которому принадлежит набор $[\tilde{\sigma}]$.

Обозначим через $\tilde{\nu}(\tilde{\sigma})$ (или, короче, $\tilde{\nu}$) номер подмножества, которому принадлежит набор $\tilde{\sigma}$, а через $\tilde{\nu}'(\tilde{\sigma})$ (или, короче, $\tilde{\nu}'$) — номер подмножества, которому принадлежит набор $[\tilde{\sigma}]$ (очевидно, что если набор $\tilde{\sigma}$ принадлежит циклу, то $\tilde{\nu}' = \tilde{\nu}$).

Пусть далее (здесь $0 \leqslant d, d', d'' < Q$)

$Qk+d=i$ — номер набора $[\tilde{\sigma}]$,

$Qk' + d'$ — номер начального набора цикла (цикла набора $\tilde{\sigma}$),

$Qk'' + d''$ — номер конечного набора цикла.

Вычисления проходят следующим образом. Сначала определяется «место расположения» набора $\tilde{\sigma}$ в графе G_F:

— выясняется, находится набор в цикле или в дереве;

— определяется номер подмножества, содержащего этот набор;

— выясняется специфика расположения набора $[\tilde{\sigma}]$ в цикле;

— вычисляются номера основных частей кода, в объединении которых находятся набор $\tilde{\sigma}$ и результирующий набор.

Затем, если набор $\tilde{\sigma}$ находится в дереве, работают подсхемы, которые либо вычисляют результирующий набор, либо готовят информацию для работы в цикле набора $\tilde{\sigma}$. Наконец (или сразу, если набор $\tilde{\sigma}$ расположен в цикле), работают подсхемы, связанные с циклом набора $\tilde{\sigma}$.

Цикл будем называть *коротким*, если все его наборы находятся не более чем в двух соседних кусках кода; в противном случае цикл будем называть *длинным*. *Кусок кода* будем называть *средним* (средним куском кода, содержащим наборы некоторого длинного цикла), если все наборы этого куска кода (расположенные в циклах) принадлежат этому циклу; *начальным* (соответственно, *конечным*), если он имеет наименьший (соответственно, наибольший) номер среди кусков кода, содержащих наборы из этого цикла. При этом может оказаться, что наборы некоторого короткого цикла расположены в начальном или конечном куске кода (другого) длинного цикла.

Определим теперь наборы $\tilde{\varepsilon} = (\varepsilon_1, \varepsilon_2, \varepsilon_3)$ и $\tilde{\varepsilon}' = (\varepsilon_1', \varepsilon_2')$.

Набор $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$:

$(0, 0, 0)$ — набор $\tilde{\sigma}$ принадлежит циклу;

$(1, 0, 0)$ — набор $\tilde{\sigma}$ принадлежит дереву 1-го размера;

$(1, 0, 1)$ — набор $\tilde{\sigma}$ принадлежит дереву 2-го размера;

$(1, 1, 0)$ — набор $\tilde{\sigma}$ принадлежит дереву 3-го размера;

$(1, 1, 1)$ — набор $\tilde{\sigma}$ принадлежит дереву 4-го размера.

Набор $(\varepsilon_1', \varepsilon_2')$:

$(0, 0)$ — набор $[\tilde{\sigma}]$ принадлежит короткому циклу;

$(0, 1)$ — набор $[\tilde{\sigma}]$ принадлежит длинному циклу и расположен в начальном куске цикла;

$(1, 0)$ — набор $[\tilde{\sigma}]$ принадлежит длинному циклу и расположен в конечном куске цикла;

$(1, 1)$ — набор $[\tilde{\sigma}]$ принадлежит длинному циклу и расположен в среднем куске цикла.

Замечание 5. Если набор $[\tilde{\sigma}]$ принадлежит длинному циклу, то по наборам $\tilde{\varepsilon}$, $\tilde{\varepsilon}'$, $\tilde{\nu}$ этот цикл однозначно определяется. Поэтому длина $|\tilde{\lambda}|$ этого цикла вычисляется по наборам $\tilde{\varepsilon}$, $\tilde{\varepsilon}'$, $\tilde{\nu}$.

Мы будем использовать обозначение $J = \frac{Mn}{\log(Mn)}$.

Поскольку задачей этой статьи является установление асимптотического соотношения (1), то при построении вспомогательных операторов мы часто будем использовать грубые оценки их сложности, не влияющие на окончательный результат. В частности, будут использоваться подсхемы, вычисляющие $(\lceil C_{(1)} \log\log M + C_{(2)} \rceil, \lceil (C_{(3)} \log M)^{C_{(4)}} + C_{(5)} \rceil)$-функции (для некоторых $C_{(1)}, C_{(2)}, C_{(3)}, C_{(4)}, C_{(5)}$). Множество таких вектор-функций будем обозначать через $\mathfrak{F}_{C_{(1)}, C_{(2)}, C_{(3)}, C_{(4)}, C_{(5)}}$. Очевидно, что каждая функция из $\mathfrak{F}_{C_{(1)}, C_{(2)}, C_{(3)}, C_{(4)}, C_{(5)}}$ может быть реализована со сложностью не более $(C_{(6)} \log M)^{C_{(7)}}$, т. е. $o(J)$. Поэтому мы не будем уточнять значения констант в оценках сложности таких функций и вместо записи

$$F \in \mathfrak{F}_{C_{(1)}, C_{(2)}, C_{(3)}, C_{(4)}, C_{(5)}}$$

будем использовать запись

$$F \in \mathfrak{F}.$$

Положим *)

$$Q_1 = (\log M)^{C_{24}}, \qquad Q = Q_2 = 2^{q_2}, \quad q_2 = \lceil \log M - C_{25} \log\log M \rceil,$$
$$Q_3 = \frac{M}{(\log M)^{C_{26}}}, \quad Q_4 = \frac{M}{(\log M)^{C_{27}}}, \qquad C_{25} - C_{24} > C_{26}, \quad C_{26} > C_{27}, \quad C_{27} > 4.$$

В этом случае число R основных частей удовлетворяет условию

$$R \sim (\log M)^{C_{21}},$$

а длина записи a номера основной части кода удовлетворяет условию

$$a \sim C_{27} \log\log M.$$

Очевидно, что для длины записи a' номеров кусков кода, деревьев 3-го размера и (удлиненных) поддеревьев деревьев 4-го размера выполнено соотношение

$$a' \leqslant C_{28} \log\log M.$$

Разобьем наборы $\tilde{\tau}$ на отрезки длины Q $(Q = 2^{q_2})$ по их значениям: $Qb \leqslant |\tilde{\tau}| < Q(b+1)$. Эти отрезки будем называть *t-отрезками*. Поскольку $|\tilde{\tau}| < 2^m$, то $b < 2^{m-q_2}$, и число t-отрезков имеет порядок $(\log M)^{C_{25}}$; длина записи номера t-отрезка имеет порядок $\log\log M$. Номер t-отрезка определяется старшими $m - q_2$ разрядами набора $\tilde{\tau}$. Обозначим через $\tilde{\tau}^{(b)}$ первый набор из t-отрезка с номером b, т. е. такой набор, что $|\tilde{\tau}^{(b)}| = bQ$.

Вычисление номеров основных частей кода, содержащих результирующий набор

Лемма 4. *Если исходный набор находится в цикле или в дереве 1-го или 2-го размера, то результирующий набор находится либо в том же множестве, что и исходный набор, либо в цикле — в одном из 6 кусков, определяемых наборами $\tilde{\varepsilon}$, $\tilde{\varepsilon}'$, $\tilde{\nu}$ и номером t-отрезка, содержащего набор $\tilde{\tau}$.*

Доказательство. Если результирующий набор находится в дереве (1-го или 2-го размера), то он находится в том же куске кода, что и

*) Нам будет удобно, чтобы параметр Q_2 был степенью двойки; для параметров Q_1, Q_3, Q_4 это несущественно.

исходный набор. Поэтому достаточно рассмотреть случай, когда результирующий набор находится в цикле. Рассмотрим несколько случаев.

(1) Исходный набор $\tilde{\sigma}$ расположен в цикле (простой случай).

Если исходный набор расположен в коротком цикле (а это определяют наборы $\tilde{\varepsilon}$ и $\tilde{\varepsilon}'$), то весь цикл набора $\tilde{\sigma}$ расположен в объединении кусков с номерами $k-1, k, k+1$, а потому и результирующий набор находится в одном из этих кусков.

Пусть теперь исходный набор расположен в длинном цикле (это определяют наборы $\tilde{\varepsilon}$ и $\tilde{\varepsilon}'$). Если $k''-k' \leqslant 5$, то число кусков, в которых расположен весь цикл, не превосходит 6. Поэтому результирующий набор расположен в одном из этих кусков, т. е. в кусках с номерами $k', k'+1, \ldots, k''-1, k''$.

Рассмотрим теперь случай $k''-k' \geqslant 6$.

Обозначим через $Qs+g$ остаток от деления bQ на $|\tilde{\lambda}|$ (здесь $0 \leqslant g \leqslant Q$). Если $i = kQ$ (т. е. $d=0$) и $|\tilde{\tau}| = bQ$, то номер j результирующего набора равен $kQ+sQ+g (\mathrm{mod}\, |\tilde{\lambda}|)$, т. е.

$$j = \begin{cases} kQ+sQ+g, & \text{если } kQ+sQ+g \leqslant k''Q+d'', \\ kQ+sQ+g-|\tilde{\lambda}|, & \text{если } kQ+sQ+g > k''Q+d''. \end{cases}$$

Т. е. значение j однозначно определяется числами k, s, g и $|\tilde{\lambda}|$; при этом числа s и g определяются числами b и $|\tilde{\lambda}|$. Таким образом, в данном частном случае значение j однозначно определяется числами k, b и $|\tilde{\lambda}|$ (а число $|\tilde{\lambda}|$ — числом k и наборами $\tilde{\varepsilon}$ и $\tilde{\varepsilon}'$ — см. замечание 5). Но в общем случае $i = kQ+d$ и $|\tilde{\tau}| = bQ+\Delta$ $(0 \leqslant d < \Delta \leqslant Q-1)$. Поэтому результирующий набор будет находиться на отрезке длины $2Q$ (на цикле), начинающемся набором с номером j. Этот отрезок также однозначно определяется числами k, b и $|\tilde{\lambda}|$. Он может располагаться в нескольких кусках кода (от двух до пяти); возможные номера кусков также однозначно определяются числами k, b и $|\tilde{\lambda}|$ (см. рис. 10; здесь $\hat{k}$ — номер куска кода, содержащего набор с номером j).

