Д. Ю. Черухин

НИЖНИЕ ОЦЕНКИ СЛОЖНОСТИ БУЛЕВЫХ ФУНКЦИЙ

МОСКВА • 2005

Черухин Д. Ю. Нижние оценки сложности булевых функций. — М.: График Без Границ, 2005. — 72 с.

Брошюра является введением в круг задач, относящихся к нижним оценкам сложности булевых функций. В ней доказаны нелинейные нижние оценки сложности в классических моделях вычислений: формулах, контактных схемах, монотонных схемах из функциональных элементов. За основу был взят материал, который автор обычно рассказывает на просеминаре по дискретной математике для студентов младших курсов мехмата МГУ. Для замкнутости изложения добавлены вводные сведения о булевых функциях и некоторые факты из асимптотической теории сложности. В конце каждого параграфа есть упражнения для самостоятельной работы. Изложение опирается на элементарную математику. Для понимания достаточно знаний в рамках школьной программы.

Для студентов младших курсов и старшеклассников, интересующихся дискретной математикой.

Черухин Дмитрий Юрьевич
Нижние оценки сложности булевых функций

Редактор: Черухина С. Е.
Набор, вёрстка и издание автора

Неформальное некоммерческое издательство «График Без Границ».
Свёрстано с помощью системы TEX. Подписано в печать 16.05.2005.
Отпечатано на принтере. Объём 4,5 п. л. Тираж 10 экз.

Содержание

Глава I. Введение

§ 1. Булевы функции

Булева функция — это функция одной или нескольких переменных, у которой каждая переменная и сама функция могут принимать только значения 0 и 1. Область определения булевой функции конечна, поэтому её можно задать таблицей, явно указав значения на каждом наборе значений переменных. Основными булевыми функциями являются функции И, ИЛИ, НЕ. Функция И также называется *конъюнкцией* и обозначается знаком & или знаком умножения, например $x\&y$ или $x\cdot y$. Функция ИЛИ (*дизъюнкция*) обозначается знаком $\vee$, а функция НЕ (*отрицание*) — знаком $\neg$ перед своим аргументом или чертой над аргументом, например, $\overline{x}$. Мы будем также использовать функции: *импликацию* ($\to$), *эквиваленцию* ($\leftrightarrow$) и *исключающее* ***ИЛИ*** ($\oplus$). Запись $x \to y$ означает: "если x, то y" или "из x следует y"; запись: $x \leftrightarrow y$ — "x равносильно y". Наконец, запись $x \oplus y$ означает: "либо x, либо y, но не оба одновременно". Значения всех перечисленных функций приведены в таблице 1.

Булевы функции играют важную роль в логике: некоторые из них (а именно, перечисленные выше) используются для составления сложных высказываний из простых и называются *логическими связками*; при этом ложь обозначается нулём, а истина — единицей. Например, рассмотрим высказывание:

$$\text{"если идёт дождь, то солнце не светит".} \tag{1}$$

Оно составлено из двух простых: "идёт дождь" и "солнце светит", соединённых связками "если..., то..." (т. е. импликацией) и НЕ. Обозначим высказывание "идёт дождь" буквой A, а высказывание "солнце светит" — буквой B. Тогда сложное высказывание (1) можно записать более формально: "если A, то не B", или совсем формально: $A \to \overline{B}$.

Истинно ли сложное высказывание (1) — зависит только то того, истинны ли высказывания A и B, но не зависит от их смысла.

x	y	$x \mathbin{\&} y$	$x \vee y$	$\overline{x}$	$x \to y$	$x \leftrightarrow y$	$x \oplus y$
0	0	0	0	1	1	1	0
0	1	0	1	1	1	0	1
1	0	0	1	0	0	0	1
1	1	1	1	0	1	1	0

Табл. 1

A	B	$\overline{B}$	$A \to \overline{B}$
0	0	1	1
0	1	0	1
1	0	1	1
1	1	0	0

Табл. 2

Если считать A и B переменными, принимающими значения "истина" и "ложь", то истинность высказывания (1) определяется булевой функцией $A \to \overline{B}$. Построим для этой функции таблицу истинности (таблица 2), воспользовавшись таблицами истинности отрицания и импликации. Посмотрев на таблицу 2, мы сможем определить, в каких случаях высказывание (1) истинно. Оно истинно во всех случаях, кроме одного: $A = B = 1$, другими словами, когда дождь идёт и солнце светит. Эта ситуация возможна, а значит, исходное высказывание не всегда истинно.

Другое важное применение булевы функции имеют в электронной технике. Второе применение тесно связано с первым; можно сказать, что машины "мыслят" на языке человеческой логики, видимо, потому, что этот язык наиболее прост. Рассмотрим такую задачу. Допустим, нам нужно построить прибор, который в каждый момент времени определяет, истинно ли высказывание (1). Допустим также, что у нас уже есть приборы, определяющие идёт ли дождь и светит ли солнце. Каждый из приборов свой ответ выдаёт в виде определённого напряжения на выходе: высокое напряжение соответствует единице, низкое — нулю. Кроме того, пусть у нас есть запас

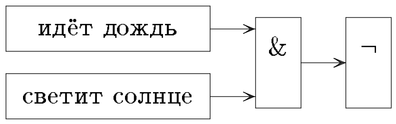

Рис. 1

элементов И, ИЛИ, НЕ и каждый элемент реализует соответствующую функцию, т. е. по комбинации напряжений на своих входах устанавливает соответствующее напряжение на выходе. Тогда наша задача сводится к логической задаче: выразить функцию $A \to \overline{B}$ через функции &, $\vee$ и $\neg$. Данную задачу можно дополнить следующим требованием: выразить функцию так, чтобы для её вычисления было использовано как можно меньшее число элементов (из соображения экономии).

В нашем примере решение можно угадать: заметим, что таблица для функции $A \to \overline{B}$ есть отрицание таблицы функции $A \,\&\, B$, другими словами, $A \to \overline{B} = \overline{A \,\&\, B}$; итоговая схема прибора изображена на рис. 1. Однако, если бы нам понадобилось построить экономную схему для более сложной функции, то пришлось бы перебирать много вариантов. Другого решения, по-видимому, нет: здесь мы вторгаемся в сферу хаоса. Теория сложности может предложить два разных решения этой проблемы: во-первых, разработаны схемы, которые можно использовать (с небольшими изменениями) для широкого класса функций и которые являются почти оптимальными в худшем случае; во-вторых, построены хорошие схемы для естественных (просто определяемых или часто встречающихся) функций. Наиболее актуальная проблема в данной области (скорее математическая, нежели техническая) состоит в доказательстве того, что построенная схема для естественной функции, кажущаяся неупрощаемой, действительно оптимальна или близка к ней. Это и есть проблема нижних оценок сложности.

Упражнение 1. *Докажите, что схема, изображенная на рис. 1, оптимальна (а именно, не существует схемы для данной задачи с меньшим числом элементов* ***И, ИЛИ, НЕ****).*

Упражнение 2. *Выразите функции $\to$, $\leftrightarrow$, $\oplus$ через функции* ***И, ИЛИ, НЕ.***

§ 2. Формулы

Начнём изучение нижних оценок с наиболее простого класса схем, а именно, с формул в базисе $B_0 = \{\&, \vee, \neg\}$. *Базисом* обычно называется множество функций, которые считаются в данной задаче элементарными, т. е. через которые требуется выразить исследуемую функцию. Переменными будем называть буквы, имеющие, возможно, числовые индексы, например, x, y_2. Формулы в базисе B_0 определим по индукции:

а) любая переменная является формулой;

б) если F и G — формулы, то формулами являются выражения $(F \,\&\, G)$, $(F \vee G)$ и $\overline{F}$;

в) каждая формула может быть получена по правилу а) или б).

Пример формулы: $\overline{(\overline{x} \vee (y \,\&\, x))}$. Обозначим эту формулу через F. Построение формулы F по индукции выглядит так: переменные (x, y) являются формулами. Тогда выражения $\overline{x}$ и $y \& x$ — формулы, а значит, их дизьюнкция, т. е. $\overline{x} \vee (y \& x)$ — формула. Наконец, F является отрицанием последней формулы. При записи формул можно использовать следующие сокращения: во-первых, можно опустить скобки, если при этом порядок действий не нарушается; порядок действий по умолчанию таков: вначале выполняется НЕ, затем — И, в последнюю очередь — ИЛИ; во-вторых, знак конъюнкции можно опустить (заметим, что конъюнкция действует как умножение, а знак умножения опускается). Пользуясь этими правилами, формулу F можно записать в виде: $\overline{\overline{x} \vee yx}$.

Каждая формула *задат* некоторую булеву функцию. Нашей ближайшей целью является доказать, что любую булеву функцию можно задать некоторой формулой в базисе B_0. Базис, обладающий данным свойством, называется *полным*.

Утверждение 1 (разложение по переменной). *Для любой булевой функции f от n переменных справедливо тождество:*

$$f(x_1, \ldots, x_n) = \overline{x}_1 f(0, x_2, \ldots, x_n) \vee x_1 f(1, x_2, \ldots, x_n). \qquad (2)$$

Доказательство. Чтобы убедиться в тождестве (2), достаточно рассмотреть два случая: $x_1 = 0$, $x_1 = 1$. Справедливы равенства:

$x \,\&\, 0 = 0$, $x \,\&\, 1 = x$, $x \vee 0 = x$. Используя их, преобразуем при $x_1 = 0$ правую часть выражения (2):

$$\overline{0} \,\&\, f(0, x_2, \ldots, x_n) \vee 0 \,\&\, f(1, x_2, \ldots, x_n) =$$

$$= 1 \,\&\, f(0, x_2, \ldots, x_n) \vee 0 = f(0, x_2, \ldots, x_n),$$

т. е. мы получили левую часть тождества (2). Аналогично, в случае $x_1 = 1$ имеем:

$$\overline{1} \,\&\, f(0, x_2, \ldots, x_n) \vee 1 \,\&\, f(1, x_2, \ldots, x_n) =$$

$$= 0 \,\&\, f(0, \ldots, x_n) \vee f(1, x_2, \ldots, x_n) = f(1, x_2, \ldots, x_n). \ \square$$

Утверждение 2. *Любую булеву функцию можно задать формулой в базисе B_0.*

Доказательство. Пусть n — число аргументов функции f. Проведём индукцию по n.

Базис индукции: $n = 1$. Существует 4 функции от одной переменной: $0, 1, x, \overline{x}$. Выражения x и $\overline{x}$ являются формулами, константы выражаются так: $0 = x \,\&\, \overline{x}$, $1 = x \vee \overline{x}$.

Индуктивный переход: пусть для всех функций от $(n-1)$-й переменной утверждение доказано. Рассмотрим произвольную булеву функцию f от n переменных. Разложим f по первой переменной в соответствии с тождеством (2). Полученные при этом функции $f_0 = f(0, x_2, \ldots, x_n)$ и $f_1 = f(1, x_2, \ldots, x_n)$ зависят от $(n-1)$-й переменной $x_2, \ldots, x_n$. По предположению индукции каждую из этих функций можно выразить формулой над B_0. Пусть F_0, F_1 — соответствующие формулы. Тогда функция f выражается формулой $\overline{x}_1 F_0 \vee x_1 F_1$. $\square$

В качестве примера найдём выражение в базисе B_0 для функции $x \to y$. Из таблицы заметим, что $0 \to y = 1$, $1 \to y = \overline{y}$. Поэтому:

$$x \to y = \overline{x}(0 \to y) \vee x(1 \to y) = \overline{x} \,\&\, 1 \vee x\overline{y} = \overline{x} \vee xy.$$

Упражнение 3. *Докажите, что любую булеву функцию, не равную тождественно нулю, можно представить в виде $K_1 \vee \ldots \vee$*

K_s, где каждое из выражений $K_1, \ldots, K_s$ является конъюнкцией некоторых переменных или их отрицаний (формула вида $K_1 \vee \ldots \vee K_s$ называется дизъюнктивной нормальной формой, сокращенно — ДНФ).

Упражнение 4. *Пусть $B_1 = \{\&, \oplus, 1\}$.*

а) Дайте определение формулы в базисе B_1;

б) докажите, что базис B_1 является полным (указание: придумайте разложение по переменной, имеющее вид: $f = xg \oplus h$, где функции g и h не зависят от x).*

§ 3. Сложность формулы и функции

Сложностью формулы называется число вхождений в неё переменных. Например, сложность формулы $\overline{x} \vee xy$ равна трём. Сложность формулы F будем обозначать через $L(F)$. Так, $L(\overline{x} \vee xy) = 3$. Как мы выяснили ранее, каждую функцию можно выразить некоторой формулой. Одну и ту же функцию f можно выразить, в общем случае, разными формулами. Среди этих формул нас интересуют те, сложность которых минимальна; такие формулы назовём *минимальными* для функции f. Сложность минимальной формулы функции f будем называть сложностью *функции* f и обозначать её через $L_\Phi(f)$. При этом подразумевается, что мы используем базис B_0.

Как мы уже заметили, $x \to y = \overline{x} \vee xy$, т. е. импликацию можно реализовать некоторой (возможно, не минимальной) формулой сложности 3, а значит, $L_\Phi(\to) \leqslant 3$. Покажем, что это неравенство можно улучшить. Воспользуемся тождеством $x \vee yz = (x \vee y)(x \vee z)$ и другими, уже знакомыми тождествами:

$$x \to y = \overline{x} \vee xy = (\overline{x} \vee y)(\overline{x} \vee x) = (\overline{x} \vee y) \cdot 1 = \overline{x} \vee y.$$

Итак, $L_\Phi(\to) \leqslant 2$. Осталось доказать, что оценка 2 неулучшаема, чем мы сейчас и займёмся.

Скажем, что функция f *существенно* зависит от i-й переменной, если найдутся такие два набора значений переменных, отличающиеся только по i-й координате, на которых функция f принимает разные значения. Например, функция $x \to y$ принимает разные значения на наборах $(1, 0)$ и $(0, 0)$. Эти наборы отличаются только по первой коодинате, которая соответствует переменной x. Значит, функция $x \to y$ существенно зависит от x. Аналогично, функция $x \to y$ принимает разные значения на наборах $(1, 0)$ и $(1, 1)$, значит, $x \to y$ существенно зависит от y. Напротив, функция $x \vee \overline{x}$ существенно не зависит от переменной x, так как она тождественно равна константе 1.

Утверждение 3. *Если функция f существенно зависит от переменной x_i, то переменная x_i присутствует в любой формуле, выражающей функцию f.*

Доказательство проведём от противного. Допустим, что некоторая формула F выражает функцию f и не содержит переменную x_i. Найдутся два набора констант $\tilde{\alpha}$ и $\tilde{\beta}$, отличающихся только по i-й координате, такие, что f на них принимает разные значения. Но для формулы F наборы $\tilde{\alpha}$ и $\tilde{\beta}$ совпадают, так как F не содержит переменной x_i, а значит, F принимает на этих наборах одинаковые значения. Противоречие. □

Следствие 1. *Если функция f существенно зависит от k переменных, то $L_\Phi(f) \geqslant k$.*

Действительно, каждая переменная, от которой f существенно зависит, должна входить в минимальную формулу функции f, поэтому сложность минимальной формулы не меньше k. □

Мы знаем, что функция $x \to y$ существенно зависит от двух переменных, поэтому $L_\Phi(\to) \geqslant 2$. Окончательно, $L_\Phi(\to) = 2$.

Упражнение 5. *Убедитесь в справедливости тождеств:* $x(y \vee z) = xy \vee xz$, $x \vee yz = (x \vee y)(x \vee z)$.

Упражнение 6. *Найдите* $L_\Phi(x \to (y \to x))$.

Упражнение 7. *Докажите, что сложность любой функции равна сложности её отрицания.*

Глава II. Линейные функции

§ 4. Верхняя оценка $4n^2$

Линейной функцией n переменных называется функция $x_1 \oplus x_2 \oplus \oplus \ldots \oplus x_n$. Обозначим её через Λ_n. Заметим, что $x \oplus y = x + y \pmod 2$, поэтому $\Lambda_n = x_1 + x_2 + \ldots + x_n \pmod 2$. Другими словами, Λ_n — счётчик чётности суммы n переменных: значение функции равно нулю тогда и только тогда, когда сумма чётна. Нашей ближайшей целью (§§ 4, 5) является построение экономных формул для функции Λ_n и подсчёт их сложности, т. е. получение *верхней* оценки сложности линейной функции.

Сложность функции Λ_n обозначим через λ_n. Отметим, что $\lambda_1 = 1$. Построим формулу для функции $x \oplus y$, используя разложение по переменной, а также равенство $\overline{y} = y \oplus 1$:

$$x \oplus y = \overline{x}(0 \oplus y) \vee x(1 \oplus y) = \overline{x}y \vee x\overline{y}.$$

Таким образом, $\lambda_2 \leqslant 4$. В случае суммы трёх переменных имеем:

$$x \oplus y \oplus z = \overline{x}(y \oplus z) \vee x(\overline{y \oplus z}).$$

Функции $y \oplus z$ и $\overline{y \oplus z}$ имеют равную сложность (см. упражнение 7), поэтому

$$\lambda_3 \leqslant 2(\lambda_2 + 1) \leqslant 2(4 + 1) = 10.$$

В общем случае получим:

$$\Lambda_n = \overline{x}_n \Lambda_{n-1} \vee x_n \overline{\Lambda_{n-1}}. \tag{3}$$

Отсюда следует рекуррентное неравенство:

$$\lambda_n \leqslant 2(\lambda_{n-1} + 1).$$

Найдём явную оценку для λ_n. Прибавим к обеим частям двойку:

$$\lambda_n + 2 \leqslant 2(\lambda_{n-1} + 1) + 2 = 2(\lambda_{n-1} + 2)$$

и выпишем цепочку неравенств:

$$\lambda_n + 2 \leqslant 2(\lambda_{n-1} + 2) \leqslant 4(\lambda_{n-2} + 2) \leqslant \ldots \leqslant 2^{n-1}(\lambda_1 + 2).$$

Окончательно найдём

$$\lambda_n \leqslant 3 \cdot 2^{n-1} - 2.$$

Мы доказали следующую лемму:

Лемма 1. $\lambda_n \leqslant 3 \cdot 2^{n-1} - 2.$

Оказывается, что построенные нами формулы далеки от оптимальных. Даже при $n = 4$ можно построить лучшую формулу. Для этого нужно не отщеплять по одной переменной (как мы делали), а делить множество переменных пополам (или примерно пополам, если число переменных нечётно). Рассмотрим функцию $S = x \oplus y \oplus z \oplus t$. Введём вспомогательные переменные A и B и обозначим $A = x \oplus y$, $B = z \oplus t$. Тогда $S = A \oplus B$. Значит, $S = \overline{A}B \vee A\overline{B}$. Теперь вместо A и B подставим готовые формулы для функций $x \oplus y$ и $z \oplus t$. В итоге получим:

$$S = \overline{A}B \vee A\overline{B} = \overline{(x \oplus y)}(z \oplus t) \vee (x \oplus y)\overline{(z \oplus t)} =$$

$$= \overline{(\overline{x}y \vee x\overline{y})}(\overline{z}t \vee z\overline{t}) \vee (\overline{x}y \vee x\overline{y})\overline{(\overline{z}t \vee z\overline{t})}. \tag{4}$$

Заметим, что лемма 1 даёт оценку $\lambda_4 \leqslant 3 \cdot 2^3 - 2 = 22$, а из формулы (4) следует $\lambda_4 \leqslant 16$. Мы видим, что метод деления пополам более эффективен. Применим этот метод в общем случае. Вначале предположим, что n — степень двойки, $n = 2^k$. Обозначим через $\Pi_{n/2}$ функцию $x_{n/2+1} \oplus \ldots \oplus x_n$. Тогда

$$\Lambda_n = \Lambda_{n/2} \oplus \Pi_{n/2} = \overline{\Lambda_{n/2}}\Pi_{n/2} \vee \Lambda_{n/2}\overline{\Pi_{n/2}}. \tag{5}$$

Функции $\Lambda_{n/2}$, $\Pi_{n/2}$ и их отрицания имеют одну и ту же сложность, поэтому из (5) следует рекуррентное соотношение:

$$\lambda_n \leqslant 4\lambda_{n/2}. \tag{6}$$

Выпишем цепочку рекуррентных соотношений:

$$\lambda_n = \lambda_{2^k} \leqslant 4\lambda_{2^{k-1}} \leqslant 4^2\lambda_{2^{k-2}} \leqslant \ldots \leqslant 4^k\lambda_1 = 4^k = n^2,$$

из которой следует

Лемма 2. *Если $n = 2^k$, то $\lambda_n \leqslant n^2$.*

Теперь построим формулы для произвольного n. Идея такова: рассмотрим ближайшую к n степень двойки (не меньшую n), построим формулу для этой степени двойки и удалим из неё лишние переменные. Возможность удаления переменных требует обоснования. Дело в том, что, подставив в формулу вместо переменной константу, мы получим уже не формулу (в базисе B_0). Такое выражение назовём *псевдоформулой*: псевдоформула — выражение, полученное из формулы заменой некоторых переменных константами (формула является псевдоформулой). Сложностью псевдоформулы назовём число вхождений в неё переменны (константы не считаются). Скажем, что псевдоформулы *эквивалентны*, если они тождественно равны, т. е. их значения совпадают на каждом наборе значений переменных.

