

# Этюды о методах синтеза быстрых схем

И. С. Сергеев<sup>1</sup>

предварительная версия 6 от 8 ноября 2024 г.

(добавлены пункты 9.3, 11.4, исправлены опечатки)

<sup>1</sup>e-mail: isserg@gmail.com

## Предисловие

Быстрые вычисления — одна из наиболее значимых областей приложения усилий математиков. В современном мире — царстве разнообразных форм электроники и искусственного интеллекта — практически все сферы жизни пронизаны плодами этой теории.

Целью настоящих заметок является систематизация наиболее плодотворных идей, ведущих к быстрым методам вычислений. Изложение построено вокруг вычислительной модели схем из функциональных элементов, а также ее разновидности — формул. Это сделано сознательно: с одной стороны, чтобы не перегружать понятийный аппарат, с другой — чтобы не пытаться объять необъятное, пусть это и ограничивает выбор приложений для иллюстрации идей.

Приводимые примеры охватывают преимущественно булевы и арифметические (алгебраические) вычисления как наиболее практически востребованные и интенсивно исследуемые теорией сложности. Стоит отметить, что популярные, ставшие классическими монографии о сложности булевых функций [330, 184, 59, 231] написаны с акцентом на нижние оценки сложности. Способам быстрых вычислений в них отводится второстепенная роль — в основном, для демонстрации точности нижних оценок. Настоящая работа отчасти восполняет этот пробел.

Алгебраическим алгоритмам, как правило, имеющим непосредственный выход на приложения, уделяется больше внимания. Пожалуй, наиболее полно современная теория быстрых алгебраических алгоритмов изложена в [200], повышенное внимание вычислениям с матрицами и многочленами уделяется в [145, 275]. Хорошим введением в теорию численных алгоритмов служат книги [24, 160]. Настоящая работа включает только несколько фрагментов теории, демонстрирующих разнообразие вычислительных приемов.

Конечно, выбор сюжетов для книги был продиктован вкусами автора. Помимо широко известных классических методов синтеза в ней рассказывается о результатах последних 10–20 лет. Автор надеется, что в нескольких случаях ему удалось предложить более прозрачные доказательства, чем в оригинальных работах и прочих источниках. Теория быстрых вычислений продолжает развиваться, хотя каждый следующий шаг дается все труднее.

*Игорь Сергеев  
Сходня/Москва,  
апрель 2022 г.*

# Оглавление

<b>Основные понятия, принятые соглашения и обозначения</b>	<b>6</b>
<b>1 Последовательный метод</b>	<b>11</b>
Схемы для линейной функции 11. Стандартная схема сумматора 12. Наибольший общий делитель (бинарный алгоритм) 13. Формулы для натуральных чисел (метод Зелински) 14. Алгоритмы динамического программирования для функции проводимости и гамильтониана 17. Монотонные схемы для многочлена Кирхгофа (метод Фомина—Григорьева—Кошевого) 19.	
<b>2 Деление пополам</b>	<b>22</b>
Сложность формул для линейной функции 22. Умножение чисел (метод Карапубы) 23. Умножение матриц (метод Штрассена) 24. Быстрое преобразование Фурье 25. Параллельные префиксные схемы (метод Ладнера—Фишера) 26. Параллельные схемы сумматоров (метод Храпченко) 29. Другие приложения (переход между системами счисления, быстрое вычисление наибольшего общего делителя, сортировка и монотонные схемы для пороговых функций, мультиплексорная сложность многочленов) 30.	
<b>3 Метод общей части</b>	<b>34</b>
Аддитивные цепочки (метод Брауэра) 34. Асимптотически минимальные схемы для булевых функций (метод Лупанова) 35. Схемы для линейных булевых операторов (метод Нечипорука) 37. Сложность монотонных схем (принцип локального кодирования) 39. Сложность схем для мультиплексорной функции (метод Клейна—Патерсона) 41.	
<b>4 Метод потенциалов</b>	<b>43</b>
Глубина схем для сложения по модулю 3 (метод Чина) 43. Формульная сложность линейной функции в тернарном базисе (метод Чоклер—Цвика) 44. Глубина схем для многократного сложения (метод Патерсона—Пиппенджера—Цвика) 45. Параллельное перестроение арифметических формул (методы Брента—Кука—Маруямы и Препараты—Маллера) 49.	
<b>5 Метод приближений</b>	<b>53</b>
Быстрое деление чисел (метод Кука) 53. Быстрое деление с остатком комплексных многочленов (метод ван дер Хувена) 54. Умножение матриц (границ-	