(2) Исходный набор находится в дереве 1-го или 2-го размера.

Как и в случае (1), здесь требуется рассмотреть лишь ситуацию, когда $k''-k' \geqslant 6$. В этом случае результирующий набор находится либо в дереве — и поэтому в том же куске кода, что и исходный набор, либо в цикле. Последняя ситуация отличается от рассмотренной в случае (1) тем, что часть пути, идущего от исходного набора до результирующего, может иметь на цикле длину не только из отрезка $[bQ, (b+1)Q-1]$, но и из предыдущего, т. е. $[(b-1)Q, bQ-1]$ (см. рис. 11). Это приводит к возможности нахождения результирующего набора дополнительно в «предыдущем» (на цикле) куске. Соответствующие варианты приведены на рис. 11 (здесь j — номер набора $r(\tilde{\sigma})$, а $\hat{k}$ — номер соответствующего куска кода).

Лемма доказана.

Пусть исходный набор $\tilde{\sigma}$ находится в дереве 3-го или 4-го размера, и пусть T — малое поддерево, содержащее набор $\tilde{\sigma}$ и соответствующее множеству с номером $\tilde{\nu}$. Пусть $\tilde{\sigma}^{(0)}$ — набор с наименьшим номером в поддереве T. Введем обозначение

$$u = \varrho(\tilde{\sigma}^{(0)}, r(\tilde{\sigma})).$$

Пусть набор $\tilde{\tau}$ принадлежит t-отрезку с номером b (т. е. $bQ \leqslant |\tilde{\tau}| < (b+1)Q$). Очевидно, что если

$$bQ \geqslant u,$$

то результирующий набор находится в цикле (в случае дерева и 3-го, и 4-го размера).

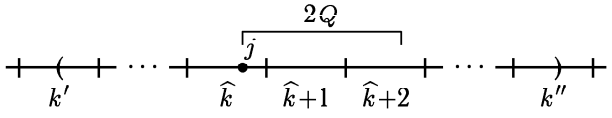
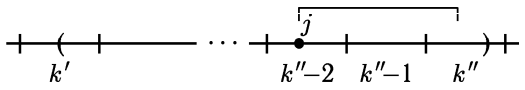
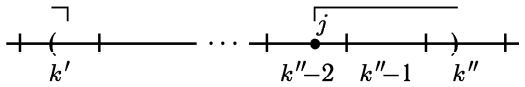
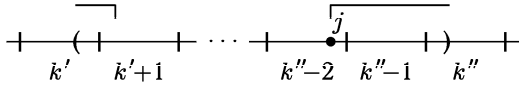
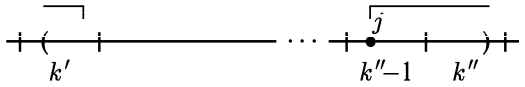
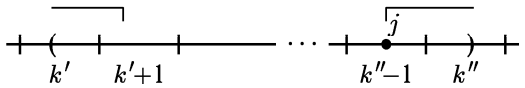
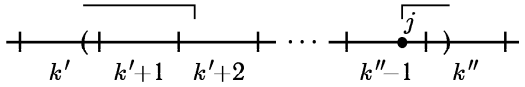
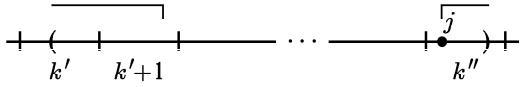
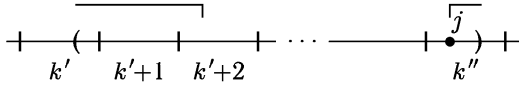
Условие	Ситуации	Номера кусков кода, в которых может находиться результирующий набор
$\widehat{k} \leqslant k''-3$		$\widehat{k}$, $\widehat{k}+1$, $\widehat{k}+2$
$\widehat{k}=k''-2$	а) 	$k''-2$, $k''-1$, k''
	б) 	$k''-2$, $k''-1$, k'', k'
	в) 	$k''-2$, $k''-1$, k'', k', $k'+1$
	Примечание. В случае в) отрезок длины $2Q$ не может пересекаться с куском кода с номером $k'+2$ так как тогда были бы использованы куски с номерами $k''-1$ и $k'+1$ и части других кусков. Таким образом, во всех трех вариантах а), б), в) отрезок длины $2Q$ содержится в объединении кусков кода с номерами $k''-2$, $k''-1$, k'', k', $k'+1$.	
$\widehat{k}=k''-1$	а) 	$k''-1$, k'', k'
	Примечание. В случае а) отрезок длины $2Q$ не может поместиться в объединение только двух кусков кода с номерами $k''-1$ и k''.	
	б) 	$k''-1$, k'', k', $k'+1$
	в) 	$k''-1$, k'', k', $k'+1$, $k'+2$
	Примечание. Во всех трех вариантах а), б), в) отрезок длины $2Q$ содержится в объединении кусков кода с номерами $k''-1$, k'', k', $k'+1$, $k'+2$.	
$\widehat{k}=k''$	а) 	k'', k', $k'+1$
	б) 	k'', k', $k'+1$, $k'+2$
	Примечание. В обоих вариантах а) и б) отрезок длины $2Q$ содержится в объединении кусков кода с номерами k'', k', $k'+1$, $k'+2$.	

Рис. 10

Лемма 5. *Если исходный набор находится в дереве* 3*-го размера, то результирующий набор находится либо в соответствующем дереве, либо в одном из* 6 *кусков кода, определяемых наборами* $\widetilde{\varepsilon}$, $\widetilde{\nu}$ *и номером* t*-отрезка, содержащего набор* $\widetilde{\tau}$.

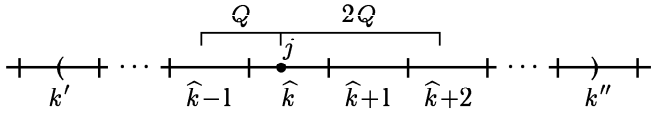
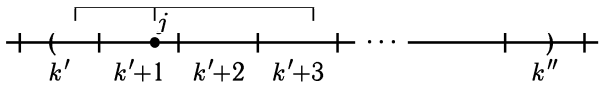
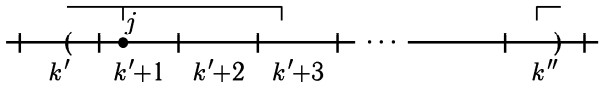
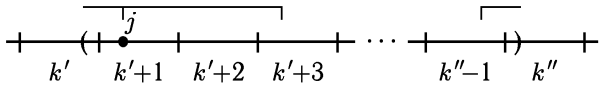
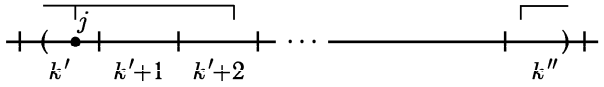
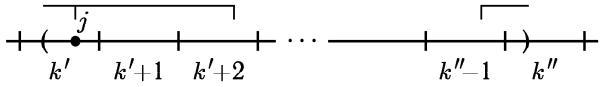
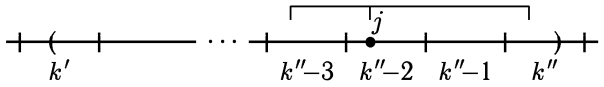
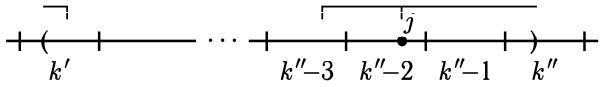
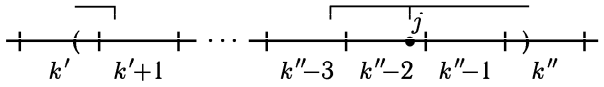
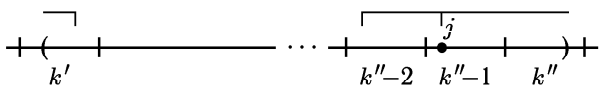
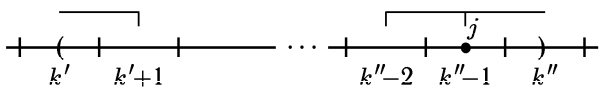
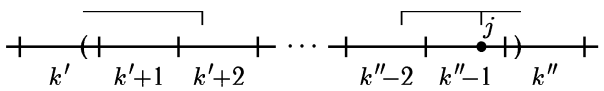
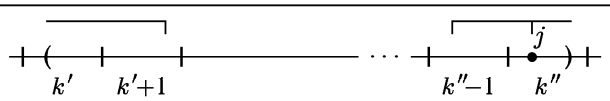
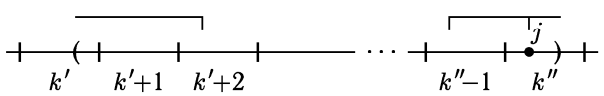
Условие	Ситуации	Номера кусков кода, в которых может находиться результирующий набор
$\widehat{k}\leqslant k''-3$ $\widehat{k}\geqslant k'+2$		$\widehat{k}-1,\ \widehat{k},\ \widehat{k}+1,\ \widehat{k}+2$
$\widehat{k}=k'+1$	а) 	$k',\ k'+1,\ k'+2,\ k'+3$
	б) 	$k',\ k'+1,\ k'+2,\ k'+3,\ k''$
	в)  Во всех трех вариантах отрезок длины $3Q$ содержится в объединении кусков кода $k',\ k'+1,\ k'+2,\ k'+3,\ k''-1,\ k''$.	$k',\ k'+1,\ k'+2,\ k'+3,$ $k''-1,\ k''$
$\widehat{k}=k'$	а) 	$k',\ k'+1,\ k'+2,\ k''$
	б)  В обоих вариантах отрезок длины $3Q$ содержится в объединении кусков кода $k',\ k'+1,\ k'+2,\ k''-1,\ k''$.	$k',\ k'+1,\ k'+2,\ k'+3,$ $k''-1,\ k''$
$\widehat{k}=k''-2$	а) 	$k''-3,\ k''-2,\ k''-1,\ k''$
	б) 	$k',\ k''-3,\ k''-2,\ k''-1,\ k''$
	в)  Во всех трех вариантах отрезок длины $3Q$ содержится в объединении кусков кода $k',\ k'+1,\ k''-3,\ k''-2,\ k''-1,\ k''$.	$k',\ k'+1,$ $k''-3,\ k''-2,\ k''-1,\ k''$
$\widehat{k}=k''-1$	а) 	$k',\ k''-2,\ k''-1,\ k''$
	б) 	$k',\ k'+1,\ k''-2,\ k''-1,\ k''$
	в)  Во всех трех вариантах отрезок длины $3Q$ содержится в объединении кусков кода $k',\ k'+1,\ k'+2,\ k''-2,\ k''-1,\ k''$.	$k',\ k'+1,\ k'+2,$ $k''-2,\ k''-1,\ k''$
$\widehat{k}=k''$	а) 	$k',\ k'+1,\ k''-1,\ k''$
	б)  В обоих вариантах отрезок длины $3Q$ содержится в объединении кусков кода $k',\ k'+1,\ k'+2,\ k''-1,\ k''$.	$k',\ k'+1,\ k'+2,\ k''-1,\ k''$