Утверждение 4. *Если псевдоформула F не равна тождественно константе, то существует эквивалентная ей формула F', имеющая ту же, или меньшую сложность.*

Доказательство. Проведём индукцию по общему числу символов в выражении F, включая переменные, константы, операции И, ИЛИ, НЕ, скобки.

Базис индукции: F состоит из одного символа. Этим символом может быть переменная или константа. В первом случае F является формулой и можно положить $F' = F$; во втором случае F тождественно равна константе.

Индуктивный переход: для всех формул с меньшим, чем в F, числом символов утверждение доказано. Возможен один из следующих случаев: 1) $F = \overline{G}$; 2) $F = G \mathbin{\&} H$; 3) $F = G \vee H$. В любом случае, если хотя бы одна из псевдоформул G, H тождественно равна константе, то общее число символов в F можно уменьшить, пользуясь тождествами: $\overline{0} = 1$, $\overline{1} = 0$, $0 \mathbin{\&} x = 0$, $1 \mathbin{\&} x = x$, $0 \vee x = x$,

$1 \vee x = 1$. Например, если в случае 2 формула G тождественно равна 1, то можно упростить $F = 1 \mathbin{\&} H = H$. После того, как формула F будет упрощена, применим к ней предположение индукции и докажем утверждение. Осталось рассмотреть возможность, когда и G и H (в случаях 2, 3) тождественно не равны константе. Применив к псевдоформулам G и H предположение индукции, найдём формулы G' и H', тождественно равные исходным и имеющие не большую сложность. Положим в случае 1: $F' = \overline{G'}$, а в случаях 2 и 3, соответственно, $F' = G' \mathbin{\&} H'$ и $F' = G' \vee H'$. Требования утверждения, очевидно, выполнены. $\square$

Вернёмся к задаче оценивания числа λ_n. Пусть 2^k — наименьшая степень двойки такая, что $n \leqslant 2^k$. Тогда $2^{k-1} < n$. По лемме 2 существует формула F для функции Λ_{2^k}, имеющая сложность 2^{2k}. Подставим в формулу F вместо каждой из лишних переменных (т. е. $x_{n+1}, \ldots, x_{2^k}$) константу 0. Получим псевдоформулу F_1, задающую функцию Λ_n. Функция Λ_n тождественно не равна константе, поэтому, согласно утверждению 4, её можно задать формулой такой же, или меньшей сложности. Итак, $\lambda_n \leqslant L(F_1) \leqslant L(F)$. Окончательно получим:

$$\lambda_n \leqslant 2^{2k} = 4 \cdot (2^{k-1})^2 < 4n^2.$$

Нами доказана

Лемма 3. $\lambda_n < 4n^2$.

Упражнение 8. *Докажите, что если $n < m$, то $\lambda_n \leqslant \lambda_m$.*

Упражнение 9. *Существует ли функция двух переменных, более сложная, чем функция $x \oplus y$?*

§ 5. Верхняя оценка $9/8n^2$

В этом параграфе мы уточним верхнюю оценку сложности, извлекаемую из построения формулы методом деления пополам. Заметим, что представление (5) и рекуррентное неравенство (6) верны для любого чётного n. Если же n нечётно, а именно, $n = 2m + 1$, то разобъём переменные на две примерно равные части: $x_1, \ldots, x_m$ и

n	1	2	3	4	5	6	7	8	9	10
$\lambda_n \leqslant$	1	4	10	16	28	40	52	64	88	112
$\lambda_n \geqslant$	1	4	10	16	28	39	52	64	84	102

Табл. 3

$x_{m+1}, \ldots, x_n$. Действуя аналогично случаю чётного n, введём обозначение $\Psi_{m+1} = x_{m+1} \oplus \ldots \oplus x_n$ и выпишем тождество

$$\Lambda_n = \overline{\Lambda_m}\Psi_{m+1} \vee \Lambda_m\overline{\Psi_{m+1}},$$

следствием которого является неравенство

$$\lambda_{2m+1} \leqslant 2(\lambda_m + \lambda_{m+1}). \tag{7}$$

Из соотношений (6) и (7) несложно вывести верхние оценки чисел λ_n для небольших n, например $\lambda_5 \leqslant 2(\lambda_2 + \lambda_3) \leqslant 2(4+10) = 28$. При $n \leqslant 10$ эти оценки выписаны во второй строке таблицы 3. В третьей строке собраны известные нижние оценки для этих чисел; некоторые из них мы докажем в последующих параграфах. Следует отметить, что ни для одного n не предложено формулы (задающей функцию Λ_n) более простой, чем можно построить описанным выше методом деления пополам. В то же время для почти всех n не доказано, что более простых формул не существует. Например, при $n = 6$ точное значение сложности неизвестно; известно только, что $39 \leqslant \lambda_6 \leqslant 40$.

Введём обозначение: p_n — наибольшая степень двойки, не превосходящая n. Например, $p_7 = 4$, $p_8 = 8$. Число p_n можно задать формулой: $p_n = 2^{[\log_2 n]}$. Следующая теорема даёт точное значение сложности формул, которые можно получить методом деления пополам.

Теорема 1. $\lambda_n \leqslant 3np_n - 2p_n^2$.

Доказательство. Проведём полную индукцию по n.

Базис индукции, $n = 1$, очевиден: $\lambda_1 = 1 = 3-2 = 3\cdot 1\cdot p_1 - 2p_1^2$.

Индуктивный переход. Пусть для всех чисел, меньших n, теорема доказана. Рассмотрим два случая: 1) n чётно; 2) n нечётно.

В случае 1) воспользуемся неравенством (6). Пусть $m = \frac{n}{2}$. Для числа m теорема уже доказана, кроме того, $p_{2m} = 2p_m$ (действительно: $2^{[\log_2(2m)]} = 2^{[\log_2 m+1]} = 2^{[\log_2 m]+1} = 2p_m$). Поэтому имеем:

$$\lambda_n \leqslant 4\lambda_m \leqslant 4(3mp_m - 2p_m^2) = 3 \cdot 2m \cdot 2p_m - 2(2p_m)^2 = 3np_n - 3p_n^2.$$

Случай 2). Пусть $n = 2m + 1$. Вначале докажем неравенство

$$\lambda_{m+1} \leqslant 3(m+1)p_m - 2p_m^2. \tag{8}$$

Если $p_{m+1} = p_m$, то (8) следует из предположения индукции для числа $m + 1$. Случай $p_{m+1} \neq p_m$ возможен только, если $m + 1$ — степень двойки. Рассмотрим эту возможность отдельно. Пусть $m+ +1 = 2^k$. Тогда $p_m = 2^{k-1}$. В лемме 2 мы показали, что $\lambda_{m+1} \leqslant 2^{2k}$, поэтому

$$\lambda_{m+1} \leqslant 2^{2k} = 3 \cdot 2^{2k-1} - 2^{2k-1} = 3(m+1)p_m - 2p_m^2.$$

Итак, неравенство (8) доказано. Поскольку n — нечётное число и $n > 1$, то n не является степенью двойки, а значит, $p_n = p_{n-1} = p_{2m} = 2p_m$. Используя неравенства (7), (8) и предположение индукции для числа m, окончательно получим

$$\lambda_n \leqslant 2(\lambda_m + \lambda_{m+1}) \leqslant 2(3mp_m - 2p_m^2 + 3(m+1)p_m - 2p_m^2) =$$

$$= 6(2m+1)p_m - 8p_m^2 = 3n \cdot 2p_m - 2(2p_m)^2 = 3np_n - 2p_n^2. \ \square$$

Следствие 2. $\lambda_n \leqslant 9/8n^2$.

Доказательство. С учётом теоремы 1 достаточно доказать, что

$$3np_n - 2p_n^2 \leqslant 9/8n^2.$$

Перенесём все слагаемые в правую часть:

$$\frac{9}{8}n^2 - 3np_n + 2p_n^2 = \frac{9n^2 - 24np_n + 16p_n^2}{8} = \frac{(3n - 4p_n)^2}{8} \geqslant 0. \ \square$$

Упражнение 10. *Докажите, что из теоремы* 1 *нельзя вывести следствие вида* $\lambda_n \leqslant Cn^2$ *ни для какой константы* C *меньшей, чем* $9/8$.

Упражнение 11. *Пусть функция* f *от переменных* $x_1, \ldots, x_n$ *задана в виде системы, состоящей из* m *линейных уравнений по модулю* 2. *Каждое уравнение имеет вид:*

$$c_1x_1 \oplus c_2x_2 \oplus \ldots \oplus c_nx_n = c_0,$$

где $c_1, \ldots, c_n$, c_0 — *константы* 0 *или* 1. *Функция* f *равна единице тогда и только тогда, когда выполнены все уравнения. Докажите, что* $L_\Phi(f) \leqslant 9/8mn^2$.

§ 6. Нижняя оценка $2n$

Перейдём теперь к изучению нижних оценок сложности, составляющих основное содержание этой брошюры. Простейшую нижнюю оценку можно извлечь из следствия 1 (§3). Функция Λ_n существенно зависит от всех своих переменных, а значит, $\lambda_n \geqslant n$. В данном параграфе мы докажем менее тривиальную (но всё ещё линейную по n) оценку $\lambda_n \geqslant 2n$ при $n \geqslant 2$.

Формулу назовём *приведённой*, если каждое отрицание стоит в ней непосредственно над переменной. Например, формула $\overline{x} \vee xy$ приведена, а формула $\overline{\overline{x} \vee xy}$ нет.

Утверждение 5. *Для любой формулы существует эквивалентная ей приведённая формула, имеющая ту же сложность.*

Доказательство основано на тождествах: $\overline{\overline{x}} = x$, $\overline{x \,\&\, y} = \overline{x} \vee \overline{y}$, $\overline{x \vee y} = \overline{x} \,\&\, \overline{y}$; последние два тождества называются законами *де Моргана*. Чтобы преобразовать произвольную формулу к приведённой, будем последовательно применять эти тождества слева направо, при этом в качестве x и y будем использовать не только переменные, но и более сложные формулы. Например, $\overline{\overline{x} \vee xy} = \overline{\overline{x}} \,\&\, \overline{xy} = x \,\&\, \overline{xy} = x \,\&\, (\overline{x} \vee \overline{y})$. Если на каком-то шаге преобразования ни одно из тождеств не применимо, то все имеющиеся отрицания стоят над переменными, т. е. мы получили приведённую формулу.

Осталось показать, что через конечное число шагов процесс остановится. Для этого достаточно указать некоторую, связанную с формулой, величину, которая принимает натуральные значения и уменьшается при каждом преобразовании. *Длиной* отрицания назовём общее число переменных и операций И, ИЛИ, находящихся под этим отрицанием. Рассмотрим сумму длин всех отрицаний, входящих в формулу. Если мы применяем преобразование $\overline{\overline{x}} = x$, то два отрицания снимаются, а каждое из них имеет длину, большую нуля. При применении законов де Моргана одно отрицание заменяется двумя, но сумма длин этих двух отрицаний на единицу меньше длины исходного отрицания, так как из-под отрицания выпадает связующая операция (& или $\vee$). Итак, при каждом преобразовании сумма длин отрицаний уменьшается. Пусть для начальной формулы эта величина равнялась N, тогда не позднее, чем через N шагов мы получим приведённую формулу. □

Для приведённой формулы можно ввести понятие *литерала*: литерал — это либо переменная, если над ней не стоит отрицание, либо переменная вместе со стоящим над ней отрицанием. Например, в формуле $\overline{x} \vee xy$ три литерала: $\overline{x}$, x и y.

Скажем, что функция f *возрастает* по i-й переменной, если для любых двух наборов значений переменных, отличающихся только по i-й координате, значение функции f на меньшем наборе меньше или равно значению f на большем наборе (бо́льшим считается тот набор, в котором на i-м месте стоит единица). Аналогично, скажем, что f *убывает* по i-й переменной, если для любых двух наборов, отличающихся только по i-й координате, значение функции f на меньшем наборе больше или равно значению на большем наборе. Например, функция $x \& y$ возрастает по переменной x. Действительно, для любых двух наборов $(0, \alpha)$ и $(1, \alpha)$, отличающихся по первой координате (соответствующей переменной x) справедливо неравенство: $0 \,\&\, \alpha = 0 \leqslant 1 \,\&\, \alpha$. В то же время, функция $x \,\&\, y$ не убывает по x потому, что неравенство $0 \,\&\, \alpha \geqslant 1 \,\&\, \alpha$ неверно при $\alpha = 1$.

Утверждение 6. *Пусть приведённая формула F задаёт функцию f. Тогда:*

а) если в формуле F нет литералов $\overline{x}_i$, то функция f возрастает по переменной x_i;

б) если в F нет литералов x_i, то f убывает по x_i.

Доказательство. а) Проведём индукцию по построению формулы F.

Базис индукции: F — литерал. Если F содержит переменную, отличную от x_i, то функция f не зависит существенно от x_i. Поэтому значения функции f на любых двух наборах констант, отличающихся только i-й координатой, равны, а значит, f возрастает (и убывает) по x_i. Если же $F = x_i$, то утверждение очевидно.

Индуктивный переход: $F = G \,\&\, H$ или $F = G \vee H$ (случай $F = \overline{G}$ мы не рассматриваем, так как формула приведена). Пусть формулы G и H задают функции g и h соответственно. Формулы G, H не содержат литералов $\overline{x}_i$, поэтому, по предположению индукции, функции g и h возрастают по x_i. Справедлив следующий общий факт: если функции $\varphi_1, \ldots, \varphi_k$ возрастают по переменной x_i, а функция ψ возрастает по всем переменным, то их композиция, т. е. функция $\psi(\varphi_1, \ldots, \varphi_k)$, возрастает по x_i. Продемонстрируем это в интересующем нас случае $k = 2$.

Пусть $\tilde{\alpha}$ и $\tilde{\beta}$ — произвольные наборы, которые отличаются только по i-й координате и набор $\tilde{\alpha}$ больше. Из возрастания по x_i функций φ_1, φ_2 следует $\varphi_1(\tilde{\alpha}) \geqslant \varphi_1(\tilde{\beta})$ и $\varphi_2(\tilde{\alpha}) \geqslant \varphi_2(\tilde{\beta})$. Тогда справедлива цепочка неравенств:

$$\psi(\varphi_1(\tilde{\alpha}), \varphi_2(\tilde{\alpha})) \geqslant \psi(\varphi_1(\tilde{\beta}), \varphi_2(\tilde{\alpha})) \geqslant \psi(\varphi_1(\tilde{\beta}), \varphi_2(\tilde{\beta}));$$

первое неравенство — в силу возрастания функции ψ по 1-й переменной, второе — в силу возрастания по 2-й переменной. Следовательно, $\psi(\varphi_1, \varphi_2)$ возрастает по x_i.

Заметим наконец, что функции И, ИЛИ возрастают по обеим переменным, а значит, функции gh и $g \vee h$ возрастают по x_i.

б) Сведём утверждение к пункту а). Уберём в формуле F отрицания со всех вхождений переменной x_i. Получим некоторую формулу F', которая задаёт функцию $f' = f(x_1, \ldots, x_{i-1}, \overline{x}_i, x_{i+1}, \ldots, x_n)$ и не содержит литералов x_i. Согласно п. а) функция f' возрастает по x_i. Заметим, что $f = f'(x_1, \ldots, x_{i-1}, \overline{x}_i, x_{i+1}, \ldots, x_n)$. Поэтому, если значение переменной x_i уменьшается, а значения остальных переменных не изменяются, то значение функции $\overline{x}_i$ увеличивается;

следовательно (в силу возрастания f' по i-й переменной), значение функции f увеличивается. Итак, f убывает по x_i. $\square$

Следствие 3. *а) Если функция f не возрастает по переменной x_i, то в любой формуле, задающей функцию f, содержится литерал $\overline{x}_i$;*

б) если функция f не убывает по x_i, то в любой формуле, задающей f, содержится литерал x_i.

Доказательство от противного (для случая а): допустим, что в некоторой формуле F, задающей функцию f, нет литералов $\overline{x}_i$. Тогда, согласно утверждению 6, функция f возрастает по x_i. Противоречие. Случай б) аналогичен. $\square$

Вернёмся к рассмотрению линейной функции. Вначале покажем, что при $n \geqslant 2$ функция Λ_n не возрастает и не убывает ни по одной своей переменной. В силу симметрии достаточно доказать это для первой переменной. Невозрастание следует из примера:

$$1 \oplus 1 \oplus 0 \oplus \ldots \oplus 0 = 0 \ngeqslant 1 = 0 \oplus 1 \oplus 0 \oplus \ldots \oplus 0,$$

а неубывание — из примера:

$$1 \oplus 0 \oplus \ldots \oplus 0 = 1 \nleqslant 0 = 0 \oplus 0 \oplus \ldots \oplus 0.$$

Итак (в силу следствия 3), в любой приведённой формуле, задающей функцию Λ_n, должны присутствовать хотя бы по одному разу литералы $x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n$. В то же время, любую формулу можно преобразовать к приведённой формуле без изменения сложности, а значит, сложность любой формулы, задающей функцию Λ_n, не меньше чем $2n$. Нами доказана

Лемма 4. *Если $n \geqslant 2$, то $\lambda_n \geqslant 2n$.*

Следствие 4. $\lambda_2 = 4$.

Упражнение 12. *Убедитесь, что законы де Моргана верны.*

Упражнение 13. *Найдите сложность функции от n переменных, принимающей значение 1 только в случаях, когда значения всех переменных равны между собой.*

§ 7. Нижняя оценка $n^{3/2}$

В этом параграфе мы докажем ещё одну нижнюю оценку для чисел λ_n, а именно, $\lambda_n \geqslant n^{3/2}$. Новая оценка для маленьких n не намного лучше уже имеющейся, $2n$, а для $n = 2, 3$ даже хуже. Но если увеличивать число n, то функция $n^{3/2}$ станет со временем больше любой линейной функции от n. Такого рода нижние оценки называются *нелинейными*. В теории нижних оценок сложности интерес представляют, в основном, нелинейные оценки.