ный ранг) 57. Монотонные схемы для сортировки (AKS-метод) 60. Другие приложения (быстрое вычисление логарифма и экспоненты) 68.

## **6 Алгебраический метод 70**

Умножение булевых матриц 70. Сложность формул для сложения по модулю 7 (метод ван Лейенхорста) 71. Умножение чисел при помощи ДПФ (метод Шёнхаге—Штассена) 72. Параллельные схемы деления чисел (метод Бима—Кука—Гувера) 74. Модулярная композиция многочленов (метод Уманса) 76. Другие приложения (умножение в полях Мерсенна, арифметика в нормальных базисах конечных полей) 80.

## **7 Специальная кодировка 83**

Схемная сложность суммирования битов 83. Вещественная сложность комплексного ДПФ (метод ван Бускирка) 85. Умножение матриц (ускорение метода Штассена) 88. Сложность формул для сложения по модулю 5 (метод Сергеева) 89. Быстрое возведение многочленов в степень (метод Монтгомери) 91. Другие приложения (формулы для симметрических булевых функций, почти монотонная сложность булевых функций, интерполяция и вычисление значений многочлена в точках арифметической прогрессии) 92.

## **8 Принципы двойственности 95**

Сложность универсальной матрицы (принцип транспонирования) 95. Параллельные схемы сумматоров (метод Гринчука) 96. Умножение прямоугольных и квадратных матриц (трилинейное тождество Пана) 99. Сложность рациональной функции и ее градиента (метод Баура—Штассена) 100.

## **9 Вероятностный метод 105**

Монотонные формулы для симметрических пороговых функций (метод Хасина) 105. Монотонные формулы для функции голосования (метод Вэльянта) 106. Параллельные схемы для логического перманента 109. Сложность линейных операторов с плотными матрицами 112.

## **10 Метод массового производства 115**

Групповые линейные преобразования 115. Вычисление булевой функции на нескольких входных наборах (метод Улига) 116. Умножение матриц (метод прямых сумм Шёнхаге) 117. Умножение матриц (лазерный метод Штассена) 121. Другие приложения (монотонные схемы для слой-функций) 123.

## **11 Комбинаторные методы 124**

Монотонные схемы для симметрической функции с порогом 2 (метод Адлемана) 124. Асимптотическая сложность формул (метод сфер Лупанова) 125. Мультиплективная сложность многочленов (метод Ловетта) 127. Сложность схем для логического перманента 128. Монотонная сложность многочлена циклических блужданий 132.

## **12 Разное 136**

Мультиликативная сложность булевых функций (метод ортогональных систем Нечипорука) 136. Глубина булевых функций (метод Ложкина балансировки деревьев) 137. Глубина схем для многократного сложения (градиентный метод) 139.

**13 Дополнение. Схемы ограниченной глубины 142**

Линейные схемы глубины 2 для матрицы Серпинского (градиентный метод) 143.  $\oplus\wedge\oplus$ -схемы для булевых функций (метод Селезневой) 146.  $AC[\oplus]$ -схемы глубины 4 для функции голосования (метод Оливейры—Сантанама—Сринивасана) 149. Линейные схемы для матрицы Сильвестра (метод сечения) 152. Перестроение арифметических схем в  $\Sigma\Pi\Sigma$ -схемы (метод сечения) 154. Перестроение арифметических схем в  $\Sigma\Pi\Sigma$ -схемы 157.