Рис. 11

Доказательство. Очевидно, что если

$$bQ \geqslant u,$$

то результирующий набор находится в цикле. Здесь ситуация аналогична рассмотренной в лемме 4. Как и при доказательстве леммы 4, достаточно рассмотреть случай $k'' - k' \geqslant 6$. В этом случае вместо величины $|\tilde{\tau}|$ следует рассмотреть величину $bQ - u$ (заметим, что u однозначно определяется по номеру $\tilde{\nu}$ подмножества, содержащего исходный набор $\tilde{\sigma}$, и числу b). В этом случае необходимо учесть две неопределенности (по Q): за счет неточности задания $|\tilde{\tau}|$ — только номером t-отрезка; за счет неточности положения исходного набора в подмножестве (малом дереве). Третья неопределенность — положение корня $r(\tilde{\sigma})$ всего дерева 3-го размера в куске кода может не учитываться, так как $r(\tilde{\sigma})$ однозначно определяется всем деревом, а дерево — набором $\tilde{\tau}$ (номером подмножества).

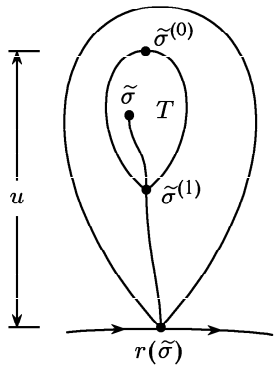


Рис. 12

Если

$$bQ \leqslant u - 2Q, \tag{4}$$

то результирующий набор находится в рассматриваемом дереве 3-го размера, и потому в той же основной части кода, что и исходный набор $\tilde{\sigma}$. В самом деле (см. рис. 12; здесь T — малое поддерево, содержащее набор $\tilde{\sigma}$, а $\tilde{\sigma}^{(1)}$ — корень этого поддерева).

$$\begin{aligned}\varrho(\tilde{\sigma}, \tilde{\sigma}_\omega) = |\tilde{\tau}| < bQ + Q \leqslant (\text{учитывая } (4)) \leqslant u - Q = \\ = \varrho(\tilde{\sigma}^{(0)}, r(\tilde{\sigma})) - Q = \varrho(\tilde{\sigma}^{(0)}, \tilde{\sigma}^{(1)}) + \varrho(\tilde{\sigma}, r(\tilde{\sigma})) - \varrho(\tilde{\sigma}, \tilde{\sigma}^{(1)}) - Q \leqslant \\ \leqslant \varrho(\tilde{\sigma}^{(0)}, \tilde{\sigma}^{(1)}) + \varrho(\tilde{\sigma}, r(\tilde{\sigma})) - Q \leqslant \varrho(\tilde{\sigma}, r(\tilde{\sigma})).\end{aligned}$$

Если

$$u - 2Q < bQ < u,$$

то результирующий набор находится либо в рассматриваемом дереве 3-го размера, либо в цикле, на расстоянии менее $2Q$ от набора $r(\tilde{\sigma})$, т. е. в объединении не более 7 основных частей кода (одна часть — содержащая дерево, и не более 6 частей, содержащих 6 кусков кода из циклов).

Лемма доказана.

Лемма 6. *Если исходный набор находится в дереве 4-го размера, то результирующий набор находится либо в удлиненном дереве среднего поддерева, содержащего данный набор, либо в другом удлиненном дереве* $[T']$, *либо в цикле, в одном из 6 кусков кода. Номер дерева* T' *и номера кусков кода в цикле однозначно определяются наборами* $\tilde{\varepsilon}, \tilde{\varepsilon}'\tilde{\nu}$ *и номером* b t*-отрезка, содержащего набор* $\tilde{\tau}$.

Доказательство. Как и в доказательстве леммы 5, пусть исходный набор $\tilde{\sigma}$ находится в множестве с номером $\tilde{\nu}$, определяющем поддерево T.

Если $bQ \geqslant u$, то результирующий набор находится в цикле, и дальнейшая часть рассуждения повторяет сказанное при доказательстве леммы 5.

Пусть теперь $bQ < u$. Рассмотрим набор $\tilde{\sigma}'$, такой что $\varrho(\tilde{\sigma}^{(0)}, \tilde{\sigma}') = bQ$. В этом случае набор $\tilde{\sigma}'$ может находиться либо в цикле (первый случай), либо в исходном дереве 4-го размера (второй случай), причем какой из случаев имеет место, однозначно определяется набором $\tilde{\nu}$ и числом b. В первом случае ситуация аналогична рассмотренной в доказательстве леммы 5. Во втором случае рассмотрим среднее поддерево T', содержащее набор $\tilde{\sigma}'$.

Так же, как и в доказательстве леммы 5, можно убедиться в том, что результирующий набор находится в удлиненном дереве $[T']$. Номер этого дерева однозначно определяется набором $\tilde{\nu}$ и числом b.

Таким образом, результирующий набор находится не более чем в 7 основных частях кода — не более 6 в цикле и в основной части, содержащей удлиненное дерево $[T']$.

Лемма доказана.

Лемма 7. *Существует оператор A_R из $\mathfrak{F}$, который по наборам $\tilde{\varepsilon}$, $\tilde{\varepsilon}'$, $\tilde{\nu}$ и номеру b t-отрезка, содержащего набор $\tilde{\tau}$, вычисляет номера 8 различных основных частей кода, в объединении которых содержатся исходный набор, набор $r(\tilde{\sigma})$ и результирующий набор.*

Доказательство. В случае, если исходный набор принадлежит циклу или дереву 1-го или 2-го размера, это основная часть кода, содержащая исходный набор (и набор $r(\tilde{\sigma})$) и не более 6 основных частей кода, содержащих соответствующие куски кода (в цикле). В случае дерева 3-го размера — основная часть кода, содержащая исходный набор (она же содержит и набор $r(\tilde{\sigma})$ — корень дерева) и не более 6 кусков кода (в цикле). В случае дерева 4-го размера это либо основные части кода, содержащие коды двух удлиненных средних деревьев (если результирующий набор содержится в дереве), либо основные части кода, содержащие исходный набор $\tilde{\sigma}$, набор $r(\tilde{\sigma})$ и не более 6 кусков кода (в цикле)(если результирующий набор содержится в цикле). Из лемм 4 — 6 следует, что потребуется не более 8 основных частей кода с требуемыми свойствами. Для удобства декодирования к ним добавляются еще некоторые основные части, так чтобы оказалось ровно 8 различных основных частей кода.

Принадлежность оператора A_R множеству $\mathfrak{F}$ следует из того, что длины наборов $\tilde{\varepsilon}$, $\tilde{\varepsilon}'$, $\tilde{\nu}$ и b, а также номера основных частей кода имеют порядок $\log\log M$.

Лемма доказана.

Замечание 6. Число (восемь) основных частей кода, объединение которых содержит исходный набор и результирующий набор, не является принципиальным. Важно, что это число ограничено абсолютной константой.

Вычисление результирующего набора. Построение схемы

Оператор*) A_6 по набору $\tilde{\sigma}$ вычисляет наборы $\tilde{\varepsilon}$, $\tilde{\varepsilon}'$ и $\tilde{\nu}$. Этот оператор реализует некоторую $(n, d+5)$-функцию, определенную на M наборах. Поскольку $M > n^2$, то в соответствии с теоремой (**)**) один разряд этой функции вычисляется со сложностью, асимптотически равной $\frac{M}{\log M}$, а все

*) Здесь мы продолжаем нумерацию операторов и схем, начатую при рассмотрении случая $M \leqslant n^2$.

**) Теорема (**). *Пусть*

$$\frac{N}{n(\log n)^{1+\delta}} \to \infty \tag{5}$$

(*здесь δ — произвольная положительная константа*). *Тогда*

$$\tilde{L}(N, n) \sim \frac{N}{\log N}.$$

Это утверждение доказано Л. А. Шоломовым [10, с. 216] с использованием результата Э. И. Нечипорука о доопределениях булевых наборов [7]. Впоследствии А. Е. Андреев ослабил условие (5) до минимально необходимого

$$\frac{N}{n\log n} \to \infty$$

(см. [1, с. 44, теорема 1]).

разряды — со сложностью асимптотически равной $\frac{2dM}{\log M}$, т. е. имеющей порядок *)не более $\frac{M \log\log M}{\log M}$. Легко проверить, что

$$L(A_6) = o(J).$$

Оператор A_7, в случае, если набор $[\widetilde{\sigma}]$ принадлежит длинному циклу, по наборам $\widetilde{\varepsilon}, \widetilde{\varepsilon}', \widetilde{\nu}$ вычисляет набор $\widetilde{\lambda}$ — длину цикла, содержащего набор $[\widetilde{\sigma}]$ (см. замечание 5). Очевидно, что

$$A_7 \in \mathfrak{F}.$$

Оператор A_8, в случае, если набор $[\widetilde{\sigma}]$ принадлежит длинному циклу, по наборам $\widetilde{\varepsilon}, \widetilde{\varepsilon}', \widetilde{\nu}$ вычисляет числа k', k'', d', d''. Очевидно, что

$$A_8 \in \mathfrak{F}.$$

Оператор A_9 по номерам $l_1, \ldots, l_8$ основных частей кода (содержащих в совокупности исходный набор и результирующий набор) вычисляет эти части. Здесь мы воспользуемся вариантом теоремы Д. Улига об одновременном вычислении значений булевой функции на нескольких наборах значений переменных асимптотически без увеличения сложности (по сравнению с вычислением значения на одном наборе), о которой говорилось в начале статьи (см. добавление 1, теорема Д1). Легко проверить, что условия этой теоремы выполнены: здесь $t=8$ ($T=3$), $r=R$, $p=\lceil Q_4 \rceil$. Поэтому

$$L(A_9) \lesssim \frac{Mn}{\log(Mn)}.$$

Введем обозначения. Пусть $(\alpha_0, \alpha_1, \ldots, \alpha_{l-1})$ — произвольный набор **) и $(\alpha_k, \alpha_{k+1}, \ldots, \alpha_{k+h-1})$ — произвольный отрезок этого набора. Числа k (номер начального разряда отрезка) и h (длина отрезка) или двоичные записи этих чисел будем называть *координатами* этого отрезка.