Прежде, чем изложить общий метод, рассмотрим случай $n = 3$. *Подформулой* назовём часть формулы, полученную на некоторой шаге построения этой формулы. Например, xy и $\overline{x} \vee xy$ — подформулы формулы $\overline{\overline{x} \vee xy}$. Пусть F — минимальная формула, задающая функцию $x \oplus y \oplus z$; в силу утв. 5 можно считать, что F приведена. Мы знаем, что в F обязательно содержатся литералы $x, \overline{x}, y, \overline{y}, z, \overline{z}$. Допустим, что сложность формулы F равна шести. Тогда выписанные литералы встречаются ровно по одному разу. Рассмотрим какой-нибудь литерал; например, x. Обозначим через $\circ$ операцию ($\&$ или $\vee$), которая непосредственно действует на этот литерал, а через G — подформулу (формулы F), с которой данный литерал связан операцией $\circ$. Подформулу G назовём *соседней* с литералом G. Итак, в формуле F имеется подформула $x \circ G$ (или $G \circ x$, но в этом случае G и x можно поменять местами).

Заметим, что G не может состоять только из литерала $\overline{x}$, поскольку в этом случае формулу F можно упростить, заменив подформулу $x \circ \overline{x}$ константой ($x \mathbin{\&} \overline{x} = 1$, $x \vee \overline{x} = 0$) и удалив затем константы согласно утв. 4. Итак, G содержит некоторый литерал, отличный от $\overline{x}$. В формулу $x \circ G$ вместо переменной x можно подставить такую константу, что полученное выражение после упрощения не будет содержать подформулу G. Например, если $\circ = \&$, то возьмём $x = 0$; тогда $xG = 0 \cdot G = 0$. Если же $\circ = \vee$, то возьмём $x = 1$: $x \vee G = 1 \vee G = 1$. Подставим найденную константу вместо x во всю формулу F и произведём упрощения (включая удаление констант). Мы получим некоторую формулу, которая задаёт функцию $y \oplus z$ или $y \oplus z \oplus 1$ и имеет при этом сложность, не большую трёх. Действительно, из формулы F мы удалили два литерала, соответствующих переменной x и ещё один литерал другой переменной, который входил в G. Получили противоречие с тем, что $\lambda_2 = 4$. Противоречие показывает, что допущение $\lambda_3 = 6$ не верно, а значит, $\lambda_3 \geqslant 7$.

Перейдём к рассмотрению общего случая.

Утверждение 7. *Если литерал x_i (или $\overline{x}_i$) входит в минимальную формулу, тождественно не равную константе, то соседняя с ним подформула не содержит переменную x_i.*

Доказательство. Рассмотрим один из четырёх возможных случаев, а именно, когда литерал — x_i, а ближайшая к нему операция — &. Другие случаи аналогичны. Пусть G — соседняя с данным литералом подформула. Предположим, что в G есть переменная x_i. Покажем тогда, что формулу x_iG, а значит, и всю формулу F можно упростить, что противоречит минимальности последней. Действительно, подставим в формулу G вместо переменной x_i константу 1. Полученную псевдоформулу обозначим через G'. Покажем, что верно тождество $x_iG = x_iG'$. Если $x_i = 0$, то $x_iG = 0 = x_iG'$. Если же $x_i = 1$, то $x_iG = G = G' = x_iG'$. Итак, формулу x_iG можно упростить до псевдоформулы x_iG'; следоватлельно и F можно упростить. Заметим наконец, что F не равна тождественно константе, поэтому, после упрощения, константы из неё можно удалить и получить более простую формулу, чем F формулу. □

Продолжим рассмотрение общего случая. Пусть F — минимальная приведённая формула, задающая функцию Λ_n $(n \geqslant 2)$ и x_i — некоторая её переменная. Пусть также x_i входит в F ровно m раз. Для каждого литерала, соответствующего переменной x_i, имеется своя соседняя подформула. По утв. 7, ни одна из этих подформул не содержит переменной x_i, а значит, и не содержит соседней подформулы для другого литерала переменной x_i (так как сам этот литерал находится рядом со своей соседней подформулой). Другими словами, рассматриваемые подформулы не пересекаются (не содержат общих литералов). Кроме того, каждая из них содержит хотя бы один литерал, отличный от x_i и $\overline{x}_i$.

Для каждого литерала вида x_i или $\overline{x}_i$ найдётся некоторая константа, такая, что если её подставить вместо данного литерала, то соседняя с ним подформула пропадёт после упрощения (чтобы это проверить, нужно рассмотреть 4 случая: $x_i \mathbin{\&} G$, $x_i \vee G$, $\overline{x}_i \mathbin{\&} G$, $\overline{x}_i \vee G$). Всего имеется m констант, соответствующих m литералам переменной x_i. Выберем ту из них, которая встречается наибольшее число раз; это число не меньше, чем $[m/2]$. Подставим выбранную константу вместо x_i во всю формулу F. Полученную в результате псевдоформулу упростим, затем удалим из неё константы, и то, что останется, обозначим через F'. Сложность формулы F' уменьшилась, по сравнению с формулой F, как минимум, на $m + [m/2]$: слагаемое m соответствует m удалённым литералам переменной x_i, а слагаемое $[m/2]$ — тем соседним с литералами подформулам, которые удалились при упрощении (ранее мы выяснили, что каждая из них содержит свой литерал, отличный от x_i, $\overline{x}_i$).

Мы пришли к неравенству:

$$L(F') \leqslant L(F) - m - \left[\frac{m}{2}\right] \leqslant L(F) - \frac{3}{2}m.$$

Далее, переменную x_i выберем так, чтобы число m было как можно больше. Из n возможных переменных найдётся такая, которая входит в формулу F не меньше, чем $L(F)/n$ раз, поэтому

$$L(F') \leqslant L(F) - \frac{3}{2}m \geqslant L(F) - \frac{3}{2} \cdot \frac{L(F)}{n} = L(F) \cdot \left(1 - \frac{3}{2n}\right).$$

Наконец, заметим, что $L(F) = \lambda_n$ (в силу минимальности F), а формула F' задаёт одну из функций Λ_{n-1} или $\overline{\Lambda_{n-1}}$ (с точностью до переименования переменных), поэтому $L(F') \geqslant \lambda_{n-1}$. Нами доказана

Лемма 5. *Если $n \geqslant 2$, то $\lambda_n \left(1 - \frac{3}{2n}\right) \geqslant \lambda_{n-1}$.*

Найдём явную оценку для λ_n.

Теорема 2. $\lambda_n \geqslant n^{3/2}$.

Доказательство проведём индукцией по n.

Базис индукции: $n = 1$. Имеем: $\lambda_1 = 1 \geqslant 1^{3/2}$.

Индуктивный переход: $n - 1 \mapsto n$. Рассмотрим цепочку неравенств:

$$\lambda_n \left(1 - \frac{3}{2n}\right) \geqslant \lambda_{n-1} \geqslant (n-1)^{3/2} \geqslant n^{3/2}\left(1 - \frac{3}{2n}\right). \qquad (8)$$

Первое из них доказано в лемме 5, второе верно по предположению индукции. Для доказательства третьего проведём выкладки:

$$(n-1)^{3/2} \geqslant n^{3/2}\Big(1-\frac{3}{2n}\Big),$$

$$2n(n-1)^{3/2} \geqslant n^{3/2}(2n-3),$$
$$2(n-1)^{3/2} \geqslant \sqrt{n}(2n-3),$$
$$4(n-1)^3 \geqslant n(2n-3)^2,$$
$$4n^3-12n^2+12n-4 \geqslant 4n^3-12n^2+9n,$$
$$3n \geqslant 4,$$

а это верно при $n \geqslant 2$. Из неравенства (8) следует $\lambda_n \geqslant n^{3/2}$. $\square$

Упражнение 14. *а) Докажите, что* $\lambda_3 \geqslant 9$*;*
б) Допустим, $\lambda_5 = 28$*. Докажите тогда, что* $\lambda_6 \geqslant 39$*.*

Упражнение 15. *Найдите сложность функции голосования трёх переменных: значение функции равно единице тогда и только тогда, когда большинство её переменных (две или три) принимают значение* 1.

§ 8. Нижняя оценка n^2

В заключительном параграфе этой главы мы докажем ещё одну нижнюю оценку сложности, на этот раз — близкую к окончательной, а именно, $\lambda_n \geqslant n^2$. Мы уже отмечали, что точное значение числа λ_n в общем случае не найдено, но известна верхняя оценка $\lambda_n < 9/8n^2$. Разрыв между верхней и нижней оценкой ограничен константой (в данном случае 9/8). В таких случаях мы говорим, что сложность найдена *по порядку*.

Если метод доказательства предыдущего параграфа тесно связан со спецификой функции Λ_n (при подстановке констант линейная функция переходит в линейную), то метод, который мы изложим ниже, может быть применён к любой функции (но давать, возможно, меньшую оценку). Сложность функции мы оценим некоторой величиной, вычисляемой по функции более просто, чем собственно её сложность. Такая величина называется *функционалом* сложности. Доказательство того, что данная величина — функционал сложности, обычно не сложно и производится по индукции. Сложнее придумать сам функционал. Это — своего рода искусство.

До конца параграфа будем считать, что все рассматриваемые функции зависят от одних и тех же переменных — $x_1, \ldots, x_n$. Таким образом, каждая булева функция определена на множестве наборов длины n, состоящих из нулей и единиц. Обозначим через $f^{-1}(1)$ множество тех наборов, на которых функция f равна единице, а через $f^{-1}(0)$ — множество тех наборов, на которых функция f равна 0. Далее будем использовать обозначения из теории множеств: $a \in B$ (элемент a принадлежит множеству B), $A \subseteq B$ (каждый элемент множества A принадлежит множеству B) и $|A|$ (число элементов в множестве A).

Пусть M_0 и M_1 — такие множества наборов, что $M_0 \subseteq f^{-1}(0)$ и $M_1 \subseteq f^{-1}(1)$. Два набора $\tilde{\alpha}$ и $\tilde{\beta}$ назовём *соседними*, если они отличаются ровно по одной координате. Например, набор при $n = 3$ набор $(0, 1, 0)$ является соседним с наборами $(1, 1, 0)$, $(0, 0, 0)$ и $(0, 1, 1)$. Обозначим через R множество всех таких пар наборов $(\tilde{\alpha}, \tilde{\beta})$, что $\tilde{\alpha} \in M_0$, $\tilde{\beta} \in M_1$ и наборы $\tilde{\alpha}$ и $\tilde{\beta}$ являются соседними. Отметим, что множество R задаётся множествами M_0 и M_1.

Теорема 3. *Для любой функции f и любых допустимых непустых множеств M_0, M_1, R выполнено:*

$$L_\Phi(f) \geqslant \frac{|R|^2}{|M_0| \cdot |M_1|}. \tag{9}$$

Прежде, чем доказывать эту теорему, выведем с её помощью оценку n^2. Пусть $f = \Lambda_n$, $M_0 = f^{-1}(0)$, $M_1 = f^{-1}(1)$. Рассмотрим произвольный набор $\tilde{\alpha} \in M_0$ и произвольный соседний с ним набор $\tilde{\beta}$. Заметим, что сумма элементов набора $\tilde{\beta}$, по сравнению с набором $\tilde{\alpha}$, увеличилась или уменьшилась на единицу. Значит, если сумма была чётна, то она станет нечётной, следовательно, значение функции f на наборе $\tilde{\beta}$ равно 1, т. е. $\tilde{\beta} \in M_1$. А тогда пара $(\tilde{\alpha}, \tilde{\beta})$ принадлежит множеству R.

Мы видим, что каждому набору $\tilde{\alpha} \in M_0$ соответствует столько пар наборов из множества R, сколько у $\tilde{\alpha}$ соседних наборов, а их ровно n. Наконец, заметим, что каждому набору $\tilde{\alpha}$ соответствуют свои пары наборов вида $(\tilde{\alpha}, \tilde{\beta})$. Таким образом, множество R как минимум в n раз больше (по числу элементов) множества M_0, т. е. $|R| \geqslant n|M_0|$. Аналогично доказывается неравенство $|R| \geqslant n|M_1|$. Перемножив их, получим $|R|^2 \geqslant n^2|M_0| \cdot |M_1|$. Отсюда и из (9) заключаем:

$$\lambda_n = L_\Phi(\Lambda_n) \geqslant \frac{|R|^2}{|M_0| \cdot |M_1|} \geqslant n^2.$$

Мы доказали

Следствие 5. $\lambda_n \geqslant n^2$.

Теперь **докажем** теорему 3. Обозначим через F минимальную формулу для функции f. Проведём индукцию по построению формулы F.

Базис индукции: F — переменная. Пусть $F = x_i$. Напомним, мы считаем, что функция f зависит от переменных $x_1, \ldots, x_n$. Множество M_0 состоит из некоторых наборов, у которых i-я координата равна 0, а множество M_1 — из некоторых наборов, у которых i-я координата равна 1. Для каждого набора $\tilde{\alpha} \in M_0$ существует единственный соседний набор $\tilde{\beta}$, который может принадлежать множеству M_1, а именно, $\tilde{\beta}$ отличается от $\tilde{\alpha}$ по i-й координате. Таким образом, каждому набору $\tilde{\alpha} \in M_0$ соответствует не более одной пары $(\tilde{\alpha}, \tilde{\beta})$ из множества R. Следовательно, $|R| \leqslant |M_0|$. Аналогично доказывается неравенство $|R| \leqslant |M_1|$. Перемножив их, получим $|R|^2 \leqslant |M_0| \cdot |M_1|$. Заметим, что $L(f) = 1$. Окончательно имеем:

$$L(f) = 1 \geqslant \frac{|R|^2}{|M_0| \cdot |M_1|}.$$

Индуктивный переход. Возможны случаи: 1) $F = \overline{G}$; 2) $F = G \,\&\, H$; 3) $F = G \vee H$. Третий случай можно свести к первым двум, если воспользоваться законом де Моргана: $G \vee H = \overline{\overline{G} \,\&\, \overline{H}}$. Нужно только заметить, что при этом преобразовании сложность формулы не изменяется.

Случай 1). Пусть формула G задаёт функцию g. Нам даны множества $M_0 \subseteq f^{-1}(0)$ и $M_1 \subseteq f^{-1}(1)$ (множество R по ним определяется). Задача состоит в том, чтобы построить множества M_0 и M_1 для функции g; мы их будем обозначать через M_0^g и M_1^g. При этом требуется, чтобы выполнялись вложения $M_0^g \subseteq g^{-1}(0)$, $M_1^g \subseteq g^{-1}(1)$. После того, как эти множества будут построены, мы применим к функции g предположение индукции, затем вернёмся к прежним множествам M_0 и M_1 и преобразуем нижнюю оценку для функции g к нижней оценке для f.

В случае 1) ситуация устроена просто. Так как $g = \overline{f}$, то $g^{-1}(0) = f^{-1}(1)$ и $g^{-1}(1) = f^{-1}(0)$. Поэтому достаточно множества M_0 и M_1 поменять местами, т. е. положить $M_0^g = M_1$, $M_1^g = M_0$. При этом в множестве R^g (т. е. множестве R для функции g) будет столько же элементов, сколько их имеется в R. Наконец заметим, что сложности функций f и g равны (упражнение 7 из § 3). Используя предположение индукции для функции g, получим

$$L_\Phi(f) = L_\Phi(g) \geqslant \frac{|R^g|^2}{|M_0^g| \cdot |M_1^g|} = \frac{|R|^2}{|M_1| \cdot |M_0|}.$$

Случай 2). Схема доказательства та же, что и в случае 1. Через g и h обозначим функции, заданные формулами G и H. Нам даны множества M_0, M_1 и R для функции f. Построим по ним множества M_0^g, M_1^g, R^g для функции g и множества M_0^h, M_1^h и R^h, соответствующие функции h. Рассмотрим множество M_1. Если $\tilde{\beta} \in M_1$, то f на наборе $\tilde{\beta}$ принимает значение 1. Тогда из равенства $f = gh$ следует, что функции g и h на наборе $\tilde{\beta}$ принимают значения 1. Следовательно, ничто не препятствует тому, чтобы включить набор $\tilde{\beta}$ как в множество M_1^g, так и в множество M_1^h. Так мы и поступим, положив

$$M_1^g = M_1^h = M_1. \tag{10}$$

С множеством M_0 ситуация сложнее. Если $\tilde{\alpha} \in M_0$, то $f(\tilde{\alpha}) = 0$. Тогда возможны три случая: а) $g(\tilde{\alpha}) = 0$ и $h(\tilde{\alpha}) = 0$; б) $g(\tilde{\alpha}) = 0$ и $h(\tilde{\alpha}) = 1$; в) $g(\tilde{\alpha}) = 1$ и $h(\tilde{\alpha}) = 0$. Другими словами, множество

M_0 естественно распадается на три части, соответствующие трём этим случаям. Как из этих частей скомпоновать два множества M_0^g и M_0^h — не очевидно. Выясним, при каких условиях множества M_0^g и M_0^h будут определены корректно (т. е. будет выполнено $M_0^g \subseteq g^{-1}(0)$, $M_0^h \subseteq h^{-1}(0)$)

Любой набор из случая а) может быть включён как в множество M_0^g, так и в множество M_0^h. Набор из случая б) может быть включён только в множество M_0^g. Наконец, набор из случая в) может быть включён только в M_0^h. Будем действовать в рамках этих ограничений. Все наборы из случая б) отнесём к множеству M_0^g, наборы из случая в) — к множеству M_0^h, а наборы из случая а) как-нибудь разделим между множествами M_0^g и M_0^h. Для нас важно только, что каждый элемент из M_0 отнесён либо к множеству M_0^h, либо к M_0^g, но не к обоим одновременно. Другими словами, выполнено

$$|M_0^g| + |M_0^h| = |M_0|. \tag{11}$$

Наконец, выясним, как соотносятся между собой множества R, R^g и R^h. Рассмотрим произвольную пару $(\tilde{\alpha}, \tilde{\beta}) \in R$. Набор $\tilde{\alpha} \in M_0$ отнесён нами или к множеству M_0^g, или к M_0^h, а набор $\tilde{\beta} \in M_1$ в силу (10) принадлежит как множеству M_1^g, так и M_1^h. Если $\tilde{\alpha}$ отнесён к M_0^g, то пара $(\tilde{\alpha}, \tilde{\beta})$ принадлежит множеству R^g; в противном случае эта пара принадлежит множеству R^h. Кроме того, множества R^g и R^h не пересекаются т. е. не содержат одну и ту же пару $(\tilde{\alpha}, \tilde{\beta})$, так как в противном случае набор $\tilde{\alpha}$ принадлежал бы как множеству M_0^g, так и множеству M_0^h. Итак, каждая пара из R попадает либо в множество R^g, либо в R^h, но не в оба одновременно, а значит,

$$|R^g| + |R^h| = |R|. \tag{12}$$

Выпишем цепочку равенств и неравенств, а затем поясним её:

$$L_\Phi(f) = L_\Phi(g) + L_\Phi(h) \geqslant \frac{|R^g|^2}{|M_0^g| \cdot |M_1^g|} + \frac{|R^h|^2}{|M_0^h| \cdot |M_1^h|} \geqslant$$
$$\geqslant \frac{(|R^g| + |R^h|)^2}{(|M_0^g| + |M_0^h|) \cdot |M_1|} = \frac{|R|^2}{|M_0| \cdot |M_1|}. \tag{13}$$

Первое равенство вытекает из очевидного равенства $L(F) = L(G) + L(H)$ и следующего соображения: формула F минимальна для f (т. е. $L(F) = L_\Phi(f)$), поэтому и формулы G и H минимальны для, соответственно, функций g и h (т. е. $L(G) = L_\Phi(g)$ и $L(H) = L_\Phi(h)$). Действительно, если бы хотя бы одна из формул G, H допускала упрощение, то и формулу F можно было бы упростить, подставив в неё упрощённую версию формулы G (или H).