**Литература 161**

## Основные понятия

Базовая вычислительная модель, которую мы рассматриваем, — это *схемы из функциональных элементов* (далее, просто схемы). *Схема над базисом* (множеством функций)  $\mathcal{B}$  — это ориентированный граф без ориентированных циклов, в котором вершины, не имеющие входящих ребер, отмечены как входы, а вершины, не имеющие исходящих ребер, отмечены как выходы.

Входам приписаны символы переменных или констант базиса  $\mathcal{B}$ , остальным вершинам<sup>1)</sup> — символы функций из базиса  $\mathcal{B}$ . Функционирование схемы определяется естественным образом, от входов к выходам: в каждой вершине вычисляется сопоставленная ей функция, аргументами которой служат функции, поступающие по входящим в вершину ребрам.

Схема *реализует* оператор  $F$ , если на выходах схемы вычисляются все компоненты оператора. На рис. 1а) изображена схема, вычисляющая арифметическую сумму трех бит  $2y_2 + y_1 = x_1 + x_2 + x_3$  по правилам<sup>2)</sup>  $y_1 = x_1 \oplus x_2 \oplus x_3$ ,  $y_2 = x_1(x_2 \oplus x_3) \oplus x_2x_3$ . В наиболее распространенной ситуации схема имеет один выход и реализует некоторую функцию.

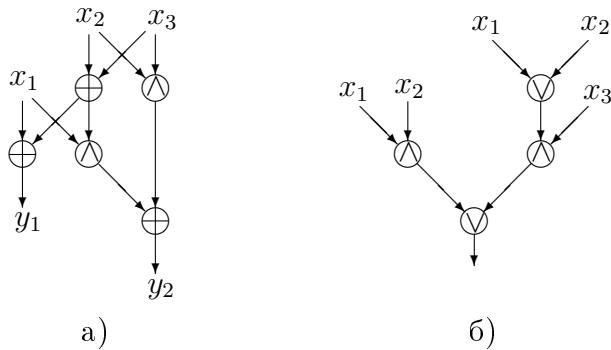


Рис. 1: Пример схемы (а) и формулы (б)

*Сложность схемы* определяется как число вершин в ее графе без учета входов. *Сложность оператора*  $F$  при реализации схемами над базисом  $\mathcal{B}$  определяется как сложность минимальной реализующей его схемы и обозначается через  $C_{\mathcal{B}}(F)$ . *Глубина схемы* — это длина (измеряемая в ребрах или функциональных элементах) максимального ориентированного пути, соединяющего вход и выход схемы. По аналогии, *глубина оператора*  $F$  определяется как минимальная глубина реализующей его схемы, и обозначается как  $D_{\mathcal{B}}(F)$ . Сложность и глубина класса (т.е. множества) операторов  $\mathcal{F}$  определяются как  $C_{\mathcal{B}}(\mathcal{F}) = \max_{F \in \mathcal{F}} C_{\mathcal{B}}(F)$  и  $D_{\mathcal{B}}(\mathcal{F}) = \max_{F \in \mathcal{F}} D_{\mathcal{B}}(F)$ .

Известно, что сложность вычисления оператора в любом полном конечном булевом базисе — одна и та же с точностью до порядка.

<sup>1</sup>Эти вершины называются *функциональными элементами*.

<sup>2</sup>Здесь и далее символ конъюнкции будет опускаться, как это принято для мультипликативных операций.

Схемы над базисом из единственной полугрупповой операции  $\{+\}$  называют-ся *аддитивными схемами*. Ввиду особой роли аддитивных схем в теории синтеза для сложности реализации линейного оператора  $L_A$  с матрицей  $A$  такими схемами мы используем специальное обозначение  $L_+(A)$  вместо  $C_{\{+\}}(L_A)$ ; при этом удобно говорить, что схема вычисляет саму матрицу  $A$ . Пусть также  $L(A)$  означает сложность универсальной аддитивной схемы, т.е. правильно вычисляющей матрицу  $A$  в любой коммутативной полугруппе.