Пусть W_l — оператор, который по произвольному набору $\widetilde{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{l-1})$ длины l и двум наборам $\widetilde{\beta}'$ и $\widetilde{\beta}''$ длины d (где $d=\lceil \log l \rceil$, $|\widetilde{\beta}'|=k$, $|\widetilde{\beta}''|=h$) строит набор $(\alpha_k, \alpha_{k+1}, \ldots, \alpha_{k+h-1}, 0, \ldots, 0)$ (будем говорить, что оператор W_l *выделяет* в наборе $\widetilde{\alpha}$ *часть* этого набора *с координатами* $k=|\widetilde{\beta}'|$ и $h=|\widetilde{\beta}''|$).

Как известно (см., например, [3, с. 42–43]),

$$L(W_l) \leqslant C_{31} l \log l.$$

Дальше работа проходит в два этапа.

Э т а п 1. Набор $\widetilde{\sigma}$ принадлежит дереву. В этом случае либо результирующий набор вычисляется сразу (если он находится в том же поддереве, что и набор $\widetilde{\sigma}$ (т. е. в дереве 1-го, 2-го или 3-го размера, или в среднем поддереве в случае дерева 4-го размера), либо вычисляется информация для последующей работы в цикле или в другом поддереве в случае дерева 4-го размера.

---

*) Здесь нет необходимости использовать оценку теоремы (**) в полном объеме. Достаточно, например, оценки с точностью до порядка или несколько более грубой оценки. Однако грубой оценки из сноски из Добавления 3 недостаточно.

**) В данном случае удобно нумеровать разряды, начиная с 0.

Пусть $Q' = Q_3 + 2Q_2$. Будем называть *частью кода* F, *соответствующей* (быть может, удлиненному) *дереву* T, матрицу размера $Q' \times (n+8)$, изображенную на рис. 13 (здесь s' — высота кода дерева T).

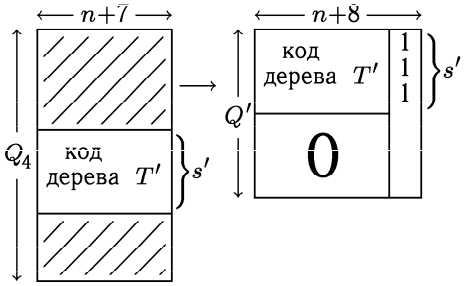


Рис. 13

Пусть выполнены условия

(1) основная часть кода F содержит код дерева T 1-го, 2-го, 3-го размера или код (удлиненного) среднего поддерева дерева 4-го размера,

(2) набор $\tilde{\sigma}$ содержится в дереве T.

Пусть оператор DKU по основной части кода F и набору $\tilde{\sigma}$ образует часть кода F, соответствующую дереву *) T.

Лемма 8. *Существует схема* S *для* DKU, *такая что*

$$L(S) = o(J).$$

Доказательство. Схема S состоит из трех подсхем S', S'' и S'''.

Подсхема S' вычисляет координаты кода поддерева T в основной части кода F. Сначала отмечается строка, содержащая набор $\tilde{\sigma}$, посредством сравнения (на совпадение) этого набора со всеми наборами из $\mathfrak{M}$ в F — в виде набора (столбца), имеющего единственную единицу в строке (в разряде), содержащей набор $\tilde{\sigma}$; пусть эта строка имеет номер i_0. Это осуществляет схема сложности не более $C_{32}Q_4 n$. Затем определяется номер i_1 строки, ближайшей сверху в коде к i_0-й строке, имеющей единицу в 5-м дополнительном столбце кода — это строка, с которой начинается код поддерева, содержащего набор $\tilde{\sigma}$ (очевидно, что $i_1 < i_0$). Это осуществляет схема сложности не более $C_{33}Q_4 \log Q_4$. Аналогично определяется номер i_2 строки, ближайшей снизу в коде к i_0-й строке, имеющей единицу в 5-м дополнительном столбце кода — это строка, на которой оканчивается код поддерева T. Пусть $\tilde{\psi}^{(1)}$, $\tilde{\psi}^{(2)}$, $\tilde{\psi}^{(3)}$ — такие наборы, что $|\tilde{\psi}^{(1)}| = i_1$, $|\tilde{\psi}^{(2)}| = i_2$, $|\tilde{\psi}^{(3)}| = i_2 - i_1 + 1$ (длина кода). Тогда $\tilde{\psi}^{(1)}$ и $\tilde{\psi}^{(3)}$ — координаты кода дерева T. Поэтому сложность подсхемы S' оценивается следующим образом:

$$L(S') \leqslant C_{33}Q_4(n + \log Q_4).$$

Подсхема S'' выделяет в основной части кода код поддерева T. Это осуществляют $n+7$ штук схем выбора части набора W_{Q_4}. Поэтому

$$L(S'') \leqslant (n+7)L(W_{Q_4}) \leqslant C_{34}nQ_4 \log Q_4.$$

Легко проверить, что

$$L(S'') = o(J).$$

В самом деле, поскольку $M > n^2$, то $n < \sqrt{M}$ и $\log(Mn) \asymp \log M$. Поэтому

$$\frac{L(S'')}{J} \leqslant \frac{nQ_4 \log Q_4}{\frac{Mn}{\log(Mn)}} \asymp \frac{nQ_4(\log Q_4)\log(Mn)}{Mn} \asymp \frac{Q_4(\log M)^2}{M} \asymp \frac{1}{(\log M)^{C_{27}-2}} \to 0.$$

*) Индекс Q_4 в обозначении оператора DKU опускаем.

Наконец, подсхема S''', используя набор $\widetilde{\psi}^{(3)}$, строит 8-й столбец части кода. Легко видеть, что

$$L(S''') \leqslant C_{35} Q_4.$$

Таким образом,

$$L(S) = o(J).$$

Пусть DK — оператор, который по матрице размера $Q' \times (n+8)$, содержащей код дерева, и по наборам $\widetilde{\sigma}$ (из дерева) и $\widetilde{\tau}$ вычисляет

(1) разряд δ, равный 1, если результирующий набор находится в дереве, и равный 0, если результирующий набор находится вне дерева (исходного, не удлиненного),

(2) при $\delta = 1$ — результирующий набор,

(3) при $\delta = 0$ — набор $\widetilde{\sigma}^*$, являющийся корнем дерева T,

(4) расстояние $\widetilde{\tau}'$ от набора $\widetilde{\sigma}^*$ до результирующего набора.

Л е м м а 9. *Существует схема S для оператора DK, такая что*

$$L(S) \leqslant C_{36} Q'(n + Q').$$

Д о к а з а т е л ь с т в о. Сначала к матрице K добавляется матрица, состоящая из номеров всех наборов из $\mathfrak{M}$, содержащихся в дереве (включая корень); образуется матрица K'. Это осуществляет схема $S_{(1)}$. Очевидно, что существует схема $S_{(1)}$, такая что

$$L(S_{(1)}) \leqslant C_{37} Q' \log Q'.$$

Следующая схема $(S_{(2)})$ сравнивает (на совпадение) набор $\widetilde{\sigma}$ с каждым набором из дерева и, используя результаты работы схемы $S_{(1)}$, определяет номер $\widetilde{p}_{(0)}$ набора $\widetilde{\sigma}$ в дереве. Очевидно, что существует схема $S_{(2)}$, такая что

$$L(S_{(2)}) \leqslant C_{38} Q' n.$$

Затем работает схема $S_{(3)}$ — это схема декодирования $UD_{Q'}$;

$$L(S_{(3)}) \leqslant C_{39} Q' \log Q'.$$

Эта схема вычисляет разряд δ и при $\delta = 1$ номер $\widetilde{p}_{(1)}$ результирующего набора (в дереве); при $\delta = 0$ вычисляется расстояние $\widetilde{\mu}$ от набора $\widetilde{\sigma}$ до корня дерева.

Затем работает схема $S_{(4)}$. Если $\delta = 1$, то по номеру $\widetilde{p}$ результирующего набора (в дереве) с помощью матрицы K' определяется сам результирующий набор. Если $\delta = 0$, то выделяется корень дерева; величина $\widetilde{\tau}'$ определяется соотношением $\widetilde{\tau}' = \widetilde{\tau} - \widetilde{\mu}$.

З а м е ч а н и е 7. Схема S вычисляет результирующий набор, если он принадлежит дереву 1-го, 2-го и 3-го размера, а также дереву 4-го размера, если результирующий набор принадлежит тому же (быть может, удлиненному) среднему дереву, что и исходный набор.

Рассмотрим теперь случай, когда исходный набор принадлежит дереву 4-го размера, но результирующий набор принадлежит (удлиненному) среднему дереву, отличному от исходного.

Пусть

$$bQ < u.$$

В этом случае удлиненное среднее поддерево $[T']$, содержащее результирующий набор, однозначно определяется набором $\widetilde{\nu}$ и числом b.

Пусть A_{10} — оператор, который по $\tilde{\nu}$ и b определяет

(1) номер l' удлиненного среднего поддерева $[T']$, содержащего результирующий набор,

(2) номер $s(\tilde{\sigma})$ вершины (в нумерации вершин поддерева T'), через которую путь, ведущий от исходного набора к результирующему набору, входит в поддерево T',

(3) расстояние $|\tilde{\tau}''|$ от корня $\tilde{\sigma}^*$ среднего (не удлиненного) поддерева, содержащего исходный набор, до вершины $s(\tilde{\sigma})$.

Очевидно, что

$$A_{10} \in \mathfrak{F}.$$

Пусть A_{11} — оператор, который по числам $|\tilde{\tau}'|$ и $|\tilde{\tau}''|$ определяет длину пути $|\tilde{\tau}'''|$ от набора $s(\sigma)$ до результирующего набора $\tilde{\sigma}_\omega$. Очевидно, что $|\tilde{\tau}'''| = |\tilde{\tau}'| - |\tilde{\tau}''|$. Поэтому

$$L(A_{11}) \leqslant C_{40} \log M.$$

Пусть A_{12} — оператор, который по 8 основным частям кода и номеру l' среднего дерева образует часть кода для поддерева с номером l'.