Далее, первое неравенство в (13) следует из предположения индукции для функций g и h. Последнее равенство в (13) следует из равенств (11) и (12). Нам осталось доказать второе неравенство в (13). Во первых, воспользуемся равенством (10) и разделим обе части неравенства на $|M_1|$. Осталось доказать неравенство:

$$\frac{|R^g|^2}{|M_0^g|} + \frac{|R^h|^2}{|M_0^h|} \geqslant \frac{(|R^g| + |R^h|)^2}{|M_0^g| + |M_0^h|}.$$

Обозначим: $p = |R^g|$, $q = |R^h|$, $r = |M_0^g|$, $s = |M_0^h|$. Перепишем неравенство в новых обозначениях и преобразуем его:

$$\frac{p^2}{r} + \frac{q^2}{s} \geqslant \frac{(p+q)^2}{r+s},$$

$$p^2 s(r+s) + q^2 r(r+s) \geqslant rs(p+q)^2,$$

$$p^2 rs + p^2 s^2 + q^2 r^2 + q^2 rs - p^2 rs - 2pqrs - q^2 rs \geqslant 0,$$

$$p^2 s^2 - 2pqrs + q^2 r^2 \geqslant 0,$$

$$(ps - qr)^2 \geqslant 0.$$

Неравенство (13), а вместе с ним и теорема 3, доказаны. □

Упражнение 16. *Найдите, чему равны мощности множеств M_0, M_1 и R, построенных нами при выводе следствия* 5 *из теоремы* 3 *(мощность множества — это число элементов в нём).*

Упражнение 17. *Получите нижнюю оценку для функции голосования от n переменных (n нечётно): функция принимает значение* 1 *только в случаях, когда большинство её переменных принимают значение* 1.

Глава III. Универсальные функции

§ 9. Формулы в линейном базисе

В главе II мы вывели две нелинейные нижние оценки сложности в случае, когда используется базис $B_0 = \{\&, \vee, \neg\}$. В этом и следующем параграфах мы рассмотрим другой базис — $B_1 = \{\&, \vee, \neg, \oplus\}$. Новизна ситуации состоит в том, что все оценки главы II относились к линейной функции $\Lambda_n = x_1 \oplus \ldots \oplus x_n$. Её выражение в базисе B_0 достаточно сложно потому, что базис B_0 не содержит функцию $\oplus$. Если же базис содержит функцию $\oplus$, как, например, B_1, то линейная функция задаётся своей естественной формулой: $x_1 \oplus \ldots \oplus x_n$, а значит, имеет сложность n. Понятно, что использовать методы из главы II в этом случае мы не можем.

Оказывается, что для базиса B_1 существует другой подход к получению нижних оценок сложности, более общий, чем изложенные раньше. Он может быть применён, вообще говоря, к произвольному конечному базису и даже к другим, отличным от формул, классам схем. Например, к контактным схемам, которые мы рассмотрим в § 11. В этом методе оценивается сложность универсальных функций, которые мы введём в § 10. Универсальная функция характеризуется тем, что из неё можно, подставляя константы вместо переменных, получить любую булеву функцию от заданного числа переменных. Метод основан на том факте, что среди булевых функций существуют достаточно сложные. Мы их не может указать явно, но можем доказать их существование, что мы сейчас и сделаем.

Многие понятия и утверждения, относящиеся к базису B_1, вводятся и доказываются аналогично тому, как это было сделано для базиса B_0. Для того, чтобы получить определение формулы в базисе B_1, нужно к пункту б) определения формулы в базисе B_0 приписать, что если F и G — формулы, то $(F \oplus G)$ является формулой. Чтобы определиться с сокращённой записью формул, условимся, что операция $\oplus$ выполняется в самую последнюю очередь. Например, запись $x \oplus y \vee z$ означает: $(x \oplus (y \vee z))$. Сложность формулы и сложность функции в случае базиса B_1 определяются так же, как для базиса B_0: сложность формулы — это число символов переменных в ней, а сложность функции — минимальная сложность формулы, задающей эту функцию. Сложность формулы F для любого базиса имеет одно и то же обозначение $L(F)$, так как не зависит от того, над каким базисом рассматривается формула. Единственное отличие состоит в обозначении сложности функции: сложность функции f при использовании базиса B_1 будем обозначать $L^1_\Phi(f)$. Например, $L^1_\Phi(\Lambda_n) = n$, а $L_\Phi(\Lambda_n) \geqslant n^2$.

Понятие приведённой формулы применимо и к базису B_1. Напомним, приведённой мы называем формулу, в которой каждое отрицание действует непосредственно на переменную. Для каждой формулы существует эквивалентная ей приведённая формула. Чтобы это показать, нужно наряду с тождествами $\overline{\overline{x}} = x$, $\overline{x \mathbin{\&} y} = \overline{x} \vee \overline{y}$ и $\overline{x \vee y} = \overline{x} \mathbin{\&} \overline{y}$ использовать правило снятия отрицания с функции $\oplus$. Мы знаем, что $\overline{x} = x \oplus 1$, поэтому $\overline{x \oplus y} = x \oplus y \oplus 1 = x \oplus \overline{y}$ (или $\overline{x} \oplus y$). Для примера построим приведённую формулу: $\overline{xy \oplus y \vee z} = \overline{xy} \oplus y \vee z = \overline{x} \vee \overline{y} \oplus y \vee z$.

Наша ближайшая задача — оценить число приведённых формул, имеющих заданную сложность l. Вообще говоря, число переменных бесконечно. Поэтому будем рассматривать формулы, в которых содержатся только переменные $x_1, \ldots, x_n$. Пусть приведённая формула имеет сложность l. Если из неё удалить все скобки, то мы получим выражение вида:

$$y_1 \ \circ_1 \ y_2 \ \circ_2 \ \ldots \ \circ_{l-1} \ y_l, \tag{14}$$

где $y_1, \ldots, y_l$ — некоторые переменные, возможно, с отрицаниями (другими словами, литералы), а $\circ_1, \ldots, \circ_{l-1}$ — логические связки: $\&$, $\vee$ или $\oplus$.

Подсчитаем число выражений вида (14). Каждый из l литералов $y_1, \ldots, y_l$ можно выбрать одним из $2n$ способов, а именно, имеется n переменных без отрицаний и n переменных с отрицаниями. Значит l литералов одновременно можно выбрать $(2n)^l$ способами. Аналогично, каждую из связок $\circ_1, \ldots, \circ_{l-1}$ можно выбрать тремя способами: $\&$, $\vee$ или $\oplus$, а значит, одновременно все связки можно выбрать 3^{l-1} способами. Литералы и связки выбираются независимо друг от друга, значит, общее число различных выражений вида (14) равно $(2n)^l \cdot 3^{l-1}$. Далее, нужно подсчитать число вариантов расстановки скобок в выражении (14), умножить его на число различных выражений (14) и мы получим число формул сложности l. Однако подсчёт числа вариантов расстановки скобок достаточно сложен (ответом являются числа Каталана) и не входит в наши планы. Этот вопрос мы оставляем заинтересованным читателям.

Мы же оценим число формул другим, достаточно грубым способом. А именно, приведённая формула — это выражение, состоящее из следующего набора символов: $2n$ литералов, 3 логических связки и две скобки; итого — $2n+5$ различных символов. Подсчитаем теперь длину формулы, т. е. общее число символов в ней. В формуле имеется l литералов, $(l-1)$ связка и некоторое количество скобок. Согласно определению формулы, каждой связке соответствует ровно одна пара скобок, а значит, общее число скобок равно $2(l-1)$. Итак, длина формулы равна $4l-3$. Количество всевозможных выражений длины $4l-3$, в которых каждый символ выбирается $(2n+5)$ способами, равно $(2n+5)^{4l-3}$. Каждая рассматриваемая формула входит в это множество выражений, поэтому искомое число формул не больше чем $(2n+5)^{4l-3}$. Для удобства дальнейших вычислений заменим это число бóльшим, а именно, $(2n+5)^{4l-3} \leqslant (7n)^{4l}$, так как $n \geqslant 1$. Мы доказали

Утверждение 8. *Число приведённых формул сложности l, в которые входят только переменные $x_1, \ldots, x_n$, не больше чем $(7n)^{4l}$.*

Следствие 6. *Число приведённых формул, содержащих только переменные $x_1, \ldots, x_n$ и имеющих сложность, не большую чем l, не превосходит $(7n)^{8l}$.*

Действительно, если сложность формулы не больше чем l, то сложность может быть равна $1, 2, \ldots, l$, поэтому достаточно просуммировать оценки, которые даёт утверждение 8 в случаях $l=1$, $l=2$, …, $l=l$. Получим сумму геометрической прогрессии:

$$(7n)^4 + (7n)^{4\cdot 2} + \ldots + (7n)^{4l} = \frac{(7n)^{4(l+1)} - (7n)^4}{(7n)^4 - 1}.$$

Для упрощения оценки отбросим знаменатель и вычитаемое в числителе (при этом число не уменьшится), а также заменим $l+1$ на $2l$. □

Мы оценили число формул. Теперь подсчитаем число булевых функций от n переменных. Каждая такая функция определена на множестве наборов длины n, состоящих из нулей и единиц. Вначале найдём число таких наборов. На каждом месте набора может стоять нуль или единица. Число способов произвольным образом расставить символы 0 и 1 на n местах равно 2^n. Далее, функция определяется тем, каковы её значения на всех наборах. Значение функции на каждом наборе можно выбрать двумя способами: 0 или 1, а всего имеется 2^n наборов, значит, число способов выбрать значения функции на всех наборах равно 2^{2^n}. Итак

Утверждение 9. *Число булевых функций от n переменных равно 2^{2^n}.*

Теперь сравним число формул и число функций. Рассмотрим все функции от n переменных. Число их конечно, значит, существует некоторая функция, обладающая наибольшей сложностью. Пусть эта сложность равна l. Тогда каждую из рассматриваемых функций можно задать приведённой формулой, сложность которой не больше чем l. Можно считать, что во всех этих формулах используются только переменные $x_1, \ldots, x_n$, однозначно соответствующие аргументам функций. Разным функциям отвечают разные формулы, значит, число функций от n переменных не больше числа приведённых формул, имеющих сложность, равную l или меньшую.

Используя следствие 6 и утверждение 9, получим

$$2^{2^n} \leqslant (7n)^{8l}.$$

Прологарифмируем по основанию 2:

$$2^n \leqslant 8l \cdot \log_2(7n).$$

Наконец, выразим l и упростим вид оценки (считаем, что $n \geqslant 2$):

$$l \geqslant \frac{2^n}{8\log_2(7n)} \geqslant \frac{2^n}{8\log_2(8n)} = \frac{2^n}{8(\log_2 n + 3)} \geqslant \frac{2^n}{8 \cdot 4\log_2 n}.$$

Число l было выбрано так, что существует функция от n переменных, имеющая сложность l, поэтому нами доказана

Лемма 6. *Для каждого $n \geqslant 2$ существует функция от n переменных, сложность которой не меньше чем $c \cdot \frac{2^n}{\log_2 n}$, где $c = 1/32$.*

Упражнение 18. *Найдите такое n, для которого существует функция от n переменных, имеющая сложность, не меньшую n^2 (в случае базиса B_1).*

Упражнение 19*. *Используя числа Каталана, точно подсчитайте число формул, оцененное нами в утверждении 8.*

§ 10. Эффективная нижняя оценка

Сейчас мы введём две универсальные функции, которые будем называть *первой и второй универсальными*. Затем для второй из них докажем нелинейную нижнюю оценку сложности.

Первая универсальная функция зависит от двух наборов переменных: $x_1, x_2, \ldots, x_k$ и $y_1, y_2, \ldots, y_n$, причём $n = 2^k$. Переменные $x_1, \ldots, x_k$ называются *управляющими*, а переменные $y_1, \ldots, y_n$ — *рабочими*. Значение функции вычисляется так. Набор значений управляющих переменных $x_1, \ldots, x_k$ рассматривается как запись некоторого целого числа в двоичной системе счисления. Это число обозначим через x. Тогда $x = 2^{k-1}x_1 + 2^{k-2}x_2 + \ldots + 2x_{k-1} + x_k$. Число x принадлежит отрезку $[0,\ 2^k-1]$, т. е. $[0,\ n-1]$, поэтому различные числа x однозначно соответствуют рабочим переменным: число 0 соответствует переменной y_1, $\ldots$, число $n-1$ — переменной y_n; вообще, число x соответствует переменной y_{x+1}. Значением универсальной функции является значение той рабочей переменной, которая соответствует числу x, определяемому управляющими переменными.

Первую универсальную функцию обозначим через U_n. Её определение можно записать формально:

$$U_n(x_1, \ldots, x_k,\ y_1, \ldots, y_n) = y_{x+1}, \quad \text{где } x = 2^{k-1}x_1 + \ldots + 2x_{k-1} + x_k.$$

Например, пусть $k = 2$. Тогда $n = 2^2 = 4$. Функция U_4 зависит от переменных $x_1, x_2,\ y_1, y_2, y_3, y_4$. Найдём значение $U_4(1,1,\ 0,0,0,1)$. Во-первых, определим число x: $x = 2x_1 + x_0 = 2 \cdot 1 + 1 = 3$. Тогда значением функции является значение переменной y_4, которое равно 1. Аналогично найдём: $U_4(1,0,\ 0,0,0,1) = 0$.

Покажем, что функция U_n является универсальной, а именно докажем следующее

Утверждение 10. *Любую булеву функцию от k переменных можно получить из функции U_n, подставив некоторые константы вместо её рабочих переменных.*

Доказательство. Пусть f — булева функция от k переменных. Можно считать, что этими переменными являются $x_1, \ldots, x_k$. Каждому набору $\tilde{\alpha}$ значений переменных $x_1, \ldots, x_k$ соответствует число x, зависящее от $\tilde{\alpha}$, которое мы обозначим через $x(\tilde{\alpha})$. Числу $x(\tilde{\alpha})$ соответствует рабочая переменная $y_{x(\tilde{\alpha})+1}$. Подставим в функцию U_n вместо переменной $y_{x(\tilde{\alpha})+1}$ константу $f(\tilde{\alpha})$, т. е. значение функции f на наборе $\tilde{\alpha}$. После всех таких подстановок получится некоторая функция g, зависящая только от переменных $x_1, \ldots, x_k$. Покажем, что $g = f$. Для этого достаточно показать равенство этих функций на любом наборе констант $\tilde{\alpha}$. По определению функции U_n, значение $g(\tilde{\alpha})$ равно значению переменной $y_{x(\tilde{\alpha})+1}$. Вместо этой переменной мы подставили константу $f(\tilde{\alpha})$. Таким образом, $g(\tilde{\alpha}) = f(\tilde{\alpha})$. $\square$

В случае $k = 2$, $n = 4$ покажем, как из функции U_n получить функцию И от двух переменных. Установим соответствие между переменными $y_1, \ldots, y_4$ и наборами значений управляющих переменных: y_1 соответствует набору $(0,0)$, y_2 — набору $(0,1)$, y_3 — $(1,0)$, y_4 — $(1,1)$. Вместо рабочих переменных подставим значения функции И на соответствующих наборах, т. е. положим $y_1 = 0 \cdot 0 = 0$, $y_2 = 0$, $y_3 = 0$, $y_4 = 1$. Итак, $x_1 x_2 = U_4(x_1, x_2,\ 0,0,0,1)$.

Теперь определим 2-ю универсальную функцию. Для удобства будем считать, что k также является степенью двойки. Положим $m = n/k$. Так как n, k — степени двойки и $n \geqslant k$, то число m является целым. Введём новые переменные: $z_1, z_2, \ldots, z_n$ и разобъём их на k групп по m переменных в каждой: в первую группу войдут переменные $z_1, \ldots, z_m$, во вторую — $z_{m+1}, \ldots, z_m$, ..., в последнюю группу — переменные $z_{n-m+1}, \ldots, z_n$. Вторая универсальная функция зависит только от переменных $z_1, \ldots, z_n$ и $y_1, \ldots, y_n$. Она вычисляется следующим образом. Вначале по значениям переменных $z_1, \ldots, z_n$ вычисляются значения управляющих переменных, следуя формулам:

$$x_1 = z_1 \vee \ldots \vee z_m, \quad \ldots, \quad x_k = z_{n-m+1} \vee \ldots \vee z_n. \tag{15}$$

Затем к переменным $x_1, \ldots, x_k$ и $y_1, \ldots, y_n$ применяется 1-я универсальная функция.

Вторую универсальную функцию обозначим через V_n. Её формальное определение таково:

$$V_n(z_1, \ldots, z_n,\ y_1, \ldots, y_n) =$$

$$= U_n(z_1 \vee \ldots \vee z_m,\ \ldots,\ z_{n-m+1} \vee \ldots \vee z_n,\ y_1, \ldots, y_n).$$

Опять же рассмотрим случай $k = 2$, $n = 4$. В этом случае $m = 2$. Переменные $z_1, \ldots, z_4$ разделены на две группы: z_1, z_2 и z_3, z_4. Равенства (15) имеют вид: $x_1 = z_1 \vee z_2$, $x_2 = z_3 \vee z_4$. Вычислим функцию, которая получится из V_n, если подставить (как и в предыдущем примере) значения $y_1 = y_2 = y_3 = 0$, $y_4 = 1$:

$$V_4(z_1, z_2, z_3, z_4,\ 0, 0, 0, 1) = U_4(z_1 \vee z_2, z_3 \vee z_4,\ 0, 0, 0, 1) =$$

$$= (z_1 \vee z_2)(z_3 \vee z_4).$$

Перейдём к выводу нижней оценки сложности функции V_n. В начале рассмотрим функцию U_n. Вместо переменных $y_1, \ldots, y_n$ подставим такие константы, чтобы при этом получилась самая сложная функция от k переменных $x_1, \ldots, x_k$. Эту функцию обозначим через f_0. Будем считать, что $k \geqslant 2$. Тогда в силу леммы 6 имеем:

$$L_\Phi^1(f_0) \geqslant c \cdot \frac{2^k}{\log_2 k}, \quad \text{где } c = \frac{1}{32}. \tag{16}$$

Теперь вернёмся к функции V_n и рассмотрим её минимальную формулу F в базисе B_1. Подставим в формулу F такие же константы вместо переменных $y_1, \ldots, y_n$, какие мы подставили в функцию U_n, чтобы получить f_0. В результате этой подстановки мы получим псевдоформулу F_1 (что такое псевдоформула — см. § 4), которая задаёт функцию

$$f_0(z_1 \vee \ldots \vee z_m,\ z_{m+1} \vee \ldots \vee z_m,\ \ldots,\ z_{n-m+1} \vee \ldots \vee z_n).$$

Сделаем следующий шаг. В каждой из групп, на которые разбиты переменные $z_1, \ldots, z_n$, оставим ровно по одной переменной, а вместо остальных переменных подставим нули. Переменную, которую мы оставим в группе, выберем так, чтобы число её вхождений в F_1 было как можно меньше (заметим, что при выводе оценки $n^{3/2}$ в § 7 мы использовали похожую идею подстановок констант). Псевдоформулу, которая останется после всех описанных подстановок, обозначим через F_2. В каждой группе переменных мы из m переменных оставили одну, вносящую наименьший вклад в сложность. Поэтому сложность псевдоформулы F_2 меньше сложности F_1 в m или большее число раз, т. е.

$$L(F_1) \geqslant m \cdot L(F_2). \tag{17}$$

Наконец, псевдоформулу F_2 преобразуем в формулу, которую обозначим через F_3. Это преобразование похоже на то, которое мы осуществляли в утв. 4 § 4. Единственное отличие состоит в том, что нужно добавить правила удаления констант, к которым применяется операция $\oplus$. Правила таковы: $x \oplus 0 = x$, $x \oplus 1 = \overline{x}$. Формула F_3 задаёт функцию, отличающуюся от f_0 только переименованием переменных, а значит,

$$L(F_3) \geqslant L^1_\Phi(f_0). \tag{18}$$

Теперь соберём вместе все накопленные неравенства: (16), (17), (18), а также очевидные неравенства $L(F) \geqslant L(F_1)$ и $L(F_2) \geqslant L(F_3)$:

$$L(F) \geqslant L(F_1) \geqslant m \cdot L(F_2) \geqslant m \cdot L(F_3) \geqslant m \cdot L^1_\Phi(f_0) \geqslant cm \cdot \frac{2^k}{\log_2 k}.$$

Вспомним, что F — минимальная формула, задающая функцию V_n. Вспомним также, что $m = n/k$ и $k = \log_2 n$. Окончательно получим

$$L^1_\Phi(V_n) = L(F) \geqslant c \cdot \frac{n}{k} \cdot \frac{n}{\log_2 k} = c \cdot \frac{n^2}{\log_2 n \cdot \log_2 \log_2 n}.$$

Мы доказали теорему:

Теорема 4. $L^1_\Phi(V_n) \geqslant c \cdot \frac{n^2}{\log_2 n \cdot \log_2 \log_2 n}$, *где* $c = 1/32$.