*Формула* — это частный случай схемы, в котором запрещены ветвления вершин: из любой вершины графа исходит не более одного ребра. Под *сложностью формулы* принято понимать число входов переменных<sup>3)</sup>. На рис. 1б) изображен пример формулы, вычисляющей функцию голосования  $\text{maj}_3(x_1, x_2, x_3) = (x_1 \vee x_2)x_3 \vee x_1x_2$ .

Формулу можно также интерпретировать как выражение, которое можно записать в одну строчку (отсюда — название термина). Для такой интерпретации лучше подойдет следующее индуктивное формальное определение. *Формула над базисом  $\mathcal{B}$ , сложность формулы, глубина формулы и функция, реализуемая формулой*, определяются следующим образом: 0) константы базиса являются формулами сложности и глубины 0; 1) символы переменных являются формулами сложности 1, глубины 0 и реализуют соответствующие тождественные функции; 2) выражение  $G(F_1, \dots, F_k)$ , где  $G$  — символ, обозначающий отличную от константы  $k$ -местную функцию  $g \in \mathcal{B}$ , а  $F_i$  — формула сложности  $L_i$  и глубины  $D_i$ , реализующая функцию  $f_i$ , является формулой сложности  $L_1 + \dots + L_k$ , глубины  $\max\{D_1, \dots, D_k\} + 1$  и реализует функцию  $g(f_1, \dots, f_k)$ . В бинарном случае ( $k = 2$ ) для записи формулы принято использовать символы бинарных операций: вместо  $G(F_1, F_2)$  пишут  $F_1 \circ F_2$ , где  $g = x \circ y$ .

В указанном определении  $F_1, \dots, F_k$  называются *главными подформулами* формулы  $G(F_1, \dots, F_k)$ .

*Сложность оператора  $F$*  при реализации формулами над базисом  $\mathcal{B}$  (не зависит от способа определения) будем обозначать через  $\Phi_{\mathcal{B}}(F)$ . По аналогии со схемами вводится обозначение  $\Phi_{\mathcal{B}}(\mathcal{F})$  для формульной сложности класса операторов  $\mathcal{F}$ . Глубина реализации любого оператора схемами и формулами над одним и тем же базисом совпадает, поэтому мы используем единое обозначение  $D_{\mathcal{B}}$ . При исследовании глубины вычислений часто удобно рассматривать именно формулы как топологически более простой объект по сравнению со схемами.

Важный подкласс формул образуют *бесповторные формулы*. Бесповторной называется формула, в которой символ каждой переменной встречается не более одного раза. Функции, реализуемые бесповторными формулами над базисом  $\mathcal{B}$ , также называются бесповторными или, если точнее, бесповторно выражимыми в базисе  $\mathcal{B}$ .

Наиболее популярные булевы базисы: стандартный базис  $\mathcal{B}_0 = \{\vee, \wedge, \neg\}$ , базис Жегалкина  $\mathcal{B}_1 = \{\oplus, \wedge, 1\}$ , монотонный базис  $\mathcal{B}_M = \{\vee, \wedge\}$ , базис  $\mathcal{B}_2$  из всех двуместных булевых функций (бинарный базис), унарный базис  $\mathcal{U}_2 = \mathcal{B}_2 \setminus \{\oplus, \sim\}$ . Здесь « $\sim$ » означает булеву операцию эквивалентности.

---

<sup>3)</sup>Но это не очень принципиально, поскольку в формуле над конечным базисом число входов и число функциональных элементов — величины одного порядка.

Основные арифметические базисы для вычислений над полукольцом  $R$ : полный базис  $\mathcal{A}^R = \{+, -, *\} \cup \{ax \mid a \in R\}$ , линейный базис  $\mathcal{A}_L^R = \{+, -\} \cup \{ax \mid a \in R\}$ , монотонный базис  $\mathcal{A}_+^R = \{+, *\}$ , полный базис с делением  $\mathcal{A}_D^R = \{+, -, *, /\} \cup \{ax \mid a \in R\}$  (в случае базиса с делением предполагается, что  $R$  — кольцо).