Лемма 10. $L(A_{12}) = o(J)$.

Доказательство аналогично доказательству леммы 8. Отличие заключается в том, что вместо исходного набора $\tilde{\sigma}$ используется номер l' поддерева, который сравнивается со всеми номерами кусков кодов и поддеревьев, т. е. со всеми наборами во всех 8 основных частях кода, отмеченными единицами в 6-м дополнительном столбце кода.

Пусть DK'' — оператор, который по матрице размера $Q \times (n+8)$, содержащей код удлиненного поддерева $[T']$, номеру набора $s(\tilde{\sigma})$ и длине $\tilde{\tau}''$ пути от набора $s(\tilde{\sigma})$ до результирующего набора $\tilde{\sigma}_\omega$ определяет результирующий набор.

Лемма 11. *Существует схема S для оператора DK'', такая что*

$$L(S) \leqslant C_{41} Q'(n + Q').$$

Доказательство аналогично доказательству леммы 9. Отличие состоит в том, что не следует искать начальный набор в поддереве T' — это набор $s(\tilde{\sigma})$. Не следует вычислять разряд δ — известно, что результирующий набор находится в поддереве $[T']$.

Этап 2. Наконец, рассмотрим случай, когда результирующий набор находится в цикле.

Пусть оператор A_{13} по основной части кода, содержащей исходный набор, в случае, если исходный набор содержится либо в цикле, либо в дереве 1-го или 2-го размера, выделяет кусок кода, содержащий исходный набор. Этот оператор работает аналогично оператору DKU, но по исходному набору $\tilde{\sigma}$ выделяется весь кусок кода (с использованием 2-го дополнительного столбца). Очевидно, что

$$A_{13} \in \mathfrak{F}.$$

Пусть оператор A_{14} по основной части кода образует дополнительную матрицу K^0 размера $Q_4 \times m$, указывающую для каждого набора из цикла, содержащегося в основной части кода, его номер в общей нумерации всех наборов из $\mathfrak{M}$ (если соответствующая строка не содержит набор из цикла, то в матрице K^0 помещается набор из нулей). Это вычисление осуществляет система Q_4 штук m-разрядных сумматоров, последовательно прибавляющих к набору, указывающему номер начального набора куска кода, единицы из 1-го дополнительного столбца. Очевидно, что

$$L(A_{14}) \leqslant C_{42} Q_4 m = o(J).$$

В полной схеме используется 8 штук схем для A_{14} (для каждой основной части кода, выдаваемой схемой для A_9).

Пусть набор $\tilde{\sigma}$ принадлежит циклу или дереву 1-го или 2-го размера. Пусть оператор A_{15} по основной части кода, содержащей исходный набор $\tilde{\sigma}$, матрице K^0 (вычисленной оператором A_{14}), набору $\tilde{\sigma}$ и набору $\tilde{\sigma}^*$ — корню дерева, содержащего набор $\tilde{\sigma}$, вычисляет номера наборов $\tilde{\sigma}$ и $\tilde{\sigma}^*$ в общей нумерации наборов из $\mathfrak{M}$.

Лемма 12. $L(A_{15}) \leqslant C_{43} Q_4 n$.

Доказательство. Наборы $\tilde{\sigma}$ и $\tilde{\sigma}^*$ сравниваются (на совпадение) с каждым набором в основной части кода, являющимся набором из $\mathfrak{M}$ (с использованием 1-го дополнительного столбца) и берутся соответствующие строки в матрице K^0.

Пусть (в случае, если исходный набор $\tilde{\sigma}$ принадлежит дереву 4-го размера) оператор A_{16} по номеру подмножества $\tilde{\nu}$ определяет величину $|\tilde{\tau}^*|$, $|\tilde{\tau}^*| = \varrho(\tilde{\sigma}^*, r(\tilde{\sigma}))$. Очевидно, что

$$A_{16} \in \mathfrak{F}.$$

Пусть исходный набор $\tilde{\sigma}$ принадлежит дереву 3-го или 4-го размера. Пусть оператор A_{17} по наборам $\tilde{\varepsilon}$, $\tilde{\varepsilon}'$ и $\tilde{\nu}$ вычисляет номер набора $r(\tilde{\sigma})$ в общей нумерации наборов в $\mathfrak{M}$. Очевидно, что

$$A_{17} \in \mathfrak{F}.$$

Пусть оператор A_{18} по восьми основным частям кода, восьми соответствующим матрицам K^0, наборам $\tilde{\varepsilon}$, $\tilde{\varepsilon}'$, $\tilde{\nu}$, $\tilde{\sigma}$, $r(\tilde{\sigma})$, $\tilde{\nu}'$ и (в случае длинного цикла) набору $\tilde{\lambda}$ вычисляет результирующий набор.

Лемма 13. $L(A_{18}) \leqslant C_{44} Q_4 n$.

Доказательство. Если набор $[\tilde{\sigma}]$ принадлежит короткому циклу (информация об этом уже есть), то по числам $k-1$, k, $k+1$ выделяются из основных частей кода соответствующие куски. Затем в этих кусках определяются номера первого и последнего наборов (с использованием 2-го дополнительного столбца) и по ним — длина $|\tilde{\lambda}|$ цикла.

Пусть набор $\tilde{\sigma}$ принадлежит циклу, p — номер набора и s — остаток от деления числа $|\tilde{\tau}|$ на $|\tilde{\lambda}|$. Тогда номер p_ω результирующего набора равен $p + s (\mathrm{mod}\, |\tilde{\lambda}|)$, т. е.

$$p_\omega = \begin{cases} p+s, & \text{если } p \leqslant i'', \\ p+s-|\tilde{\lambda}|, & \text{если } p > i''. \end{cases}$$

Если набор $\tilde{\sigma}$ принадлежит дереву, то вычисления аналогичны, только вместо $\tilde{\sigma}$ следует использовать $r(\tilde{\sigma})$, а вместо $\tilde{\tau}$ использовать $\tilde{\tau}'$ в случае дерева 1-го, 2-го или 3-го размера и $\tilde{\tau}' - \tilde{\tau}^*$ (величина $\tilde{\tau}' = \tilde{\tau} - \tilde{\mu}$ уже вычислена оператором DK) в случае дерева 4-го размера.

В случае длинного цикла вычисления аналогичны, но длина цикла уже вычислена оператором A_7.

Наконец, имея номер результирующего набора и используя основные части кода и соответствующие матрицы K^0, можно определить результирующий набор (со сложностью порядка $Q_4 n$).

Таким образом, верхняя оценка в (1) доказана.

Доказательство нижней оценки

Рассмотрим специальные отображения — в виде циклов (длины M). Число таких отображений для одного множества равно $(M-1)!$. Для числа R таких отображений для всех множеств имеем

$$R = \binom{2^n}{M}(M-1)! = \frac{2^n(2^n-1)\cdot\ldots\cdot(2^n-M+1)}{M!}(M-1)! = \frac{2^n(2^n-1)\cdot\ldots\cdot(2^n-M+1)}{M}.$$

Одна схема может реализовать, вообще говоря, несколько таких отображений, но циклы, реализуемые одной схемой, не могут пересекаться. Поэтому число отображений, реализуемых одной схемой, не больше $\frac{2^n}{M}$, и число схем, реализующих такие отображения, не меньше, чем $R \Big/ \frac{2^n}{M}$.

Имеем

$$R \Big/ \frac{2^n}{M} = (2^n-1)\cdot\ldots\cdot(2^n-M+1).$$

Обозначим последнюю величину через Q. Покажем, что

$$\log Q \gtrsim Mn. \tag{6}$$

(при $n \to \infty$, $M \to \infty$).

Рассмотрим два случая.

Случай 1: $M \leqslant 2^n\left(1-\frac{1}{n}\right)$. В этом случае $2^n - M \geqslant \frac{2^n}{n}$ и

$$\log Q \geqslant (M-1)\log(2^n-M) \geqslant (M-1)\log(2^n/n) = (M-1)(n-\log n) \gtrsim Mn.$$

Случай 2: $M > 2^n\left(1-\frac{1}{n}\right)$. В этом случае $M \sim 2^n$ и

$$Q = \frac{(2^n-1)!}{(2^n-M)!} > \frac{(2^n-1)!}{(2^n-M)^{2^n-M}} > \frac{(2^n-1)!}{(2^n/n)^{(2^n/n)}} > \frac{(2^n-1)!}{2^{2^n}}.$$

Далее, поскольку $A! > (A/8)^A$, то

$$\log Q > \log(2^n!) - n - 2^n > 2^n\log(2^n/8) - n - 2^n = \\ = 2^n(n-3) - n - 2^n = 2^n(n-4) - n \sim Mn.$$

Из (6), поскольку число схем с n входами и выходами и h элементами не превосходит $(Ch)^{h+n}$, получается нижняя оценка сложности для рассматриваемых («циклических») отображений и тем самым и нижняя оценка в теореме.

Таким образом, теорема полностью доказана.

Добавление 1

Здесь приводится доказательство некоторого обобщения результата Д. Улига [8, 10] о возможности вычисления значения булевой функции одновременно на нескольких произвольных наборах (посредством схемы из функциональных элементов).

Результат Д. Улига состоит в следующем. Пусть t — натуральное число и $f(x_1, \ldots, x_n)$ — булева функция. Рассмотрим (nt, t)-функцию

$$F^{(t)}(x_{11}, \ldots, x_{1n}, x_{21}, \ldots, x_{2n}, \ \ldots\ , x_{t1}, \ldots, x_{tn}) = \\ = (f(x_{11}, \ldots, x_{1n}), f(x_{21}, \ldots, x_{2n}), \ \ldots\ , f(x_{t1}, \ldots, x_{tn})).$$

Эта (nt, t)-функция вычисляет значение функции $f(x_1, \ldots, x_n)$ одновременно на t (произвольных) наборах, т. е.

$$F^{(t)}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_t) = (f(\tilde{\sigma}_1), \ldots, f(\tilde{\sigma}_t)).$$

Пусть $L^t(f)$ — сложность реализации функции $F^{(t)}$ и пусть $L^t(n) = \max L^t(f(x_1, \ldots, x_n))$ (максимум берется по всем функциям от n переменных). Д. Улиг доказал следующее утверждение.