Сделаем несколько замечаний. Во-первых, при увеличении числа n функция $\log_2 n$, а тем более $\log_2 \log_2 n$, растёт очень медленно, поэтому доказанная оценка почти квадратична по n. Во-вторых, в отличие от функции, существование которой утверждается в лемме 6 (и для которой доказана очень высокая, *экспоненциальная*, нижняя оценка), функция V_n задана *явно*. А именно, мы можем достаточно просто вычислить значение функции V_n на каждом конкретном наборе из её области определения.

Теоретически можно считать, что фраза: "минимальная из самых сложных функций" задаёт функцию, для которой мы в лемме 6 получили высокую нижнюю оценку. Но такое задание не считается явным, так как вычисление значения этой функции на конкретном наборе свелось бы, в первую очередь, к поиску само́й функции, что очень трудоёмко. В теории нижних оценок сложности представляют интерес оценки только для явно заданных функций. Такие оценки называются *эффективными*.

Упражнение 20. *Что изменится в доказательстве теоремы* 4, *если в равенствах* (15) *вместо дизъюнкций использовать конъюнкции?*

Упражнение 21. *Получите нижнюю оценку для функции* V_n, *если базис состоит из отрицания и всех двуместных функций.*

§ 11. Контактные схемы

В этом параграфе мы рассмотрим другой, отличный от формул класс схем — *контактные* схемы. Вообще, *классом схем* мы называем тот или иной способ задания булевых функций, а описания конкретных функций данным способом называются *схемами*.

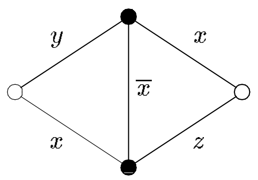

Рис. 3

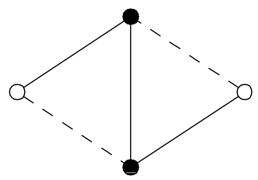
Рис. 4

В основе контактной схемы лежит *граф*. Граф состоит из двух множеств — множества *вершин* и множества *рёбер*. Вершины можно представить себе точками на плоскости или в пространстве. Если говорить формально, то ребро есть пара вершин. Ребро можно представить в виде отрезка или дуги, соединяющей эти две вершины, которые называются *концами* ребра. В данном параграфе рёбра считаются *неориентированными*, т. е. начало и конец ребра равноправны. В контактной схеме каждому ребру соответствует литерал (переменная или её отрицание). Ребро вместе с литералом называется *контактом*. Кроме того, среди вершин выделены две (они могут совпадать), называемые *полюсами*. Пример контактной схемы изображён на рис. 3. В гра́фе, составляющем основу этой схемы, имеется 4 вершины и 5 рёбер. Полюса схемы, в отличие от других вершин, незакрашены.

Если переменная, соответствующая контакту, принимает значение 0 или 1, то сам контакт переходит при этом в одно из двух состояний: замкнутое (проводящее) или разомкнутое (непроводящее). Контакт можно представить себе в виде реле, которое при одном значении переменной проводит ток между концами ребра, а при другом значении — не проводит. Более точно, если контакту приписан литерал x_i (без отрицания), то при $x_i = 1$ контакт замкнут, а при $x_i = 0$ — разомкнут. Если же контакту приписан литерал $\overline{x}_i$, то при $x_i = 1$ контакт разомкнут, а при $x_i = 0$ — замкнут. Например, на рис. 4 изображены замкнутые и разомкнутые контакты схемы из рисунка 3 в случае, когда переменные принимают следующие значения: $x = 0$, $y = z = 1$. Замкнутые контакты изображены сплошной чертой, а разомкнутые — пунктиром.

Если известна проводимость каждого контакта, то можно определить проводимость всей схемы (т. е. проводимость между её полюсами): схема проводит ток, если существует последовательность

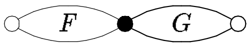

Рис. 5

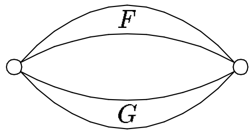

Рис. 6

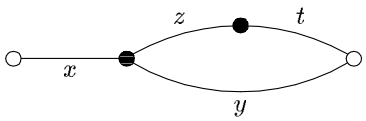

Рис. 7

замкнутых рёбер, по которой можно пройти из одного полюса в другой. Этот случай имеет место на рис. 4. Наконец, определим булеву функцию, вычисляемую контактной схемой. Если на данном наборе значений переменных схема проводит ток, то значение функции равно 1. В противном случае, значение равно 0.

Пусть, например, f_1 — булева функция, вычисляемая схемой, изображённой на рис. 3. Тогда $f_1(1,0,0) = 1$, что продемонстрировано на рис. 4. Найдём формулу, задающую функцию f. Если $x = 0$, то рассматриваемая схема обладает проводимостью только в случае, когда $y = 1$ и $z = 1$, поэтому $f_1(0,y,z) = yz$. Если же $x = 1$, то схема проводит ток в случаях, когда $y = 1$ или $z = 1$; другими словами, $f_1(1,y,z) = y \vee z$. Используя разложение по переменной, получим:

$$f_1(x,y,z) = \overline{x} f_1(0,y,z) \vee x f_1(1,y,z) = \overline{x}yz \vee x(y \vee z).$$

Действуя аналогично, можно по любой контактной схеме построить формулу, задающую ту же функцию. Верно и обратное: любой приведённой формуле в базисе B_0 соответствует некоторая контактная схема. А именно, литералу соответствует контакт, помеченный этим литералом, формуле $F \,\&\, G$ соответствует последовательное соединение схем (рис. 5), формуле $F \vee G$ — параллельное соединение схем (рис. 6). Например, формуле $x(y \vee zt)$ соответствует схема, изображенная на рис. 7. Схемы, соответствующие формулам, называются *параллельно-последовательными* или П-*схемами*.

Не все схемы являются П-схемами. Например, не является таковой схема на рис. 3.

Определим меры сложности, связанные с контактными схемами. Сложность контактной схемы — это число контактов в ней. Сложность схемы S обозначается через $L(S)$. Сложность функции в классе контактных схем определяется как минимальная сложность контактной схемы, вычисляющей данную функцию. Сложность функции f обозначается через $L_K(f)$. Например, из рис. 3 можно заключить, что $L_K(f_1) \leqslant 5$. Заметим, что если по формуле строить схему, то сложность остаётся прежней, а если по схеме строить формулу, то сложность может увеличиться, что мы ранее продемонстрировали на примере.

Далее мы применим к контактным схемам метод получения нижних оценок сложности, связанный с универсальными функциями. План действий аналогичен тому, что мы делали в §§ 9, 10: вначале оценим число контактных схем данной сложности; затем сравним это число с числом булевых функций и выведем оценку для сложности са́мой сложной функции. Наконец, с помощью метода подстановки констант вместо переменных докажем нижнюю оценку для второй универсальной функции. Вершина графа называется *изолированной*, если не существует ребра с концом в данной вершине. Контакную схему назовём *неизолированной*, если в ней не существует изолированных вершин, за исключением, быть может, полюсов. Если из контактной схемы удалить изолированную вершину, отличную от полюсов, то сложность схемы и вычисляемая ею функция не изменятся. Поэтому достаточно ограничиться неизолированными схемами.

Утверждение 11. *Число неизолированных контактных схем сложности $l \geqslant 1$, содержащих только переменные $x_1, \ldots, x_n$, не превосходит $(3nl)^{6l}$.*

Доказательство. Вначале оценим число графов с l рёбрами и не более чем k вершинами. Вершины занумеруем числами $1, 2, \ldots, k$ (возможно, не все вершины присутствуют). Для того, чтобы однозначно задать граф, достаточно указать концы каждого ребра. А именно, для каждого ребра нужно указать два числа: номера тех вершин, которые являются концами этого ребра. Таким образом, граф задаётся последовательностью из $2l$ чисел: первые два числа есть номера концов первого ребра, следующие два числа — номера концов второго ребра и т. д. Номер каждой вершины — это число от 1 до k, поэтому количество таких последовательностей равно $(2l)^k$. Число рассматриваемых графов не больше данной величины.

Рассмотрим теперь неизолированную контактную схемы сложности l. У каждого ребра имеется два конца, поэтому число вершин, являющихся концом хотя бы одного ребра, не превосходит $2l$. В схеме не допускаются изолированные вершины, за исключением полюсов, поэтому число изолированных вершин не больше двух. Итого, общее число вершин не больше чем $2l + 2$. Положим $k = 2l + 2$. Тогда число графов, соответствующих рассматриваемым схемам, не больше чем $(2l)^k$.

Каждому ребру графа соответствует литерал, который можно выбрать одним из $2n$ способов. Значит, все литералы одновременно можно выбрать $(2n)^l$ способами. Наконец, каждый полюс можно выбрать k способами, следовательно, для выбора обоих полюсов есть не больше чем k^2 вариантов. Перемножив найденные оценки для числа графов, расстановки литералов и выбора полюсов, получим, что искомое число схем не больше чем

$$(2l)^k \cdot (2n)^l \cdot k^2.$$

Учтём равенство $k = 2l - 2$ и преобразуем оценку с целью её упрощения:

$$(2l)^{2l+2}(2n)^l(2l+2)^2 \leqslant (2l)^{2l+2}(2n)^l(4l)^2 = 2^{2l+2+l+4}l^{2l+2+2}n^l =$$

$$= 2^{3(l+2)}l^{2(l+2)}n^l \leqslant (8l^2n)^{l+2} \leqslant (9n^2l^2)^{3l} = (3nl)^{6l}. \ \square$$

Лемма 7. *Существует функция от n переменных, сложность которой в классе контактных схем не меньше $c_1 \cdot \frac{2^n}{n}$, где $c_1 = 1/7$.*

Доказательство. Обозначим через l сложность самой сложной функции от n переменных в классе контактных схем. Тогда для каждой функции от n переменных найдётся схема сложности, не большей чем l. Можно считать, что схема неизолирована и содержит только переменные $x_1, \ldots, x_n$. Более того, можно дополнить схему лишними контактами, никак с ней не связанными, при этом мы добьёмся того, что сложность схемы станет равной l, а остальные условия не нарушатся. Итак, число рассматриваемых схем должно быть не меньше числа функций от n переменных, а значит,

$$(3nl)^{6l} \geqslant 2^{2^n}.$$

Прологарифмируем по основанию 2:

$$6l \cdot \log_2(3nl) \geqslant 2^n. \tag{19}$$

Далее проведём доказательство от противного. Допустим, что $l \leqslant c_1 \cdot 2^n/n$. Покажем тогда, что неравенство (19) не выполнено. Для этого проведём выкладки:

$$3nl \leqslant 3n \cdot c_1 \cdot \frac{2^n}{n} \leqslant \frac{3}{7} \cdot 2^n \leqslant 2^n,$$

$$\log_2(3nl) \leqslant \log_2(2^n) = n,$$

$$6l \cdot \log_2(3nl) \leqslant 6nl \leqslant \frac{6}{7} \cdot n \cdot \frac{2^n}{n} = \frac{6}{7} \cdot 2^n < 2^n.$$

Итак, (19) не выполнено. Противоречие. □

Теорема 5. *$L_K(V_n) \geqslant c_1 \cdot \frac{n^2}{\log_2^2 n}$, где $c_1 = 1/7$.*

Доказательство повторяет доказательство теоремы 4. Пусть S — минимальная контактная схема для функции V_n. Последовательно построим схемы S_1 и S_2. Схема S_1 получается из S, если вместо рабочих переменных подставить такие константы, при подстановке которых в функцию U_n получается самая сложная (в классе контактных схем) функция f_2. Схема S_2 получается из S_1 подстановкой нулей вместо всех переменных, за исключением избранных по одной из каждой группы.

Отличие от доказательства теоремы 4 состоит в том, что в контактных схемах по другому устроен процесс подстановки константы вместо переменной. А именно, если вместо переменной, соответствующей контакту, подставлена константа, то этот контакт

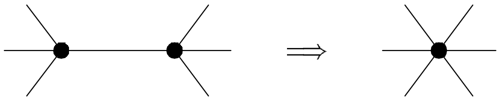

Рис. 8

становится либо всегда замкнутым, либо всегда разомкнутым. Во втором случае контакт можно удалить из схемы, а в первом — отождествить две вершины, являющиеся концами контакта, и затем удалить контакт из схемы. Отождествление вершин означает, что две вершины заменяются одной и она становится концом всех рёбер, ранее оканчившихся в первой или второй отождествляемых вершинах (см. рис. 8). Таким образом, после каждой подстановки константы мы получаем контактную схему и необходимости в схеме S_3 нет.

Наконец, можно выписать цепочку неравенств:

$$L_K(V_n) = L(S) \geqslant L(S_1) \geqslant m \cdot L(S_2) \geqslant m \cdot L_K(f_2) \geqslant$$

$$\geqslant m \cdot c_1 \cdot \frac{2^k}{k} \geqslant c_1 \cdot \frac{n^2}{k^2} = c_1 \cdot \frac{n^2}{\log_2^2 n}. \ \square$$

Упражнение 22. *Постройте минимальную контактную схему для функции f_1, заданной схемой на рис. 3.*

Упражнение 23*. *а) Докажите, что существует функция, сложность которой в классе контактных схем меньше её сложности в классе формул в базисе B_0;*

б) постройте такую функцию.

§ 12. Схемы из функциональных элементов

В заключительном параграфе главы III мы рассмотрим основной и в то же время самый сложный класс схем — схемы из функциональных элементов (СФЭ). С примером СФЭ мы уже встречались на рис. 1. Сейчас мы дадим определение СФЭ, а затем покажем, что в классе СФЭ 1-я и 2-я универсальные функции имеют линейную сложность (под этим понимается, что сложность ограничена сверху некоторой линейной функцией от n). Тем самым мы покажем, что метод получения нижних оценок сложности с помощью универсальных функций неприменим к классу СФЭ. Вообще, в настоящее время неизвестно ни одной нелинейной нижней оценки сложности в классе СФЭ для явно заданной функции. Проблема вывода такой оценки является наиболее актуальной проблемой в теории нижних оценок сложности.

Основу схемы из функциональных элементов составляет граф, на этот раз — *ориентированный*: у каждого ребра однозначно определены начало и конец; ребро *выходит* из своей начальной вершины и *входит* в конечную вершину. Вершины графа разбиты на два множества: входы схемы и элементы. Входы изображаются точками, а элементы — прямоугольниками, как на рис. 1, или треугольниками, как на рис. 9, 10. Каждому входу соответствует переменная, а каждому элементу — функция из заданного базиса (так же, как и формулы, СФЭ рассматриваются над определённым базисом).

В вершину, которая является входом схемы, не могут входить рёбра; в вершину, являющуюся элементом, входит столько рёбер, сколько аргументов имеется у функции, сопоставленной элементу; число рёбер, выходящих из вершины, неограничено. Рёбра, входящие в один элемент, упорядочены, а именно, определено, какое ребро соответствует первому аргументу функции, какое — второму и т. д. Наконец, все элементы схемы занумерованы числами $1, 2, \ldots, l$ так, что выполняется условие: если ребро соединяет два элемента, то оно выходит из элемента с меньшим номером и входит в элемент с бóльшим. Число l называется сложностью схемы.

Пример СФЭ в базисе B_0 изображён на рис. 9. Сложность этой схемы равна 5. Номера элементов выписаны рядом с элементами. Заметим, что B_0 — *симметрический* базис, а именно, любая функция, входящая в B_0, не изменяется при перестановке аргументов. Поэтому порядок следования рёбер, входящих в один и тот же элемент, не важен.

Определим булеву функцию, вычисляемую схемой из функциональных элементов. Рассмотрим произвольный набор значений переменных, соответствующих входам схемы. Припишем входам соответствующие константы из этого набора. Далее вычисление происходит последовательно. На каждом шаге вычисляется значение

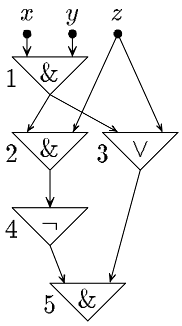

Рис. 9

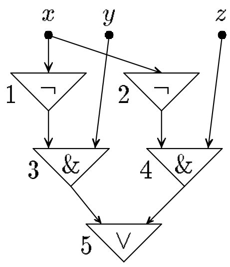

Рис. 10

на *выходе* некоторого элемента: вначале на выходе 1-го элемента, затем — 2-го и т. д. Значение на выходе последнего элемента является значением искомой булевой функции на данном наборе. Значение на выходе очередного элемента вычисляется исходя из значений, поступивших на его *входы*; входы элемента соответствуют рёбрам, входящим в этот элемент. На вход элемента поступает та константа, которая имеется на другом конце соответствующего ребра. Другим концом может быть вход схемы или выход элемента с меньшим номером, для которого значение уже определено. К константам, поступившим на входы элемента, применяется функция, приписанная элементу; полученная в результате константа подаётся на выход элемента.