### Принятые соглашения

Для упрощения записи выражений мы будем опускать обозначения базиса в функционалах сложности (скажем, писать  $C(F)$  вместо  $C_{\mathcal{B}}(F)$ ) в тех случаях, когда используемый базис понятен из контекста, например, внутри доказательства утверждений.

Аналогично, в обозначениях арифметических базисов и операторов мы будем опускать указания на полукольцо  $R$ , над которым производятся вычисления (например, писать  $\mathcal{A}$  вместо  $\mathcal{A}^R$  или  $M_n$  вместо  $M_n^R$ ), когда это понятно из контекста.

Для сравнения порядков роста неотрицательных функций мы используем стандартные обозначения:  $f \prec g$  равносильно  $f = o(g)$ ;  $f \preccurlyeq g$  равносильно  $f = O(g)$ ;  $f \succ g$  равносильно  $f = \omega(g)$ ;  $f \succcurlyeq g$  равносильно  $f = \Omega(g)$ ;  $f \asymp g$  равносильно  $f = \Theta(g)$ ;  $f \sim g$ ,  $f \lesssim g$ ,  $f \gtrsim g$  означают асимптотические равенство и неравенства.

Далее  $X$  (аналогично  $Y, Z$ ) как правило будет означать набор переменных  $x_i$ , возможно организованных в виде матрицы  $(x_{i,j})$ .

### Обозначения

$\mathbb{B}$  — булево множество  $\{0, 1\}$

$\mathbb{N}$  — множество натуральных чисел  $1, 2, 3, \dots$

$\mathbb{N}_0$  — множество неотрицательных целых чисел  $0, 1, 2, \dots$

$\mathbb{P}$  — множество простых чисел

$\llbracket n \rrbracket$  — множество  $\{0, 1, \dots, n - 1\}$

$C_n^k$  — число сочетаний из  $n$  по  $k$

$\mathcal{P}^n$  — класс булевых функций  $n$  переменных

$\mathcal{M}^n$  — класс монотонных булевых функций  $n$  переменных

$\mathcal{S}^n$  — класс симметрических булевых функций  $n$  переменных

$\mathbf{P}(A)$  — вероятность события  $A$

$\mathbf{E}[A]$  — математическое ожидание события  $A$

$\|X\|$  — вес булева вектора (число единиц)

$|A|$  — вес матрицы  $A$  (число ненулевых элементов)

$\det A$  — определитель матрицы  $A$

$u \odot v$  — покомпонентное произведение векторов  $u$  и  $v$

$A \otimes B$  — кронекерово произведение матриц  $A$  и  $B$  [59](#)

$T_1 \otimes T_2$  — тензорное произведение систем билинейных форм  $T_1$  и  $T_2$  [59](#)

$T_1 \oplus T_2$  — прямая сумма систем билинейных форм  $T_1$  и  $T_2$  [117](#)

$C_{\mathcal{B}}(F)$  — сложность реализации оператора  $F$  схемами над базисом  $\mathcal{B}$  [6](#)

$\Phi_{\mathcal{B}}(F)$  — сложность реализации оператора  $F$  формулами над базисом  $\mathcal{B}$  [7](#)

$D_{\mathcal{B}}(F)$  — глубина реализации оператора  $F$  в базисе  $\mathcal{B}$  [6](#)

$C(F), \Phi(F)$  — сложность оператора  $F$  при реализации схемами/формулами в произвольном полном булевом базисе 8

$L_+(A)$  — сложность реализации матрицы  $A$  аддитивными схемами в базисе  $\{+\}$  7

$L(A)$  — сложность универсальной аддитивной схемы для матрицы  $A$  7

$C_d^{AC}(F)$  — сложность реализации оператора  $F$   $AC$ -схемами глубины  $d$  142