Теорема. *Пусть*

$$t = 2^{o\left(\frac{n}{\log n}\right)}. \tag{Д.1}$$

Тогда

$$L^t(n) \sim \frac{2^n}{n}.$$

Эта теорема утверждает, что при условии (Д.1) выполнено соотношение $L^t(n) \sim L(n)$. Другими словами, сложность одновременной реализации функции на произвольных t наборах (при условии (Д.1)) асимптотически совпадает со сложностью ее реализации на одном наборе (в смысле функции Шеннона). Доказательство Д. Улига почти дословно переносится на случай реализации системы m функций, заданных на «первых» r наборах.

Определим функцию $L^t(r, m)$, аналогичную функции $L^t(n)$. Пусть

$$F(x_1, \ldots, x_n) = (f_1(x_1, \ldots, x_n), f_2(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$$

— некоторая (n, m)-функция и

$$\begin{aligned} F^t(x_{11}, \ldots, x_{1n}, x_{21}, \ldots, x_{2n}, \ \ldots\ , x_{t1}, \ldots, x_{tn}) = \\ = (F(x_{11}, \ldots, x_{1n}), F(x_{21}, \ldots, x_{2n}), \ \ldots\ , F(x_{t1}, \ldots, x_{tn})) = \\ = ((f_1(x_{11}, \ldots, x_{1n}), f_2(x_{11}, \ldots, x_{1n}), \ \ldots\ , f_m(x_{11}, \ldots, x_{1n})), \\ (f_1(x_{21}, \ldots, x_{2n}), f_2(x_{21}, \ldots, x_{2n}), \ \ldots\ , f_m(x_{21}, \ldots, x_{2n})), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ (f_1(x_{t1}, \ldots, x_{tn}), f_2(x_{t1}, \ldots, x_{tn}), \ \ldots\ , f_m(x_{t1}, \ldots, x_{tn}))), \end{aligned}$$

$L^t(F)$ — сложность реализации функции $F^{(t)}$ и

$$L^t(r, m) = \max_{F \in \mathfrak{F}^{r, m}} L^t(F)$$

(определение $\mathfrak{F}^{r, m}$ было дано выше).

Справедливо следующее утверждение.

Теорема Д1. *Пусть выполнены условия*

$$2^{n-1} < r \leqslant 2^n, \tag{Д.2}$$

$$m \leqslant 2^{2^{Cn}}, \quad C < 1,$$

$$t = 2^{o\left(\frac{n}{\log n}\right)}. \tag{Д.3}$$

Тогда

$$L^t(r, m) \sim \frac{rm}{\log(rm)}. \tag{Д.4}$$

Здесь будет дано доказательство верхней оценки (очевидно, что $L^t(r, m) \geqslant L(r, m)$).

Доказательство может быть проведено почти полным повторением доказательства Д. Улига из [11] (с заменой асимптотически наилучшего метода синтеза схем для одной функции на соответствующий метод для функций

из $\mathfrak{F}^{n,m}$). Однако ввиду фактической недоступности работы [10] это доказательство здесь будет приведено.

Очевидно, что теорему достаточно доказать для случая, когда $t = 2^T$ (степень двойки),

$$T = o\left(\frac{n}{\log n}\right). \tag{Д.5}$$

Выберем некоторое число d, $d = d(n)$, и каждой функции $f(x_1, \ldots, x_n)$ поставим в соответствие систему ее подфункций от первых $n-d$ переменных

$$f_i(x_1, \ldots, x_{n-d}) = f(x_1, \ldots, x_{n-d}, \alpha_1, \ldots, \alpha_d),$$

где $|(\alpha_1, \ldots, \alpha_d)| = i$.

Определим теперь функции $g_i^{(f)}(x_1, \ldots, x_{n-d})$, $i = 0, \ldots, 2^d$:

$$g_i^{(f)} = \begin{cases} f_0, & \text{если } i = 0, \\ f_{i-1} \oplus f_i, & \text{если } 0 < i \leqslant 2^d - 1, \\ f_{2^d-1}, & \text{если } i = 2^d. \end{cases}$$

Систему функций $G^{(f)} = (g_0^{(f)}, \ldots, g_{2^d}^{(f)})$ назовем *системой, соответствующей функции* f.

Из определения функций $g_i^{(f)}$ следует, что, с одной стороны,

$$f_i = g_0^{(f)} \oplus g_1^{(f)} \oplus \ldots \oplus g_i^{(f)} \tag{Д.6}$$

и, с другой стороны,

$$f_i = g_{i+1}^{(f)} \oplus \ldots \oplus g_{2^d}^{(f)}. \tag{Д.7}$$

Это обстоятельство позволяет вычислять значение функции f на любых двух наборах $\tilde{\alpha}$ и $\tilde{\beta}$, используя не пересекающиеся подмножества функций системы $G^{(f)}$. В этом и состоит основная идея конструкции Д. Улига.

Для любого набора $\tilde{\sigma}$ (любой длины) будем обозначать через $\tilde{\sigma}^+$ его правую часть длины d и через $\tilde{\sigma}^-$ — оставшуюся левую часть (т. е. $\tilde{\sigma} = (\tilde{\sigma}^-, \tilde{\sigma}^+)$). Пусть $\tilde{\alpha} = (\tilde{\alpha}^-, \tilde{\alpha}^+)$ и $\tilde{\beta} = (\tilde{\beta}^-, \tilde{\beta}^+)$ — два произвольных набора, и пусть $|\tilde{\alpha}^+| = i$, $|\tilde{\beta}^+| = j$. Если $i \leqslant j$, то

$$f(\tilde{\alpha}) = f(\tilde{\alpha}^-, \tilde{\alpha}^+) = f_i(\tilde{\alpha}^-) = (\text{в силу (Д.6)}) = g_0^{(f)}(\tilde{\alpha}^-) \oplus g_1^{(f)}(\tilde{\alpha}^-) \oplus \ldots \oplus g_i^{(f)}(\tilde{\alpha}^-)$$

$$f(\tilde{\beta}) = f(\tilde{\beta}^-, \tilde{\beta}^+) = f_j(\tilde{\beta}^-) = (\text{в силу (Д.7)}) = g_{j+1}^{(f)}(\tilde{\beta}^-) \oplus g_{j+2}^{(f)}(\tilde{\beta}^-) \oplus \ldots \oplus g_{2^d}^{(f)}(\tilde{\beta}^-).$$

При этом наборы $\tilde{\alpha}^-$ и $\tilde{\beta}^-$ подставляются в различные функции $g_h^{(f)}$. Для осуществления этих операций будут использованы специальные схемы управления.

Схема $A_{n,d}$. Эта схема работает с двумя наборами длины n и готовит информацию для $2^d + 1$ схем (или $2^d + 1$ комплектов схем) для функций системы $G^{(f)}$.

Сначала у двух наборов длины n сравниваются их правые части длины d, т. е. вычисляется функция $S_d(\tilde{x}^+, \tilde{y}^+)$:

$$S_d(\tilde{\alpha}^+, \tilde{\beta}^+) = \begin{cases} 0, & \text{если } |\tilde{\alpha}^+| \leqslant |\tilde{\beta}^+|, \\ 1, & \text{если } |\tilde{\alpha}^+| > |\tilde{\beta}^+| \end{cases}$$

(«сигнал σ»). Очевидно, что

$$L(S_d) \leqslant C_1' d.$$

После этого наборы $(\widetilde{\alpha}^-, \widetilde{\alpha}^+)$ и $(\widetilde{\beta}^-, \widetilde{\beta}^+)$ «остаются на своих местах» (если $\sigma = 0$) или «меняются местами» (если $\sigma = 1$). Это осуществляет схема D_n. Очевидно, что

$$L(D_n) \leqslant C_2' n.$$

Ясно, что для двух образовавшихся после этого наборов $\widetilde{\alpha}'$ и $\widetilde{\beta}'$ длины n выполнено условие $|(\widetilde{\alpha}')^+| \leqslant |(\widetilde{\beta}')^+|$, т. е. $i \leqslant j$.

Затем работает схема $H_{n,d}$. Она получает на входах наборы $\widetilde{\alpha}'$ и $\widetilde{\beta}'$ и подает на входы схем для функций $g_0^{(f)}, g_1^{(f)}, \ldots, g_{2^d}^{(f)}$ требуемые наборы. Именно, на входы схем для функций $g_0^{(f)}, \ldots, g_i^{(f)}$ — набор $(\widetilde{\alpha}')^-$, а на входы схем для функций $g_{j+1}^{(f)}, \ldots, g_{2^d}^{(f)}$ — набор $(\widetilde{\beta}')^-$. Это может быть осуществлено следующим образом.

1) Сначала по наборам $(\widetilde{\alpha}')^+$ и $(\widetilde{\beta}')^+$ образуются наборы $\widetilde{\varepsilon}_\alpha$ и $\widetilde{\varepsilon}_\beta$ длины $2^d + 1$, имеющие по одной единице в разрядах с номерами i и j соответственно (остальные разряды — нулевые):

$$\widetilde{\varepsilon}_\alpha = (0, \ldots, 0, 1_i, 0, 0, \ldots, 0),$$
$$\widetilde{\varepsilon}_\beta = (0, 0, \ldots, 0, 1_j, 0, \ldots, 0).$$

Это — системы всех конъюнкций d переменных или их отрицаний, расположенных в лексикографическом порядке. Поэтому наборы $\widetilde{\varepsilon}_\alpha$ и $\widetilde{\varepsilon}_\beta$ реализуются со сложностью порядка 2^d.