Рассмотрим схему из рис. 9. Пусть она вычисляет функцию f_3 от переменных x, y, z. Найдём $f_3(1,1,1)$. Входам схемы приписаны значения 1. На вход 1-го элемента поступают две единицы, значит, на выход 1-го элемента поступает $1 \cdot 1 = 1$. Аналогично ведут себя 2-й и 3-й элементы: на все их входы поступают значения 1 и, соответственно, на выходе мы получаем единицу. Далее, на вход 4-го элемента поступает 1, а на его выходе образуется $\overline{1} = 0$. Наконец, на входы 5-го элемента поступают константы 0 или 1; на его выходе образуется константа 0. Итак, $f_3(1,1,1) = 0$. Теперь зададим функцию f_3 формулой. Это можно сделать последовательно, выписывая в виде формул функции, вычисляемые на выходах 1-го, 2-го и последующих элементов. На выходе 1-го элемента вычисляется функция xy. На входы 2-го элемента подаются функции xy и z, поэтому на его выходе вычисляется функция xyz. Аналогично на выходе 3-го элемента вычисляется функция $xy \vee z$, на выходе 4-го — $\overline{xyz}$. Наконец, на выходе всей схемы вычисляется функция $\overline{xyz}\ \&\ (xy \vee z)$. Итак, $f_3(x, y, z) = \overline{xyz}\ \&\ (xy \vee z)$.

Подобным образом можно по каждой схеме выписать формулу, задающую ту же функцию. Верно и обратное: по каждой формуле можно построить схему, задающую ту же функцию. Схема строится непосредственно по формуле. Например, схема, соответствующая формуле $\overline{x}y \vee \overline{x}z$, изображена на рис. 10. Заметим, что если формула приведена и имеет сложность l, то в ней содержится $l-1$ двуместная операция и не больше, чем l отрицаний. Следовательно, сложность схемы, построенной по такой формуле, не больше, чем $2l - 1$. Мы видим, что при переходе от формулы к схеме сложность увеличивается не более чем в два раза, т. е. не изменяется по порядку. В обратную сторону это утверждение неверно. Заметим, что схема, построенная по формуле, обладает следующим свойством: из каждого элемента, кроме последнего, выходит ровно одно ребро. Другими словами, выход элемента используется в дальнейшем ровно один раз. Для произвольных схем это не верно. Например, выход первого элемента в схеме на рис. 9 используется два раза. За счёт этого эффекта одинаковые фрагменты формулы можно реализовать в схеме один раз, а значит, при раскрытии схемы в формулу сложность может существенно увеличиться.

Перейдём к выводу верхних оценок сложности для универсальных функций. Вначале построим формулу, реализующую функцию U_n. Разложим эту функцию по переменной x_1:

$$U_n(x_1, x_2, \ldots, x_k,\ y_1, \ldots, y_n) = \overline{x}_1 U_n^0 \vee x_1 U_n^1, \text{ где} \tag{20}$$

$U_n^0 = U_n(0, x_2, \ldots, x_k,\ y_1, \ldots, y_n)$, $U_n^1 = U_n(1, x_2, \ldots, x_k,\ y_1, \ldots, y_n)$.

Значение функции U_n^0 равно значению той переменной y_{x+1}, для которой число x имеет двоичную запись $0x_2 \ldots x_k$. В таком случае число x принадлежит отрезку $[0, n/2 - 1]$. Поэтому функция U_n^0 не зависит существенно от переменных $y_{n/2+1}, \ldots, y_n$. В то же время значение любой из переменных $y_1, \ldots, y_{n/2}$ может быть использовано в качестве значения функции в зависимости от значений управляющих переменных $x_2, \ldots, x_k$. Поэтому с точностью до переименования переменных и исключения несущественных переменных функция U_n^0 равна функции $U_{n/2}$. А именно,

$$U_n^0 = U_{n/2}(x_2, \ldots, x_k,\ y_1, \ldots, y_{n/2}). \tag{21}$$

Аналогично функция U_n^1 существенно не зависит от переменных $y_{n/2+1}, \ldots, y_n$. При исключении этих переменных из U_n^1 и переименовании оставшихся мы получим функцию $U_{n/2}$:

$$U_n^1 = U_{n/2}(x_2, \ldots, x_k,\ y_{n/2+1}, \ldots, y_n). \tag{22}$$

Обозначим $u_n = L_\Phi(U_n)$. Тогда из (20)–(22) следует:

$$u_n \leqslant 2(u_{n/2} + 1).$$

Прибавим 2 к обеим частям:

$$u_n + 2 \leqslant 2(u_{n/2} + 2).$$

Имеем цепочку неравенств:

$$u_n + 2 \leqslant 2(u_{n/2} + 2) \leqslant 4(u_{n/4} + 2) \leqslant \ldots \leqslant n(u_1 + 2).$$

Заметим, что $U_1(y_1) = y_1$, поэтому $u_1 = 1$. Выразив u_n, мы докажем лемму:

Лемма 8. $L_\Phi(U_n) \leqslant 3n - 2$.

Обозначим через $L_C(f)$ сложность функции f в классе СФЭ в базисе B_0, т. е. минимальную сложность схемы в данном базисе, вычисляющей функцию f.

Следствие 7. $L_C(U_n) \leqslant 6n$.

Ранее мы заметили, что по всякой приведённой формуле можно построить СФЭ, сложность которой превосходит сложность формулы не больше, чем в два раза. □

Наконец, оценим сложность функции V_n. Будем строить схему согласно определению функции V_n, используя уже построенную

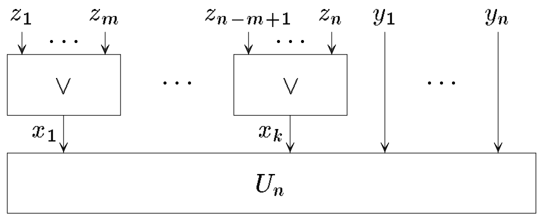

Рис. 11

схему для функции U_n (см. рис. 11). Вначале вычислим значения вспомогательных переменных $x_1, \dots, x_k$ по равенствам (15). Выходы схем, вычисляющих переменные $x_1, \dots, x_k$, соединим с входами уже построенной схемы для функции U_n. Для вычисления одной переменной x_i нужно $m-1$ элементов ИЛИ. Значит, всего необходимо $k(m-1)$ элементов ИЛИ. В итоге получим оценку:

$$L_C(V_n) \leqslant k(m-1) + L_C(U_n) < km + 6n = 7n.$$

Таким образом, доказана

Теорема 6. $L_C(V_n) \leqslant 7n$.

Теорема 6 показывает, что функция V_n имеет линейную сложность в классе СФЭ. Сопоставляя эту теорему с предыдущими (4, 5), видим, что некоторые функции можно задать с помощью СФЭ более экономно, чем с помощью контактных схем и формул. Кроме того, мы знаем, что СФЭ для любой функции не намного сложнее формулы для этой же функции. Поэтому класс СФЭ *сильнее* класса формул. Верно ли что класс СФЭ сильнее класса контактных схем — неизвестно.

Упражнение 24. *Найдите верхнюю оценку для числа* $L_\Phi(V_n)$.

Упражнение 25. *Постройте СФЭ, вычисляющую функцию* $x_1 \oplus \ldots \oplus x_n$ *с линейной сложностью.*

Упражнение 26*. *Докажите, что для некоторой константы* $c > 0$ *при любом* n *существует функция* n *переменных, сложность которой в классе СФЭ в базисе* B_0 *не меньше, чем* $c \cdot \frac{2^n}{n}$.

Глава IV. Булевы суммы

§ 13. Матрицы без прямоугольников

Матрицей размера $n \times m$ называется таблица, в которой имеется n строк, m столбцов и на пересечении каждой строки и каждого столбца стоит элемент из некоторого множества, как правило — число. Элементы матрицы не разделяются, как принято в таблицах, горизонтальными и вертикальными линиями; матрица заключается в круглые скобки. Мы будем рассматривать 0–1-матрицы, в которых элементами являются нули и елиницы. Пример 0–1-матрицы размера 3×4 приведён на рис. 12.

Нас будут интересовать матрицы специального вида, а именно, матрицы без прямоугольников. *Прямоугольником* в матрице назовём четыре единицы, стоящие на пересечении двух строк и двух столбцов. В матрице на рис. 12 имеются прямоугольники. Например, первая и вторая строки (строки матрицы нумеруются сверху вниз, а столбцы — слева направо) при пересечении с первым и четвёртым столбцами образуют прямоугольник: во всех четырёх точках их пересечения стоят единицы. Пример матрицы без прямоугольников приведён на рис. 13. В этой матрице 8 единиц. Зададимся вопросом: каково может быть максимальное число единиц в матрице размера $n \times n$ без прямоугольников. Вначале покажем, что число единиц не может быть больше некоторой величины.

Лемма 9. *В любой 0–1-матрице размера $n \times n$ без прямоугольников содержится не больше чем $n^{3/2} + n/2$ единиц.*

Доказательство. В матрице имеется n столбцов, следовательно, число пар столбцов равно C_n^2 (имеется в виду, что каждая пара неупорядочена и столбцы в паре различны). Напомним, $C_n^2 = \frac{n(n-1)}{2}$. Число пар можно подсчитать так: первый столбец в паре можно выбрать n способами, второй столбец — $(n-1)$ способом, так как второй столбец должен отличаться от первого. Значит, два столбца одновременно можно выбрать $n(n-1)$ способами. Кроме того, каждую пару $\{a, b\}$ мы сосчитали два раза: первый раз, когда

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

Рис. 12

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

Рис. 13

вначале выбрали a, а затем — b, второй раз — когда вначале выбрали b, а затем — a. Поэтому число $n(n-1)$ нужно разделить на 2. Будем считать, что для любого действительного числа x запись C_x^2 является обозначением выражения $\frac{x(x-1)}{2}$.

Назовём *парой единиц* две единицы, стоящие в одной и той же строке. Для каждого $i = 1, 2, \ldots, n$ число единиц в i-й строке матрицы обозначим через m_i. Тогда число пар единиц, расположенных в i-й строке, равно $C_{m_i}^2$. Следовательно, число пар единиц в матрице равно $C_{m_1}^2 + C_{m_2}^2 + \ldots + C_{m_n}^2$. Рассмотрим произвольную пару столбцов и пересечения этих столбцов со всеми строками. Найдётся не более одной строки, на пересечении с которой в данных двух столбцах стоят единицы. Действительно, если бы существовали две такие строки, то мы бы получили прямоугольник. Значит, в данной паре столбцов находится не более одной пары единиц. Поэтому число пар единиц не больше числа пар столбцов:

$$C_{m_1}^2 + C_{m_2}^2 + \ldots + C_{m_n}^2 \leqslant C_n^2. \tag{23}$$

Напомним, функция $f\colon \mathbb{R} \to \mathbb{R}$ называтся *выпуклой* (на $\mathbb{R}$), если для любых двух чисел x, y и для любых двух неотрицательных чисел α, β таких, что $\alpha + \beta = 1$, выполнено

$$f(\alpha x + \beta y) \leqslant \alpha f(x) + \beta f(y). \tag{24}$$

Выпуклость означает, что любой отрезок, соединяющий две точки графика функции, лежит над графиком (или на графике). Покажем, что функция C_x^2 выпукла:

$$C_{\alpha x + \beta y}^2 \leqslant \alpha C_x^2 + \beta C_y^2,$$

$$(\alpha x + \beta y)^2 - (\alpha x + \beta y) \leqslant \alpha x^2 - \alpha x + \beta y^2 - \beta y,$$

$$
\begin{gathered}
(\alpha x+\beta y)^2 \leqslant \alpha x^2+\beta y^2,\\
\alpha^2 x^2+2\alpha\beta xy+\beta^2 y^2 \leqslant \alpha x^2+\beta y^2,\\
\alpha(1-\alpha)x^2-2\alpha\beta xy+\beta(1-\beta)y^2 \geqslant 0,\\
\alpha\beta(x-y)^2 \geqslant 0.
\end{gathered}
$$

Для выпуклой функции справедливо неравенство Иенсена: для любых чисел $x_1,\ldots,x_n$ и любых неотрицательных чисел $\alpha_1,\ldots,\alpha_n$ таких, что $\alpha_1+\ldots+\alpha_n=1$, выполнено:

$$
f(\alpha_1 x_1+\ldots+\alpha_n x_n) \leqslant \alpha_1 f(x_1)+\ldots+\alpha_n f(x_n). \tag{25}
$$

Неравенство (25) можно доказать индукцией по n, опираясь на неравенство (24); см. упражнение 28. Применим неравенство Иенсена к функции C_x^2 и константам $\alpha_1=\ldots=\alpha_n=1/n$:

$$
C^2_{(m_1+\ldots+m_n)/n} \leqslant \frac{1}{n}\bigl(C^2_{m_1}+\ldots+C^2_{m_n}\bigr).
$$

Пусть m — число единиц в матрице. Тогда $m=m_1+\ldots+m_n$. Используя (23), получим

$$
C^2_{m/n} \leqslant \frac{1}{n}\bigl(C^2_{m_1}+\ldots+C^2_{m_n}\bigr) \leqslant \frac{1}{n}C^2_n.
$$

Обозначим $x=m/n$ и проведём выкладки:

$$
\begin{gathered}
\frac{x(x-1)}{2} \leqslant \frac{1}{n}\cdot\frac{n(n-1)}{2}=\frac{n-1}{2},\\
x^2-x-(n-1) \leqslant 0,\\
D=1+4(n-1)=4n-3,\\
\frac{1-\sqrt{4n-3}}{2} \leqslant x \leqslant \frac{1+\sqrt{4n-3}}{2}.
\end{gathered}
$$

Мы знаем, что $x=m/n\geqslant 0$, поэтому

$$
\frac{m}{n} \leqslant \frac{1+\sqrt{4n-3}}{2} \leqslant \frac{1+\sqrt{4n}}{2}=\sqrt{n}+\frac{1}{2}.
$$

$$\begin{pmatrix} 0 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 1 \\ 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 0 \end{pmatrix}$$

Рис. 14

$$\begin{pmatrix} E_p^0 & E_p^0 & \ldots & E_p^{0\cdot(p-1)} \\ E_p^0 & E_p^1 & \ldots & E_p^{1\cdot(p-1)} \\ E_p^0 & E_p^2 & \ldots & E_p^{2\cdot(p-1)} \\ \vdots & \vdots & \ddots & \vdots \\ E_p^0 & E_p^{p-1} & \ldots & E_p^{(p-1)^2} \end{pmatrix}$$

Рис. 15

Окончательно получим

$$m \leqslant n^{3/2} + n/2. \ \square$$

Теперь мы покажем, что для некоторых n существуют матрицы размера $n \times n$ без прямоугольников, содержащие $n^{3/2}$ единиц. Пусть p — простое число и $n = p^2$. Построим матрицу T_n размера $n \times n$, а именно, составим её из блоков. Блок — это часть матрицы, находящаяся на пересечении нескольких рядом стоящих строк и рядом стоящих столбцов. Обозначим через E_p^q блок, изображённый на рис. 14. Единица в первой строке стоит на месте с номером $q' + 1$, где q' — остаток от деления q на p; в каждой следующей строке единица стоит на одну позицию правее, причём за крайней правой позицией следует крайняя левая. Вообще, элемент на пересечении k-й строки и l-го столбца равен единице тогда и только тогда, когда

$$l - k \equiv q \pmod{p},$$

другими словами, числа $l - k$ и q дают одинаковые остатки при делении на p. Например, в случае $p = 2$ блоки E_2^0 и E_2^1 имеют вид:

$$E_2^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad E_2^1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Матрица T_n, составленная из блоков E_p^q для различных q, изображена на рис. 15. Блок с координатами (i, j), т. е. i-й блок сверху и j-й слева равен E_p^{ij}. Например, при $p = 2$ матрица T_n принимает вид:

$$T_4 = \begin{pmatrix} E_2^0 & E_2^0 \\ E_2^0 & E_2^1 \end{pmatrix}.$$

Если здесь обозначения блоков заменить самими блоками, то мы получим матрицу размера 4×4, изображённую на рис. 13.

Докажем, что в матрице T_n нет прямоугольников. Каждый элемент матрицы T_n задаётся четырьмя координатами: i, j, k, l, где i есть номер блока по вертикали, k — номер строки в блоке, j — номер блока по горизонтали, l — номер столбца в блоке. Таким образом, номер строки матрицы задаётся парой чисел (i, k), а номер столбца — парой (j, l). Допустим, что строки с координатами (i_1, k_1) и (i_2, k_2), а также столбцы с координатами (j_1, l_1) и (j_2, l_2) образуют прямоугольник. Выпишем условия того, что все четыре элемента на пересечении данных строк и столбцов равны единице:

$$\begin{cases} l_1 - k_1 \equiv i_1 j_1 \pmod p \\ l_2 - k_1 \equiv i_1 j_2 \pmod p \\ l_1 - k_2 \equiv i_2 j_1 \pmod p \\ l_2 - k_2 \equiv i_2 j_2 \pmod p \end{cases} \tag{26}$$

Сложим первое и последнее сравнения в (26) и вычтем из них второе и третье. Числа k_1, k_2, l_1, l_2 взаимно уничтожатся и мы получим:

$$0 \equiv i_1 j_1 - i_1 j_2 - i_2 j_1 + i_2 j_2 \pmod p,$$

$$0 \equiv (i_1 - j_1)(i_2 - j_2) \pmod p.$$

Видим, что $(i_1 - i_2)(j_1 - j_2)$ делится на p. Число p простое, поэтому или $i_1 - i_2$, или $j_1 - j_2$ делится на p. В то же время числа i_1, i_2, j_1, j_2 принадлежат отрезку $[1, p]$, а значит, $|i_1 - i_2| < p$ и $|j_2 - j_2| < p$. Следовательно, $i_1 - i_2 = 0$ или $j_1 - j_2 = 0$. В первом случае из первого сравнения в (26) вычтем третье и получим

$$k_2 - k_1 \equiv 0 \pmod p.$$

Таким образом, $i_1 = i_2$ и $k_1 = k_2$, значит, выбранные нами строки совпадают. Аналогично, если $j_1 - j_2 = 0$, то из системы (26) найдём, что выбранные нами столбцы совпадают. Прямоугольник должен быть образован разными строками и разными столбцами. Мы получили противоречие, доказывающее, что в матрице T_n нет прямоугольников.

Заметим, наконец, что в каждом блоке содержится p единиц и всего имеется p^2 блоков, поэтому число единиц в матрице T_n равно $p^3 = n^{3/2}$. Мы доказали лемму:

Лемма 10. *Если n — квадрат простого числа, то существует матрица размера $n \times n$ без прямоугольников с $n^{3/2}$ единицами.*

Матрицы без прямоугольников используются, например, в конечной геометрии. А именно, строкам матрицы поставим в соответствие прямые, а столбцам — точки. На пересечении строки и столбца стоит единица, если соответствующая точка лежит на прямой. Тогда аксиома "через любые две точки проходит не более одной прямой" соответствует тому, что матрица не содержит прямоугольников. Можно построить матрицы, удовлетворяющие более сильной аксиоме: "через любые две точки проходит ровно одна прямая". В следующем параграфе мы рассмотрим другое применение матриц без прямоугольников, связанное с нижними оценками сложности.