$C_d^{AC[\oplus]}(F)$  — сложность реализации оператора  $F$   $AC[\oplus]$ -схемами глубины  $d$  142

$W_d^+(A)$  — сложность реализации матрицы  $A$  линейными схемами глубины  $d$  в базисе  $\{+\}$  143

$W_d(A)$  — сложность универсальной линейной схемы глубины  $d$  для матрицы  $A$  143

$D_A(n)$  — минимальная глубина дерева на  $n$  входах из компрессоров  $A$  47

$\mathcal{B}_2$  — базис из всех двуместных булевых функций (бинарный булев базис) 7

$\mathcal{B}_0$  — стандартный булев базис  $\{\vee, \wedge, \neg\}$  7

$\mathcal{B}_1$  — базис Жегалкина  $\{\oplus, \wedge, 1\}$  7

$\mathcal{B}_M$  — монотонный булев базис  $\{\vee, \wedge\}$  7

$\mathcal{B}_3$  — базис  $\{\text{maj}_3(x, y, z), \bar{x}, 1\}$  44

$\mathcal{U}_2$  — унарный базис двуместных функций  $\mathcal{B}_2 \setminus \{\oplus, \sim\}$  7

$\mathcal{U}_k$  — унарный базис  $k$ -местных функций 44

$\mathcal{A}^R$  — полный арифметический базис  $\{+, -, *\} \cup \{ax | a \in R\}$  над полукольцом  $R$  8

$\mathcal{A}_L^R$  — линейный арифметический базис  $\{+, -\} \cup \{ax | a \in R\}$  8

$\mathcal{A}_+^R$  — монотонный арифметический базис  $\{+, *\}$  8

$\mathcal{A}_D^R$  — арифметический базис с делением  $\{+, -, *, /\} \cup \{ax | a \in R\}$  над кольцом  $R$  8

$\mathcal{A}_{D+}^R$  — монотонный арифметический базис с делением  $\{+, *, /\}$  над кольцом  $R$  19

$\text{AGC}(a, b)$  — арифметико-геометрическое среднее чисел  $a$  и  $b$  68

$c_{\mathcal{B}}$  — константа равномерности базиса  $\mathcal{B}$  49

$\mathcal{C}_k$  — цикл длины  $k$  в графе 132

$\text{mon } f$  — множество мономов многочлена  $f$  154

$\text{rk}^R T$  — ранг системы билинейных форм  $T$  над полукольцом  $R$  57

$\underline{\text{rk}}^R T$  — граничный ранг системы билинейных форм  $T$  над полукольцом  $R$  58

$\text{tw}(G)$  — древесная ширина графа  $G$  132

$\text{CONN}_n(X)$  — функция  $(s, t)$ -проводимости  $n$ -вершинного графа 17

$CW_{k,n}(X)$  — многочлен циклических блужданий длины  $k$  в графе на  $n$  вершинах 132

$D_n^R$  — оператор деления многочленов из  $R[x]$  по модулю  $x^n$  55

$\Delta\Phi_{N,\zeta}[R]$  — дискретное преобразование Фурье порядка  $N$  с примитивным корнем  $\zeta$  в кольце  $R$  25

$HAM_n(X)$  — гамильтониан порядка  $n$  18

$\text{Hom}_{G,n}(X)$  — многочлен гомоморфных отображений графа  $G$  на полный  $n$ -вершинный граф 132

$I_n(x)$  — оператор инвертирования  $n$ -разрядного числа  $x \in [1/2, 1]$  с точностью  $2^{-n}$  [53](#)

$L_A$  — линейный оператор с матрицей  $A$  [7](#)

$\Lambda_n$  — линейная булева функция  $n$  переменных,  $x_1 \oplus \dots \oplus x_n$  [11](#)

$\text{maj}_n$  — функция голосования  $n$  булевых переменных [106](#)