2) Затем по набору $\widetilde{\varepsilon}_\alpha$ образуется набор $\widetilde{\eta}$ длины $2^d + 1$, имеющий единицы в разрядах с номерами $0, 1, \ldots, i$ (остальные разряды — нулевые):

$$\eta_{2^d+1} = 0, \qquad \eta_h = \eta_{h+1} \vee (\varepsilon_\alpha)_h \quad (h \leqslant 2^d),$$
$$\widetilde{\eta} = (\underbrace{1, 1, \ldots, 1}_{i+1}, 0, \ldots, 0).$$

По набору $\widetilde{\varepsilon}_\beta$ образуется набор $\widetilde{\zeta}$ длины $2^d + 1$, имеющий единицы в разрядах с номерами $j+1, \ldots, 2^d+1$ (остальные разряды — нулевые):

$$\zeta_0 = 0, \qquad \zeta_h = \zeta_{h-1} \vee (\varepsilon_\beta)_{h-1} \quad (h \geqslant 1),$$
$$\widetilde{\zeta} = (\underbrace{0, 0, \ldots, 0}_{j+1}, 1, \ldots, 1).$$

3) Нам будет удобно использовать следующие операции над наборами:

— для наборов одинаковой длины $\widetilde{\sigma} = (\sigma_1, \ldots, \sigma_k)$, $\widetilde{\tau} = (\tau_1, \ldots, \tau_k)$

$$\widetilde{\sigma} \vee \widetilde{\tau} = (\sigma_1 \vee \tau_1, \ldots, \sigma_k \vee \tau_k)$$

(поразрядная дизъюнкция);

— для наборов, вообще говоря, разной длины $\widetilde{\sigma} = (\sigma_1, \ldots, \sigma_k)$ и $\widetilde{\tau} = (\tau_1, \ldots, \tau_l)$ образуется упорядоченная система из l наборов длины k

$$\widetilde{\sigma} \times \widetilde{\tau} = ((\sigma_1\tau_1, \sigma_2\tau_1, \ldots, \sigma_k\tau_1), (\sigma_1\tau_2, \sigma_2\tau_2, \ldots, \sigma_k\tau_2), \ldots, (\sigma_1\tau_l, \sigma_2\tau_l, \ldots, \sigma_k\tau_l)).$$

Последняя часть схемы осуществляет следующую операцию:

$$((\widetilde{\alpha}')^- \times \widetilde{\eta}) \vee ((\widetilde{\beta}')^- \times \widetilde{\zeta}).$$

В результате образуется система наборов

$$\underbrace{\underbrace{(\tilde{\alpha}')^-, \ldots, (\tilde{\alpha}')^-}_{i+1}, \tilde{0}, \ldots, \tilde{0}}_{j+1}, (\tilde{\beta}')^-, \ldots, (\tilde{\beta}')^-. \qquad (Д.8)$$

Очевидно, что

$$L(H_{n,d}) \leqslant C_3' nd.$$

и

$$L(A_{n,d}) = L(S_d) + L(D_n) + L(H_{n,d}) \leqslant C_4' n 2^d.$$

Наборы из (Д.8) подаются на схемы для функций $g_0^{(f)}, \ldots, g_{2^d}^{(f)}$. На выходах этих схем образуются значения

$$g_0^{(f)}((\tilde{\alpha}')^-), g_1^{(f)}((\tilde{\alpha}')^-), \ldots, g_i^{(f)}((\tilde{\alpha}')^-), \\ g_{i+1}^{(f)}(\tilde{0}), \ldots, g_j^{(f)}(\tilde{0}), g_{j+1}^{(f)}((\tilde{\beta}')^-), \ldots, g_{2^d}^{(f)}((\tilde{\beta}')^-). \qquad (Д.9)$$

Затем работает схема управления $B_{n,d}$.

Схема $B_{n,d}$. Эта схема получает набор (Д.9), наборы $\tilde{\varepsilon}$ и $\tilde{\delta}$ и сигнал σ. Сначала наборы из (Д.9) «умножаются» на компоненты наборов $\tilde{\varepsilon}$ и $\tilde{\delta}$ — образуются соответственно наборы

$$f_\alpha = (g_0^{(f)}((\tilde{\alpha}')^-), g_1^{(f)}((\tilde{\alpha}')^-), \ldots, g_i^{(f)}((\tilde{\alpha}')^-), 0, 0, \ldots, 0)$$

и

$$f_\beta = (0, 0, \ldots, 0, g_{j+1}^{(f)}((\tilde{\beta}')^-), g_{j+2}^{(f)}((\tilde{\beta}')^-), \ldots, g_{2^d}^{(f)}((\tilde{\beta}')^-)).$$

Затем образуется сумма (по $\mathrm{mod}\,2$) всех разрядов набора f_α — это и есть $f(\tilde{\alpha}')$, и сумма (по $\mathrm{mod}\,2$) всех разрядов набора f_β — это и есть $f(\tilde{\beta}')$. Наконец, последняя подсхема вычисляет пару $(f(\tilde{\alpha}), f(\tilde{\beta}))$. Это $(f(\tilde{\alpha}'), f(\tilde{\beta}'))$, если $\delta = 0$, и $(f(\tilde{\beta}'), f(\tilde{\alpha}'))$, если $\delta = 1$.

Окончательно для схемы $B_{n,d}$ имеем

$$L(B_{n,d}) \leqslant C_5' 2^d.$$

Опишем теперь конструкцию схемы для $F^{(t)}$ «в целом». Схема будет состоять из подсхем, реализующих подфункции вектор-функции F, и управляющих подсхем. Схема строится посредством итеративной процедуры.

Обозначим через $S(F, T, m, r, d)$ схему для функции F из $\mathfrak{F}^{r,m}$ после T-го шага (с параметром d), т. е. схему, осуществляющую вычисление функции F из $\mathfrak{F}^{r,m}$ на произвольных 2^T наборах значений переменных. Пусть $s(T, m, r, d)$ — максимум сложностей таких схем (максимум берется по всем функциям из $\mathfrak{F}^{r,m}$ при фиксированных T, m, r, d).

Будем доказывать индукцией по T неравенство

$$s(T, m, r, d) \leqslant (2^d + 1)^T \left(L\left(\left\lceil \frac{r}{2^{Td}} \right\rceil, m \right) + K 2^T (n + m) \right), \qquad (Д.10)$$

где $K = \max(C_4', C_5')$.

$T = 1$. Пусть F — произвольная (n, m)-функция из $\mathfrak{F}^{r,m}$. Рассмотрим ее подфункции, получающиеся при подстановке констант вместо правых

d разрядов (старшими разрядами считаем левые). При этом образуются $(n-d, m)$-функции из $\mathfrak{F}^{r_1, m}$, где $r_1 = \left\lceil \frac{r}{2^d} \right\rceil$. Очевидно, что соответствующие функции $y_i^{(F)}$ принадлежат $\mathfrak{F}^{r_1, m}$; ясно, что выполняется условие $2^{n-d-1} < r_1 \leqslant 2^{n-d}$. Схема $S(1, m, r, d)$ состоит из одной управляющей схемы $A_{n,d}$, 2^d+1 схем для $(n-d, m)$-функций из $\mathfrak{F}^{r_1, m}$ и m схем управления $B_{n,d}$ (для каждой из m компонент реализуемой функции — свой экземпляр схемы $B_{n,d}$). Поэтому

$$\begin{aligned} s(1, m, r, d) &\leqslant L(A_{n,d}) + mL(B_{n,d}) + (2^d+1)L(r_1, m) \leqslant \\ &\leqslant C_4' n 2^d + C_5' m 2^d + (2^d+1)L\left(\left\lceil \frac{r}{2^d} \right\rceil, m\right) \leqslant \\ &\leqslant (2^d+1)\left(L\left(\left\lceil \frac{r}{2^d} \right\rceil, m\right) + K(n+m)\right). \end{aligned}$$

Индуктивный переход $(T \to T+1)$. Входные наборы (2^{T+1} штук) разбиваются на пары. Для каждой пары берется своя схема $A_{n,d}$ (2^T таких схем). Для (n, m)-функции F из $\mathfrak{F}^{r, m}$ образуется система $(n-d, m)$-функций $G^{(F)}$ $(F_0, F_1, \ldots, F_{2^d})$; для функции F_i $(0 \leqslant i \leqslant 2^d)$ берется схема S_i, вычисляющая значения соответствующей $(n-d, m)$-функции на 2^T наборах. В силу индуктивного предположения эта схема может быть построена так, что

$$\begin{aligned} L(S_i) \leqslant s(T, m, r, d) &\leqslant (2^d+1)^T\left(L\left(\left\lceil \frac{r}{2^{Td}} \right\rceil, m\right) + KT(n-d+m)\right) \leqslant \\ &\leqslant (2^d+1)^T\left(L\left(\left\lceil \frac{r}{2^{Td}} \right\rceil, m\right) + KT(n+m)\right). \end{aligned}$$

Далее, j-я группа выходов i-й схемы $A_{n,d}$ подается на i-ю группу входов схемы S_j $(0 \leqslant j \leqslant 2^d,\ 1 \leqslant i \leqslant 2^T)$. Каждая схема S_j имеет 2^T систем выходов, по m выходов в каждой системе. Наконец, берутся $m2^T$ штук схем $B_{n,d}$ (каждая из них имеет 2^d+1 выходов); занумеруем эти схемы парами чиссл (h, i) $(1 \leqslant h \leqslant m,\ 1 \leqslant i \leqslant 2^T)$. Присоединим h-й выход i-й системы схемы S_j к j-му входу схемы $B_{n,d}$ с номером (h, i). Полученная схема будет иметь 2^{T+1} систем выходов, по m выходов в каждой. Из построения схемы следует, что она реализует функцию F на 2^{T+1} наборах. Таким образом, для всей схемы S имеем

$$\begin{aligned} L(S) &\leqslant \sum_{i=0}^{2^d} L(S_i) + 2^T L(A_{n,d}) + 2^T mL(B_{n,d}) \leqslant \\ &\leqslant (2^d+1)s(T, m, r_1, d) + C_4' 2^T n 2^d + C_5' 2^T m 2^d \leqslant \\ &\leqslant (2^d+1)(2^d+1)^T\left(L\left(\left\lceil \frac{r_1}{2^{Td}} \right\rceil, m\right) + KT(n+m)\right) + K2^{T+d}(n+m) \leqslant \\ &\leqslant (2^d+1)^{T+1}\left(L\left(\left\lceil \frac{r}{2^{(T+1)d}} \right\rceil, m\right) + KT(n+m) + \frac{K(n+m)2^{T+d}}{(2^d+1)^{T+1}}\right) \leqslant \\ &\leqslant (2^d+1)^{T+1}\left(L\left(\left\lceil \frac{r}{2^{(T+1)d}} \right\rceil, m\right) + KT(n+m) + K(n+m)\right) \leqslant \\ &\leqslant (2^d+1)^{T+1}\left(L\left(\left\lceil \frac{r}{2^{(T+1)d}} \right\rceil, m\right) + K(T+1)(n+m)\right). \end{aligned}$$

Тем самым неравенство (Д.10) доказано. Положим теперь $d = \lfloor \log n \rfloor$. Тогда $dT = o(n)$. Поэтому $\frac{r}{2^{Td}} \to \infty$ и $\left\lceil \frac{r}{2^{Td}} \right\rceil \sim \frac{r}{2^{dT}}$. Далее, $\frac{T}{2^d} \asymp \frac{T}{n} \to 0$; поэтому

$$(2^d+1)^T \sim 2^{dT}. \qquad (Д.11)$$

Используя теорему (*) (см. сноску на стр. 189), получаем

$$L\left(\left\lceil \frac{r}{2^{dT}} \right\rceil, m\right) \sim \frac{rm}{2^{dT}\log(rm)}.$$

Легко проверить, используя (Д.5), что $T(n+m) = o\left(L\left(\left\lceil \frac{r}{2^{dT}} \right\rceil, m\right)\right)$. Отсюда, используя (Д.11), получаем из (Д.10)

$$(2^d+1)^T\left(L\left(\left\lceil \frac{r}{2^{dT}} \right\rceil, m\right) + K2^T(n+m)\right) \sim \frac{rm}{\log(rm)}.$$

Тем самым теорема Д1 доказана.