Упражнение 27. *Постройте 0–1-матрицу размера 7×7, которая удовлетворяет условию: для любых двух столбцов существует ровно одна строка такая, что на её пересечении с данными столбцами стоят две единицы.*

Упражнение 28*. *Докажите неравенство Иенсена.*

§ 14. Схемы в монотонном базисе

В данном параграфе мы рассмотрим схемы из функциональных элементов в базисе $M_0 = \{\&, \vee\}$. Этот базис не является полным, в нём можно выразить только монотонные функции (см. ниже). Как мы уже отмечали в § 12, в настоящее время неизвестны нелинейные эффективные нижние оценки сложности для класса СФЭ в полном базисе. В неполном базисе M_0 такие оценки известны. Мы докажем простейшую из них, а именно, оценим сложность совместного

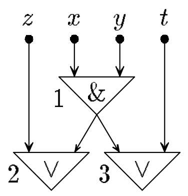

Рис. 16

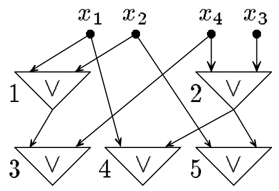

Рис. 17

вычисления системы булевых функций, связанной с матрицей T_n, которую мы ввели в § 13. Отметим, что доказать нижнюю оценку для сложности совместного вычисления нескольких функций проще, чем для сложности одной функции, но даже такие оценки в случае полного базиса неизвестны. В случае же базиса M_0 известны достаточно высокие (экспоненциальные) нижние оценки для сложности одной явно заданной функции. Это говорит о силе ограничения, связанного с отсутствием отрицания.

Скажем, что схема S сложности l вычисляет систему функций $(f_1, \ldots, f_k)$, если для каждого $i = 1, \ldots, k$ функция f_i вычисляется на выходе элемента схемы S, имеющего номер $l-k+i$. Напомним, что элементы схемы S занумерованы числами $1, \ldots, l$, поэтому функции $f_1, \ldots, f_k$ вычисляются на выходах последних k элементов схемы. Класс СФЭ позволяет вычислять некоторые системы булевых функций более экономно, чем при раздельном вычислении каждой функции системы. На рис. 16 приведён пример схемы в базисе M_0, вычисляющей систему функций $(xy \vee z,\ xy \vee t)$ со сложностью 3, тогда как при раздельном вычислении каждой функции сложность будет не менее четырёх.

Булевой суммой называется дизъюнкция одной или нескольких переменных, например, $x \vee y \vee z$. Каждой 0–1-матрице M размера $k \times n$ поставим в соответствие систему булевых сумм $(f_1, \ldots, f_k)$. Для этого столбцы матрицы M свяжем с переменными: первый столбец соответствует переменной x_1, второй — x_2 и т. д. Для каждого $i = 1, \ldots, k$ функция f_i равна дизъюнкции переменных, соответствующих тем столбцам матрицы M, на пересечении которых с i-й строкой стоит единица. Например, матрице, изображённой на рис. 12, соответствует следующая система булевых сумм: $(x_1 \vee x_2 \vee x_4,\ x_1 \vee x_3 \vee x_4,\ x_2 \vee x_3 \vee x_4)$. При раздельном вычислении этих сумм потребуется 6 элементов. Если же использовать преимущества совместного вычисления, а именно, совместно реализовать фрагмент $x_3 \vee x_4$, то достаточно пяти элементов (см. рис. 17).

Теперь рассмотрим матрицу без прямоугольников (например, на рис. 13) и соответствующую систему булевых сумм. Любые две булевы суммы имеют не более одной общей переменной. В таком случае неочевидно, каким образом можно упростить естественную схему раздельного вычисления сумм. Кажется правдоподобным, что такую схему упростить нельзя. Если рассмотреть класс схем в базисе $\{\vee\}$ (другое название — класс *вентильных* схем), то доказать минимальность данной схемы несложно. Мы рассмотрим более сложный случай, когда используется базис M_0. В случае, если используется базис B_0, содержащий отрицание, данная проблема открыта.

Теорема 7. *Пусть M — матрица размера $k \times n$ без прямоугольников, содержащая m единиц. Тогда сложность соответствующей системы булевых сумм в классе СФЭ в базисе M_0 равна $m - k$.*

Доказательство в одну сторону очевидно, а именно, сложность не больше чем $m - k$, так как такую сложность имеет естественная схема, раздельно вычисляющая суммы. Прежде, чем начать доказательство теоремы в обратную сторону, сформулируем её следствие, а затем докажем вспомогательное утверждение.

Следствие 7. *Сложность системы булевых сумм, соответствующей матрице T_n, в классе СФЭ в базисе M_0 равна $n^{3/2} - n$.*

Булеву функцию назовём *монотонной*, если она возрастает по каждой своей переменной (см. § 6). Например, функции & и $\vee$ монотонны, а отрицание — нет. Несложно показать, что композиция монотонных функций монотонна (подобные выкладки мы проводили при доказательстве утв. 6 в § 6). Поэтому формула, составленная из монотонных функций, задаёт монотонную функцию. По каждой СФЭ можно построить формулу, задающую ту же функцию (мы это делали в § 12), поэтому любая схема в базисе M_0 вычисляет монотонную функцию; более того, на выходе любого элемента такой схемы вычисляется монотонная функция (можно рассмотреть часть схемы, в которой данный элемент будет последним).

Пусть $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ и $\tilde{\beta} = (\beta_1, \ldots, \beta_n)$ — два набора, состоящие из нулей и единиц. Скажем, что набор $\tilde{\alpha}$ не меньше, чем $\tilde{\beta}$ (обозначение: $\tilde{\alpha} \leqslant \tilde{\beta}$), если каждая координата набора $\tilde{\alpha}$ не меньше соответствующей координаты набора $\tilde{\beta}$. Например, $(0,0,1) \leqslant (0,1,1)$, но $(0,0,1) \nleqslant (1,0,0)$. Любая монотонная функция f возрастает по всем переменным одновременно, а именно, если $\tilde{\alpha} \leqslant \tilde{\beta}$, то $f(\tilde{\alpha}) \leqslant f(\tilde{\beta})$. Действительно, если $\tilde{\alpha} \leqslant \tilde{\beta}$, то от набора $\tilde{\alpha}$ можно перейти к набору $\tilde{\beta}$ по цепочке наборов: $\tilde{\alpha}, \tilde{\gamma}^1, \ldots, \tilde{\gamma}^s, \tilde{\beta}$, в которой каждый следующий отличается от предыдущего ровно по одной координате и не меньше предыдущего. Из возрастания функции f по каждой переменной следует:

$$f(\tilde{\alpha}) \leqslant f(\tilde{\gamma}^1) \leqslant \ldots \leqslant f(\tilde{\gamma}^s) \leqslant f(\tilde{\beta}).$$

Скажем, что функция f не меньше функции g (обозначение: $f \leqslant g$), если на любом наборе $\tilde{\alpha}$ выполнено неравенство $f(\tilde{\alpha}) \leqslant g(\tilde{\alpha})$. Например, $xy \leqslant x \leqslant x \vee y$, но $x \nleqslant y$.

Утверждение 12. *Пусть f и g — монотонные функции, x_i — переменная, h — булева сумма. Тогда*

а) если $fg \geqslant x_i$, то $f \geqslant x_i$ и $g \geqslant x_i$;

б) если $f \vee g \geqslant x_i$, то $f \geqslant x_i$ или $g \geqslant x_i$;

в) если $fg \leqslant h$, то $f \leqslant h$ или $g \leqslant h$;

г) если $f \vee g \leqslant h$, то $f \leqslant h$ и $g \leqslant h$.

Доказательство. Для удобства будем считать, что $x_i = x_1$ и $h = x_1 \vee x_2 \vee \ldots \vee x_s$.

а) Заметим, что для любых функций f и g выполнено $f \geqslant fg$ Действительно, если $f(\tilde{\alpha}) = 1$, то $f(\tilde{\alpha}) = 1 \geqslant (fg)(\tilde{\alpha})$, а если $f(\tilde{\alpha}) = 0$, то $(fg)(\tilde{\alpha}) = 0$, следовательно, $f(\tilde{\alpha}) = 0 = (fg)(\tilde{\alpha})$. Таким образом, $f \geqslant fg \geqslant x_1$ и $g \geqslant fg \geqslant x_1$.

б) Докажем от противного. Допустим, $f \ngeqslant x_1$ и $g \ngeqslant x_1$. Тогда существуют наборы $\tilde{\alpha}$ и $\tilde{\beta}$ такие, что $f(\tilde{\alpha}) < x_1(\tilde{\alpha})$ и $g(\tilde{\beta}) < x_1(\tilde{\beta})$. Следовательно, $f(\tilde{\alpha}) = g(\tilde{\beta}) = 0$ и $x_1(\tilde{\alpha}) = x_1(\tilde{\beta}) = 1$. Значением выражения $x_1(\tilde{\alpha})$ является первая координата набора $\tilde{\alpha}$, поэтому $\tilde{\alpha} = (1, \alpha_2, \ldots)$. Аналогично, $\tilde{\beta} = (1, \beta_2, \ldots)$. Рассмотрим набор

$\tilde\gamma = \tilde\alpha \,\&\, \tilde\beta$, в котором каждая координата равна конъюнкции соответствующих координат из наборов $\tilde\alpha$ и $\tilde\beta$. Тогда $\tilde\gamma = (1, \alpha_2\beta_2, \ldots)$.

Заметим, что для любых наборов $\tilde\alpha$ и $\tilde\beta$ выполнено $\tilde\alpha \,\&\, \tilde\beta \leqslant \tilde\alpha$ (достаточно проверить неравенство для каждой координаты), поэтому $\tilde\gamma \leqslant \tilde\alpha$ и $\tilde\gamma \leqslant \tilde\beta$. Отсюда, в силу возрастания функций f и g по всем переменным, следует: $f(\tilde\gamma) \leqslant f(\tilde\alpha) = 0$ и $g(\tilde\gamma) \leqslant g(\tilde\beta) = 0$, а значит, $f(\tilde\gamma) = g(\tilde\gamma) = 0$. Имеем $(f \vee g)(\tilde\gamma) = 0$, но, в то же время, $x_1(\tilde\gamma) = 1$. Мы получили противоречие: $f \vee g \ngeqslant x_1$.

в) Подобно п. б), докажем от противного. Допустим, $f \nleqslant h$ и $g \nleqslant h$. Тогда существуют наборы $\tilde\alpha$ и $\tilde\beta$ такие, что $f(\tilde\alpha) > h(\tilde\alpha)$ и $g(\tilde\beta) > h(\tilde\beta)$, т. е. $f(\tilde\alpha) = g(\tilde\beta) = 1$, $h(\tilde\alpha) = h(\tilde\beta) = 0$. Рассмотрим набор $\tilde\gamma = \tilde\alpha \vee \tilde\beta$. Тогда $\tilde\alpha \leqslant \tilde\alpha \vee \tilde\beta = \tilde\gamma$. Аналогично $\tilde\beta \leqslant \tilde\gamma$. Из монотонности функции f следует: $1 = f(\tilde\alpha) \leqslant f(\tilde\gamma)$, поэтому $f(\tilde\gamma) = 1$. Аналогично $g(\tilde\gamma) = 1$. В итоге получаем: $(fg)(\tilde\gamma) = 1$.

Чтобы прийти к противоречию, нам осталось доказать, что $h(\tilde\gamma) = 0$. Заметим, что выражение $h(\tilde\alpha)$ принимает значение 0 тогда и только тогда, когда на первых s местах набора $\tilde\alpha$ стоят нули (мы условились, что $h = x_1 \vee \ldots \vee x_s$). Поэтому $\tilde\alpha = (0, \ldots, 0, \alpha_{s+1}, \ldots)$. Аналогично, $\tilde\beta = (0, \ldots, 0, \beta_{s+1}, \ldots)$. Тогда $\tilde\gamma = (0, \ldots, 0, \alpha_{s+1} \vee \beta_{s+1}, \ldots)$, следовательно, $h(\tilde\gamma) = 0$. Противоречие.

г) Подобно п. а) имеем $f \leqslant f \vee g$, следовательно, $f \leqslant f \vee g \leqslant h$ и $g \leqslant f \vee g \leqslant h$. $\square$

Следствие 8. *Если f и g — монотонные функции, x_i — переменная, h — булева сумма и выполнено неравенство $x_i \leqslant fg \leqslant h$ (или неравенство $x_i \leqslant f \vee g \leqslant h$), то выполнено хотя бы одно из неравенств $x_i \leqslant f \leqslant h$, $x_i \leqslant g \leqslant h$.*

Действительно, если $fg \leqslant h$, то либо $f \leqslant h$, либо $g \leqslant h$. В то же время из неравенства $x_i \leqslant fg$ следует как неравенство $x_i \leqslant f$, так и $x_i \leqslant g$. Таким образом, имеет место $x_i \leqslant f \leqslant g$ или $x_i \leqslant g \leqslant h$. Рассуждение для функции $f \vee g$ аналогично. $\square$

Перейдём к доказательству нижней оценки в теореме 7. Рассмотрим произвольную схему S в базисе M_0, которая вычисляет систему булевых сумм, соответствующую матрице M. Скажем, что элемент схемы обладает свойством (x_i, h), если функция f, вычисляемая на выходе этого элемента, удовлетворяет неравенствам $x_i \leqslant$

$f \leqslant h$. *Предками* данного элемента назовём вершины (элементы или входы схемы), которые связаны рёбрами со входами данного элемента. Следствие 8 означает, что если некоторый элемент обладает свойством (x_i, h), то хотя бы один из его предков обладает этим свойством.

В качестве h возьмём булеву сумму, соответствующую какой-нибудь строке матрицы M. Для удобства будем считать, что $h = x_1 \vee \ldots \vee x_s$. Тогда элемент E, на выходе которого вычисляется функция h, обладает свойствами $(x_1, h), \ldots, (x_s, h)$. Обозначим эти свойства через $a_1, \ldots, a_s$. Часть из них передаётся первому предку элемента E, часть — второму, т. е. происходит разделение множества $\{a_1, \ldots, a_s\}$ на два (возможно, пересекающиеся). Процесс деления на две части продолжается и далее до тех пор, пока не получатся одноэлементные множества $\{a_1\}, \ldots, \{a_s\}$. Действительно, переходя от элемента к его предкам, мы рано или поздно дойдём до входов схемы, а каждый вход обладает не более, чем одним из рассматриваемых свойств. А именно, если вход соответствует одной из переменных $x_1, \ldots, x_s$, например x_1, то он обладает только свойством x_1: легко показать, что $x_2 \nleqslant x_1$ и т. д. Если же вход соответствует переменной, отличной от $x_1, \ldots, x_s$, то он не обладает ни одним из данных свойств.

Докажем ещё одно вспомогательное утверждение.

Утверждение 13. *Пусть СФЭ S удовлетворяет условиям:*

а) некоторый элемент схемы обладает свойствами $a_1, \ldots, a_s$;

б) если элемент обладает свойством a_i, то хотя бы один его предок обладает свойством a_i;

в) каждый вход схемы обладает не более чем одним свойством.

Тогда число элементов, обладающих хотя бы двумя из этих свойств, не меньше чем $s - 1$.

Доказательство проведём индукцией по s.

Базис индукции: $s = 2$. Элемент, обладающий двумя свойствами, существует согласно п. а).

Индуктивный переход: $s - 1 \mapsto s$. Элементы, обладающие хотя бы двумя свойствами, существуют согласно п. а). Среди всех таких элементов выберем элемент с наименьшим номером и обозначим его через E. Тогда E обладает ровно двумя свойствами. Действительно, у E ровно два предка и каждый предок имеет номер, меньший чем E, следовательно, обладает не более чем одним свойством. Свойства же должны передаваться предкам. Пусть E обладает свойствами a_1 и a_2. Исключим из схемы свойство a_1, т. е. будем рассматривать только свойства $a_2, \ldots, a_s$. Применим к этим свойствам предположение индукции. Получим, что существуют $(s-2)$ элемента, обладающие хотя бы двумя из свойств $a_2, \ldots, a_s$. Элемент E не попадёт в их число, так как он обладает только одним свойством: a_2. Теперь вернём свойство a_1. Каждый из найденных $(s-2)$ элементов по-прежнему обладает двумя свойствами. Добавив к этим элементам элемент E, получим требуемые $(s-1)$ элементов. $\square$

Продолжим доказательство теоремы 7. Рассмотрим какие-нибудь две булевы суммы h и h', соответствующие строкам матрицы M. Покажем, что не существует элемента, обладающего двумя свойствами (x_i, h), (x_j, h), относящимися к функции h, и ещё хотя бы одним свойством (x_l, h'), относящимся к h'. Допустим, что такой элемент существует и на его выходе вычисляется функция f. Тогда выполнены неравенства $x_i \leqslant f \leqslant h$, $x_j \leqslant f \leqslant h$ и $x_l \leqslant f \leqslant h'$. Следовательно, $x_i \leqslant f \leqslant h'$ и $x_j \leqslant f \leqslant h'$. Таким образом, $x_i \leqslant h$, $x_i \leqslant h'$, $x_i \leqslant h'$ и $x_j \leqslant h'$. Неравенство $x_i \leqslant h$ означает, что переменная x_i входит в сумму h. Аналогичное значение имеют три других неравенства. Таким образом, переменные x_i и x_j одновременно входят и в сумму h, и в сумму h'. Значит, матрица M содержит прямоугольник, образованный строками, соответствующими h и h', и столбцами, соответствующими переменным x_i и x_j. Мы получили противоречие, показывающее, что если некоторый элемент обладает двумя свойствами, относящимися к сумме h, то он не может обладать свойствами, относящимися к другой сумме h'.

Число единиц в i-й строке матрицы M обозначим через m_i. Мы показали, что i-й строке матрицы M можно поставить в соответствие не менее чем $m_i - 1$ элементов, обладающих двумя и более свойствами, связанными с булевой суммой, отвечающей этой строке. При этом множества элементов, соответствующих разным строкам, не пересекаются. Следовательно, в схеме имеется не менее

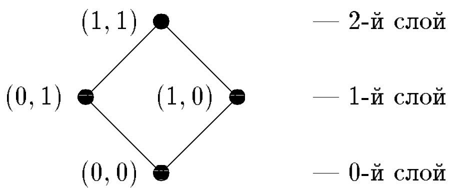

Рис. 18

чем $(m_1 - 1) + \ldots + (m_k - 1) = m - k$ различных элементов, т. е. сложность схемы не меньше $m - k$. $\square$

Упражнение 29. *Можно ли упростить схему на рис. 17, если использовать базис M_0?*

Упражнение 30. *Приведите следующие примеры, показывающие, что утверждение 12 не может быть усилено:*

1) *одна из функций f, g не монотонна и п. б) не выполнен;*

2) *f, g и h — монотонны, но h — не булева сумма и п. в) не выполнен.*

Упражнение 31*. *Докажите, что в базисе M_0 можно выразить любую монотонную функцию, не равную тождественно константе.*

§ 15. Функции первого слоя

Первым слоем называется часть области определения булевой функции, состоящая из наборов, содержащих ровно одну единицу. Например, в случае функций двух переменных первых слой состоит из наборов $(0, 1)$ и $(1, 0)$ (см. рис. 18). Обозначим через $|\tilde{\alpha}|$ число единиц в наборе $\tilde{\alpha}$. Пусть f — булева функция. Тогда функция $f^{(1)}$, определённая следующим образом:

$$f^{(1)}(\tilde{\alpha}) = \begin{cases} 0, & \text{если } |\tilde{\alpha}| = 0; \\ f(\tilde{\alpha}), & \text{если } |\tilde{\alpha}| = 1; \\ 1, & \text{если } |\tilde{\alpha}| > 1, \end{cases}$$

называется *функцией первого слоя* функции f (другие названия: *слой-функция, слайс-функция*). Таким образом, функция 1-го слоя совпадает с исходной функцией на первом слое, а на других слоях равна констате: нулю или единице.