В случае, если условие (Д.2) теоремы не выполняется, для сохранения соотношения (Д.4) приходится накладывать более сильные ограничения на порядок роста величины t, чем условие (Д.3).

Справедливо следующее утверждение.

Т е о р е м а. *Пусть выполнены условия*

$$2^{n-1} < r \leqslant 2^n, \qquad \log m = \frac{2^n}{\varphi}, \quad \varphi \to \infty,$$
$$\log t = \left\lceil \frac{\log \varphi}{\log\log \varphi} \right\rceil. \tag{Д.12}$$

Тогда

$$L'(r, m) \sim \frac{rm}{\log(rm)}.$$

Условие (Д.12) может быть несколько ослаблено.

Добавление 2. W-Разбиения деревьев

Л е м м а Д2. *Для любого натурального числа W любое ориентированное к корню дерево можно разбить на не пересекающиеся по ребрам поддеревья $T_1, T_2, \ldots$, веса которых удовлетворяют условиям*

$$W \leqslant |T_i| \leqslant 3W.$$

Будем называть поддерево дерева *граничным*, если оно имеет с остальной частью дерева лишь одну общую вершину.

Доказательство леммы Д2 основывается на следующей лемме.

Л е м м а Д2′. *В каждом дереве веса более W можно выделить такое граничное поддерево T, что*

$$W \leqslant |T| \leqslant 2W.$$

Д о к а з а т е л ь с т в о. Рассмотрим в дереве путь P, идущий от корня к некоторой концевой вершине (против ориентации ребер), удовлетворяющий следующему условию: в качестве очередной вершины этого пути берется корень поддерева с максимальным весом (среди поддеревьев, от корней которых идут ребра к данной вершине). Будем теперь идти по пути P от концевой вершины к корню до первой вершины γ, являющейся корнем поддерева, вес M которого больше W. Рассмотрим теперь два случая.

1) $M \leqslant 2W$. В этом случае в качестве искомого поддерева можно взять поддерево, «врастающее» в γ.

2) $M > 2W$. Здесь рассмотрим два подслучая.

2а) В предыдущую вершину γ' пути входит дерево T веса $W-1$. Тогда вместе с ребром (γ', γ) дерево T имеет вес W. Это поддерево можно взять в качестве искомого.

2б) В предыдущую вершину γ' пути входит дерево T веса меньше чем $W-1$. Рассмотрим все поддеревья $T_1, T_2, \ldots$, оканчивающиеся ребрами, входящими в γ. Из выбора пути P следует, что для каждого i выполнено соотношение $|T_i| \leqslant |T|+1$; и в этом подслучае $|T| < W-1$. Но $\sum |T_i| = M > 2W$. Поэтому для некоторого k будем иметь

$$W < \sum_{i=1}^{k} |T_i| \leqslant 2W.$$

Объединение поддеревьев $T_1, \ldots, T_k$ образует искомое поддерево.

Доказательство леммы Д2. Будем применять лемму Д2′, удаляя в исходном дереве поддеревья, содержащие M_i ребер, $W \leqslant M_i < 2W$, столько раз, сколько возможно. В результате останется либо пустое дерево (и требуемое разбиение получено), либо дерево, содержащее меньше чем W ребер. Присоединив его к последнему выделенному поддереву, получим поддерево, содержащее меньше чем $3W$ ребер.

Лемма доказана.

Добавление 3. Реализация частичных функций

Пусть $\mathfrak{M}$ — некоторое множество наборов длины n и f — функция, определенная на $\mathfrak{M}$. Обозначим через $\widetilde{L}(f)$ сложность простейшего доопределения функции f. Пусть

$$\widetilde{L}(\mathfrak{M}) = \max \widetilde{L}(f)$$

(максимум берется по всем функциям, определенным на $\mathfrak{M}$),

$$\widetilde{L}(M, n) = \max \widetilde{L}(\mathfrak{M})$$

(максимум берется по всем множествам $\mathfrak{M}$, содержащим M наборов длины n).

Теорема Д3. $\widetilde{L}(M, n) \leqslant C_6' M$ (*т. е. верхняя оценка не зависит от* n).

Доказательство основано на следующем утверждении.

Лемма Д3. *Для схем в базисе* $\{\&, \vee, {}^{-}\}$

$$\widetilde{L}(M, n) \leqslant 5M - 4. \qquad (Д.13)$$

Доказательство леммы проводится индукцией по M. Функция, заданная на одном наборе $\widetilde{\sigma}$ (т. е. при $M=1$), является константой. Пусть среди компонент набора $\widetilde{\sigma}$ есть и нули, и единицы (например, $\sigma_i = 0$, $\sigma_j = 1$). Тогда если $f=0$, то $f = x_i$; если $f=1$, то $f = x_j$; т. е. в этом случае $\widetilde{L}(f) = 0$. Если все компоненты набора $\widetilde{\sigma}$ одинаковы, т. е. либо сплошь нули, либо сплошь единицы, то в одном случае $f = x_i$, а в другом $f = \bar{x}_i$; т. е. в обоих случаях $\widetilde{L}(1, n) \leqslant 1$. Таким образом, при $M=1$ имеем $L(M, n) = 5M-4$.

Пусть неравенство (Д.13) справедливо для всех чисел, меньших чем M. Докажем его для M. Пусть $f(x_1, \ldots, x_n)$ — произвольная функция, определенная на $\mathfrak{M}$ ($|\mathfrak{M}| = M$). Очевидно, что существует компонента наборов, в которой имеются и 0, и 1 (в противном случае все наборы в $\mathfrak{M}$ были бы одинаковыми). Пусть для определенности это n-я компонента. Пусть $\mathfrak{M}_0$ — подмножество наборов

из $\mathfrak{M}$ вида $(\sigma_1, \ldots, \sigma_{n-1}, 0)$, а $\mathfrak{M}_1$ — подмножество наборов из $\mathfrak{M}$ вида $(\sigma_1, \ldots, \sigma_{n-1}, 1)$. Пусть $|\mathfrak{M}_0| = M_0$, $|\mathfrak{M}_1| = M_1$. Очевидно, что $1 \leqslant M_0 < M$, $1 \leqslant M_1 < M$ и $M_0 + M_1 = M$. Пусть $f_0(x_1, \ldots, x_{n-1}) = f(x_1, \ldots, x_{n-1}, 0)$ и $f_1(x_1, \ldots, x_{n-1}) = f(x_1, \ldots, x_{n-1}, 1)$. Тогда

$$f(x_1, \ldots, x_{n-1}, x_n) = \bar{x}_n f(x_1, \ldots, x_{n-1}, 0) \vee x_n f(x_1, \ldots, x_{n-1}, 1).$$

Поэтому

$$\widetilde{L}(f) \leqslant \widetilde{L}(f_0) + \widetilde{L}(f_1) + 4 \qquad (Д.14)$$

(в правой части (Д.14) участвуют четыре операции) и

$$\widetilde{L}(f) \leqslant \widetilde{L}(M_0, n) + \widetilde{L}(M_1, n) + 4. \qquad (Д.15)$$

В силу индуктивного предположения

$$\widetilde{L}(M_0, n) \leqslant 5M_0 - 4, \qquad \widetilde{L}(M_1, n) \leqslant 5M_1 - 4. \qquad (Д.16)$$

Из (Д.15) и (Д.16) получаем

$$\widetilde{L} \leqslant 5M - 4.$$

Следствие. *Для схем в базисе* $\{\&, \vee, {}^-\}$

$$\widetilde{L}(M) \leqslant 5M.$$

Теорема непосредственно получается из этого следствия.

СПИСОК ЛИТЕРАТУРЫ

1. Андреев А. Е. О сложности реализации частичных булевых функций схемами из функциональных элементов // Дискретная математика. — 1989. — Т. 1, вып. 4. — С. 36–45.
2. Лупанов О. Б. Об одном методе синтеза схем // Изв. вузов. Сер. Радиофизика. — 1958. — Т. 1, № 1. — С. 120–140.
3. Лупанов О. Б. Об одном подходе к синтезу управляющих систем — принципе локального кодирования // Проблемы кибернетики. Вып. 14. — М.: Наука, 1965. — С. 31–110.
4. Лупанов О. Б. Асимптотические оценки сложности управляющих систем. — М.: Изд-во МГУ, 1984.
5. Лупанов О. Б. О сложности реализации степеней булевой (n, n)-функции // Вестник МГУ. Математика. Механика. — 1993. — № 1. — С. 59–67.
6. Мюллер Ф. О реализации одного класса автоматных отображений // Проблемы кибернетики. Вып. 34. — М.: Наука, 1978. — С. 95–108.
7. Нечипорук Э. И. О сложности вентильных схем, реализующих булевские матрицы с неопределенными коэффициентами // Докл. АН СССР. — 1965. — Т. 163, № 1. — С. 40–42.
8. Редькин Н. П. О сложности реализации недоопределенных булевых функций // Автоматика и телемеханика. — 1969. — № 9. — С. 118–122.
9. Улиг Д. О синтезе самокорректирующихся схем с малым числом надежных элементов // Матем. заметки. — 1974. — Т. 15, № 6. — С. 937–944.
10. Шоломов Л. А. О реализации недоопределенных булевых функций схемами из функциональных элементов // Проблемы кибернетики. Вып. 21. — М.: Наука, 1969. — С. 215–226.
11. Uhlig D. Об одновременном вычислении значений функции алгебры логики на нескольких наборах // Сб. докладов III Междунар. Рабочего семинара «Математические вопросы кибернетики» (МВК'89). — Братислава, 1987. — С. 61–67.

Поступило в редакцию 20 III 2003