Заметим, что если h — булева сумма, то информация о том, какая переменная входит в h, а какая — не входит, содержится на первом слое. А именно, переменная x_i входит в сумму h тогда и только тогда, когда $h(0,\ldots,0,1,0,\ldots,0)=1$ (единица в данном наборе стоит на i-м месте). Можно сказать, что первый слой содержит информацию о *линейной части* булевой функции. Например, функции $x\vee y\vee zt$ и $x\vee y$ (рассматриваемые как функции 4-х аргументов) совпадают на первом слое. Это же верно и для представления булевых функций в виде полиномов по модулю 2 (*полиномов Жегалкина*). Так, функции $x\oplus y$ и $x\oplus y\oplus zt$ совпадают на первом слое. Более того, эти функции совпадают на первом слое с функцией $x\vee y$.

Зададимся вопросом: какова причина того, что система булевых сумм $h_1,\ldots,h_n$, соответствующая матрице T_n, сложна. Возможно, ответ состоит в том, что таковы структурные свойства матрицы, а именно, отсутствие прямоугольников. Но эти структурные свойства проявляются уже на первом слое. Тогда следует ожидать, что система функций $h_1^{(1)},\ldots,h_n^{(1)}$, совпадающая с исходной на первом слое, также сложна. Цель настоящего параграфа — показать, что это не так, а именно, сложность системы функций $h_1^{(1)},\ldots,h_n^{(1)}$ в классе СФЭ в базисе B_0 линейна (в случае базиса M_0 сложность также линейна).

Для доказательства мы построим схему линейной сложности. Идея состоит в следующем: первый слой содержит очень мало наборов по сравнению со всей областью определения, а именно, всего имеется 2^n наборов, а в первый слой входят лишь n из них. Параметризуем первый слой другими переменными: $y_1,\ldots,y_k$, где $k=\lceil\log_2 n\rceil$ и будем рассматривать функции первого слоя, как функции переменных $y_1,\ldots,y_k$. Тогда сложность их вычисления будет оцениваться относительно k, например, k в некоторой степени, что обычно достаточно для алгебраических вычислений. Но такая сложность мала по сравнению с n.

Значит, основная сложность сосредоточена при переходе от переменных $x_1,\ldots,x_n$ к переменным $y_1,\ldots,y_k$ и наоборот (шифрация и дешифрация). Легко показать, что такие преобразования осуществимы с линейной сложностью. Мы видим, что для получения нелинейных нижних оценок нельзя опираться только на структурные свойства матрицы. Необходимо также учитывать, что функции, входящие в систему, отличаются друг от друга на большом числе наборов. Прежде, чем построить итоговую схему, рассмотрим несколько вспомогательных задач, представляющих самостоятельный интерес.

Сложение. На вход схемы подаются два числа, x и y, записанные в двоичной системе счисления: $x_1, \ldots, x_k$ и $y_1, \ldots, y_k$ (разряд x_1 — старший, x_k — младший). Требуется вычислить их сумму $z = x+y$ и ответ представить в двоичной системе: $z_0, z_1, \ldots, z_k$ (z_0 — самый старший разряд). Вычисление будем производить "столбиком". А именно, сложим младшие разряды x_k и y_k и получим младший разряд суммы z_k и перенос в следующий разряд, который обозначим через t_k ($t_k = 1$, если перенос есть, и $t_k = 0$, если переноса нет). Разряды z_k и t_k можно вычислить по разрядам x_k и y_k некоторой схемой, которую обозначим через Σ_1.

Далее на каждом шаге сложим очередные два разряда x_i, y_i и перенос из предыдущего разряда t_{i+1}. При этом получим очередной разряд суммы z_i и новый перенос t_i. Схему, осуществляющую вычисление разрядов z_i и t_i по известным x_i, y_i и t_{i+1}, обозначим через Σ'. Нам достаточно знать, что схемы Σ_1 и Σ' существуют; их экономная реализация обсуждается в упражнении 32. Схема, осуществляющая сложение k-разрядных чисел, изображена на рис. 19. Обозначим её через Σ_k. Сложность этой схемы линейна, а именно

$$L(\Sigma_k) = L(\Sigma_1) + (k-1)L(\Sigma') \leqslant C_1 k, \text{ где } C_1 = \max\{L(\Sigma_1), L(\Sigma')\}.$$

Коррекция по модулю p после сложения. Допустим, мы производим операции по модулю p, при этом числа представлены в виде остатков при делении на p, т. е. принадлежат отрезку $[0,\ p-1]$. Тогда после сложения двух чисел получится число из отрезка $[0,\ 2p-2]$. Необходимо вычислить остаток от его деления на p. Пусть на входе имеется число x в двоичной записи $x_0, \ldots, x_k$ ($2^{k-1} \leqslant p < 2^k$). Прибавим к x число $2^k - p$ (с помощью схемы Σ_{k+1}). Полученное число обозначим через y. Тогда $y = x + 2^k - p \leqslant 2p - 2 + 2^k - p =$

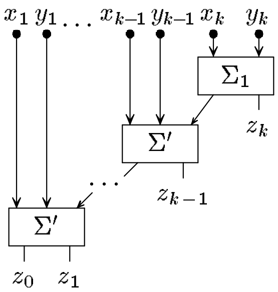

Рис. 19

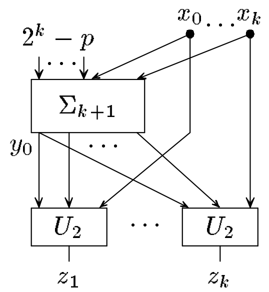

Рис. 20

$p-2+2^k < 2^{k+1}$, значит, $(k+2)$-й разряд числа y равен нулю и его можно отбросить. Останутся разряды $y_0, y_1, \ldots, y_k$.

Возможны два случая. Если $y_0 = 0$, то $y < 2^k$, следовательно, $x+2^k-p < 2^k$, т. е. $x < p$. В этом случае число x не нуждается в коррекции. Если же $y_0 = 1$, то $x \geqslant p$. В таком случае ответом является число $x - p = y - 2^k$, имеющее двоичную запись $y_1, \ldots, y_k$. Схема, осуществляющая коррекцию, изображена на рис. 20. После вычисления числа y, в зависимости от разряда y_0, на выход схемы подаётся либо число x, либо число с двоичной записью $y_1, \ldots, y_k$. Для выбора одной из двух возможностей используется универсальная функция U_2, обладающая свойствами: $U_2(0,x,y) = x$, $U_2(1,x,y) = y$. Число $2^k - p$ является константой, поэтому его зададим набором из нулей и единиц; нуль и единицу можно вычислить так: $0 = x\overline{x}$, $1 = x \vee \overline{x}$. Мы видим, что сложность построенной схемы линейна.

Сложение по модулю p состоит из сложения и коррекции после сложения. Данную схему обозначим через Σ_k^p.

Дополнение по модулю p**.** Дано число x (остаток от деления на p). Нужно вычислить $-x \pmod p$, другими словами, $p - x$. Дополнение используется при вычитании: $x - y = x + (-y)$. Вначале вычислим $y = x + (2^k - 1 - p)$. Имеем: $y \leqslant (p-1) + (2^k - 1 - p) < 2^k$, поэтому $(k+1)$-й разряд числа y равен 0. Инвертируем первые n разрядов числа y, т. е. возьмём отрицание от каждого из них. Тем самым мы вычислим число z, связанное с y равенством $z+y = 2^k-1$. Имеем: $z = 2^k - 1 - y = 2^k - 1 - x - (2^k - 1 - p) = p - x$, т. е. число z является искомым. Построенную схему обозначим через $-_p$; её сложность линейна.

Шифратор. На входы поступают разряды $x_1, \ldots, x_p$, из которых ровно один равен единице. Нужно вычислить номер этого разряда. А именно, если $x_i = 1$, то нужно выдать число $(i-1)$, записанное в двоичной системе: $y_1, \ldots, y_k$, где $2^{k-1} \leqslant p < 2^k$. Если количество единиц среди разрядов $x_1, \ldots, x_p$ отлично от 1, то значения на выходах схемы могут быть любыми.

Заметим, что каждую переменную y_i можно представить в виде булевой суммы от переменных $x_1, \ldots, x_p$. А именно, равенство $y_i = 1$ выполнено в некоторых случаях, причём каждый случай состоит в том, что некоторая переменная x_j равна единице. Явно перечислив все эти случаи, мы получим условие: $x_{j_1} = 1$ или $x_{j_2} = 1$ и т. д. Другими словами, $y_i = x_{j_1} \vee x_{j_2} \vee \ldots$ Например, $y_1 = x_{2^{k-1}+1} \vee \ldots \vee x_p$, $y_k = x_2 \vee x_4 \vee x_6 \vee \ldots$

Шифратор обозначим через CD_p. Сложность каждой булевой суммы не больше, чем $p-1$, значит, $L(CD_p) \leqslant k(p-1) < kp$. Вообще, за счёт наличия прямоугольников данную систему булевых сумм можно вычислить с линейной сложностью (см. упражнение 33).

Дешифратор решает обратную задачу: дано число y в двоичной записи: $y_1, \ldots, y_k$. Нужно выдать набор $x_1, \ldots, x_p$, в котором $x_{y+1} = 1$, а остальные значения равны нулю. Каждый разряд x_i равен единице только в одном случае, когда переменные $y_1, \ldots, y_k$ принимают определённые значения: $y_1 = \alpha_1, \ldots, y_k = \alpha_k$. Поэтому x_i равно конъюнкции переменных $y_1, \ldots, y_k$ или их отрицаний (переменная y_j берётся с отрицанием, если $\alpha_j = 0$). Для решения задачи нам достаточно вычислить систему всевозможных конъюнкций переменных $y_1, \ldots, y_k$ или их отрицаний (такие конъюнкции называются *элементарными*).

Вычисление проведём индукцией по k; сложность искомой схемы обозначим через c_k. Тогда $c_1 = 1$ (нужно вычислить функции y_1 и $\overline{y}_1$). Далее, пусть все конъюнкции от переменных $y_1, \ldots, y_{k-1}$ вычислены. Каждой такой конъюнкции K соответствуют две конъюнкции от k переменных, а именно, Ky_k и $K\overline{y}_k$. Для вычисления этих двух конъюнкций достаточно трёх элементов (два И и один НЕ). Значит, для вычисления всех конъюнкций достаточно (дополнительно к уже имеющимся) $3 \cdot 2^{k-1}$ элементов. Получим цепочку неравенств:

$$c_k \leqslant c_{k-1}+3\cdot 2^{k-1} \leqslant \ldots \leqslant c_1+3+3\cdot 2+\ldots+3\cdot 2^{k-1} = 1+3(2^k-1) \leqslant 3\cdot 2^k.$$

Мы видим, что сложность дешифратора линейна относительно p. Дешифратор обозначим через DC_p.

Перейдём к построению схемы, вычисляющей функции $h_1^{(1)}, \ldots, h_n^{(1)}$. Вначале построим схему S_1, которая вычисляет функции, совпадающие с данными на 1-м слое. Схема S_1 изображена на рис. 21. На её входы подаются значения переменных $x_1, \ldots, x_n$ (где $n = p^2$), среди которых ровно одно равно единице (другие случаи мы пока не рассматриваем). Переменные $x_1, \ldots, x_n$ соответствуют столбцам матрицы T_n (см. рис. 15), а функции $h_1^{(1)}, \ldots, h_n^{(1)}$ — строкам. Заметим, что значения функций на данном входном наборе перечислены в том столбце матрицы T_n, которому соответствует переменная, равная единице. Пусть этот столбец имеет координаты (j, l), т. е. находится в j-м блоке матрицы T_n, и является l-м столбцом в блоке (блоки и столбцы нумеруются с нуля: $0, \ldots, p-1$).

Вначале вычислим числа j и l. Для этого введём вспомогательные переменные:

$$\begin{gathered} y_1 = x_1 \vee x_2 \vee \ldots \vee x_p,\ \ldots,\ y_p = x_{p^2-p+1} \vee \ldots \vee x_{p^2}, \\ z_1 = x_1 \vee x_{p+1} \vee \ldots \vee x_{p^2-p+1},\ \ldots,\ z_p = x_p \vee \ldots \vee x_{p^2}. \end{gathered} \tag{27}$$

В каждом из наборов $(y_1, \ldots, y_p)$ и $(z_1, \ldots, z_p)$ содержится по одной единице. Положение единицы в первом наборе указывает на номер блока, а положение единицы во втором наборе — на номер столбца в блоке, которому соответствует переменная x_i, равная единице. Схема S_2 на рис. 21 вычисляет значения $y_1, \ldots, y_p$ и $z_1, \ldots, z_p$ по формулам (27). Следующие за ней шифраторы вычисляют числа j и l, записанные в двоичной системе счисления.

Каждый столбец матрицы T_n состоит из p частей, соответствующих блокам. В каждой части столбца имеется ровно одна единица. Рассмотрим столбец с координатами (j, l). Обозначим положение единицы (т. е. номер строки внутри блока) в i-й части этого столбца через k_i; здесь i и k_i отсчитываются от нуля. Справедлива формула (см. (26))

$$l - k_i \equiv ij \pmod{p},$$

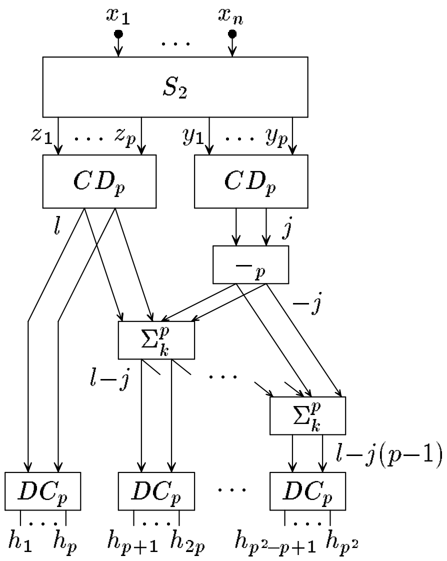

Рис. 21

откуда

$$k_i \equiv l - ij \pmod p.$$

Вычислим значения $k_0, k_1, \ldots, k_{p-1}$ последовательно: $k_0 = l$, $k_1 = k_0 + (-j) \pmod p$, $k_2 = k_1 + (-j) \pmod p$ и т. д. Для этого вначале найдём дополнение к числу j по модулю p, а затем используем цепочку сумматоров по модулю p (см. рис. 21). Наконец, для каждой части столбца мы знаем номер места, на котором стоит единица, поэтому для получения этой части столбца применим дешифратор. Например, дешифровав число k_0, получим значения функций $h_1^{(1)}, \ldots, h_p^{(1)}$; по числу k_1 вычислим значения $h_{p+1}^{(1)}, \ldots, h_{2p}^{(1)}$ и т. д.

Итак, схема S_1 построена. Оценим её сложность (здесь C_2, C_3 и C_4 — некоторые константы):

$$L(S_1) = L(S_2) + 2L(CP_p) + L(-_p) + (p-1)L(\Sigma_k^p) + pL(DC_p) \leqslant$$
$$\leqslant 2p(p-1) + 2pk + C_2k + (p-1)C_3k + 3p^2 \leqslant C_4p^2 = C_4n.$$

Мы видим, что сложность линейна. Осталось исправить значения функций на слоях, отличных от первого. Для этого по набору $x_1, \ldots,$

x_n вычислим двузначное число y_1, y_2, характеризующее количество единиц в этом наборе. Если единиц нет, то $y_1 = y_2 = 0$, если единица одна — то $y_1 = 0,\ y_2 = 1$; наконец, если число единиц не меньше двух, то $y_1 = 1,\ y_2 = 0$.

Это вычисление можно осуществить последовательно: вначале положить $y_1 = y_2 = 0$, а затем на каждом шаге по уже имеющимся значениям y_1 и y_2 и новой переменной x_i вычислить следующие значения y_1, y_2. Получится цепочка схем (подобная рис. 19), имеющая линейную сложность. Наконец, в зависимости от переменных y_1 и y_2 нужно выбрать, выдавать ли на выходы схемы нули, единицы или выдавать то, что вычислила схема S_1. Каждый выбор осуществляется с помощью функции U_2 подобно рис. 20. Нами доказана

Теорема 8. *Если $h_1, \ldots, h_n$ — булевы суммы, построенные по матрице T_n, то сложность системы функций $h_1^{(1)}, \ldots, h_n^{(1)}$ в классе СФЭ в базисе B_0 линейна.*

Упражнение 32. *Постройте схемы Σ_1 и Σ' (используемые при сложении) в базисе B_0 так, чтобы они имели сложность* 4 *и* 9 *соответственно.*

Упражнение 33. *Докажите, что сложность шифратора линейна относительно p.*

Комментарии и литература

Для дальнейшего знакомства с нижними оценками сложности лучше всего подойдёт книга Р. Г. Нигматуллина [1]. В несколько ином аспекте, с бо́льшим уклоном в сторону практики, сложность булевых функций изложена в книге Дж. Сэвиджа [2], переведённой на русский язык. Читателю, владеющему английским языком, можно рекомендовать книгу И. Вегенера [3], пользующуюся больши́м авторитетом за рубежом (эта книга доступна в Интернете). Разнообразная информация по проблеме нижних оценок сложности собрана на Интернет-сайте [4].

Первые результаты о сложности линейной функции (§§ 4, 5) получили К. Шеннон в соавторстве с Дж. Риорданом [5] и С. В. Яблонский [6]. Нижнюю оценку $n^{3/2}$ (теорема 2) доказала Б. А. Субботовская [7]. Метод получения нижних оценок сложности с помощью подстановки констант вместо переменных называется методом Субботовской. Нижнюю оценку n^2 и метод оценки сложности произвольных функций (теорема 3) доказал В. М. Храпченко [8]. Изложенное нами доказательство метода Храпченко дал Патерсон (см. [3]).

Общий приём доказательства существования достаточно сложных функций (леммы 6, 7) предложил Шеннон. Идея использовать универсальные функцию для получения нижних оценок принадлежит Э. И. Нечипоруку [9]. Более простой вариант метода Нечипорука, изложенный нами (глава III), содержится, например, в неопубликованном обзоре [10]. Нечипорук также указал матрицу без прямоугольников (Глава IV) и применил её к получению нижних оценкок сложности [11]. Слой-функции ввёл Берковиц (см. [3]).

Литература

[1] Нигматуллин Р. Г. Сложность булевых функций. — М.: Наука, 1991.

[2] Сэвидж Дж. Э. Сложность вычислений. — М.: Из-во "Факториал", 1998.

[3] Wegener I. The Complexity of Boolean Functions. — Teubner, Wiley, 1987. — http://www.eccc.uni-trier.de/eccc-local/ECCC-Books/wegener_book_readme.html.

[4] Нижние оценки сложности булевых функций. — http://boolba.narod.ru.

[5] Шеннон К. Работы по теории информации и кибернетике. — М.: ИЛ, 1963.

[6] Яблонский С. В. Реализация линейной функции в классе П-схем // ДАН СССР. — 1954. — Т. 94, 5. — С. 805–806.

[7] Субботовская Б. А. О реализации линейных функций формулами в базисе $\vee$, &, $-$ // ДАН СССР. — 1961. — Т. 136, 3. — С. 784–787.

[8] Храпченко В. М. Об одном методе получения нижних оценок сложности П-схем // Матем. заметки. — 1972. — Т. 10, 1. — С. 83–92.

[9] Нечипорук Э. И. Об одной булевской функции // ДАН СССР. — 1966. — Т. 169, 4. — С. 765–766.

[10] Кричевский Р. Е, Нечипорук Э. И., Храпченко В. М. Нелинейные нижние оценки сложности реализации булевых функций.

[11] Нечипорук Э. И. Об одной булевской матрице // Пробл. кибернетики. Вып. 21. — М.: Наука, 1969. — С. 237–240.