$M_n, M_n^R$  — операторы умножения  $n$ -разрядных двоичных чисел и многочленов степени  $< n$  над полукольцом  $R$  [23](#)

$M(n), M^R(n)$  — сглаженные функции сложности операторов  $M_n$  и  $M_n^R$  [30](#)

$MC_n^R$  — оператор модулярной композиции двух многочленов  $f, g \in R[x]$  степени  $< n$  по модулю многочлена  $h$  степени  $n$ ,  $f(g(x)) \bmod h(x)$  [76](#)

$MM_n^R$  — оператор умножения  $n \times n$  матриц над полукольцом  $R$  [24](#)

$MM_{m,n,p}^R$  — оператор умножения матриц размера  $m \times n$  и  $n \times p$  над  $R$  [59](#)

$\text{MOD}_n^m$  — оператор сложения  $n$  переменных по модулю  $m$  [43](#)

$\text{MOD}_n^{m,r}$  — индикаторная функция равенства суммы  $n$  переменных числу  $r$  по модулю  $m$ , [43](#)

$\mu_n(X; Y)$  — мультиплексорная функция порядка  $n$  от  $2^n$  информационных переменных  $y_i$ : принимает значение  $y_X$  [41](#)

$\text{НОД}_n(a, b)$  (или  $\text{НОД}(a, b)$ ) — наибольший общий делитель  $n$ -разрядных чисел или многочленов  $a$  и  $b$  степени  $< n$  [13](#) [30](#)

$QR_{n,m}, QR_{n,m}^R$  — операторы деления с остатком:  $n$ -разрядного числа на  $m$ -разрядное и многочлен степени  $< n$  на многочлен степени  $m$  над  $R$  [30](#) [55](#)

$PERM_n(X)$  — логический перманент порядка  $n$  [109](#)

$\Sigma_n$  — оператор сложения  $n$ -разрядных двоичных чисел [12](#)

$\Sigma_{m,n}$  — оператор сложения  $m$  штук  $n$ -разрядных двоичных чисел [48](#)

$SH_n(v, x)$  — оператор сдвига  $n$ -разрядного числа  $x$  на  $v$  позиций влево [14](#)

$\text{СЛ}_{m,n}$  — оператор слияния упорядоченных наборов длины  $m$  и  $n$  [32](#)

$SORT_n$  — оператор сортировки набора длины  $n$  [31](#)

$ST_G(X)$  — многочлен Кирхгофа (многочлен остовых деревьев) графа  $G$  [19](#)

$T_{n,b}(x)$  — оператор преобразования числа  $x < 2^n$  из  $b$ -ичного представления в двоичное [30](#)

$T_n^k$  — монотонная симметрическая булева функция  $n$  переменных с порогом  $k$ ,  $T_n^k = (x_1 + \dots + x_n \geq k)$  [31](#)

▷ Доказательство леммы или следствия. □

▶ Доказательство теоремы. ■

● Дополнительная информация.

# Глава 1

## Последовательный метод

$\boxed{s}$

На самом деле, метод не имеет специального общепринятого названия. Но это самый простой путь вычислений, который сразу приходит на ум — попробовать свести задачу размера  $n$  к задаче размера  $n - 1$ .

### Схемы для линейной функции $\boxed{s}$

Линейную булеву функцию  $n$  переменных  $\Lambda_n(x_1, \dots, x_n) = x_1 \oplus x_2 \oplus \dots \oplus x_n$  легко вычислить, руководствуясь правилом

$$\Lambda_n = x_n \overline{\Lambda_{n-1}} \vee \bar{x}_n \Lambda_{n-1}, \quad \text{или} \quad \Lambda_n = x_n \overline{\Lambda_{n-1}} \vee (\bar{x}_n \vee \overline{\Lambda_{n-1}}), \quad (1.1)$$

где  $\Lambda_{n-1}$  — линейная функция переменных  $x_1, \dots, x_{n-1}$ .

**Теорема 1.1.**  $C_{U_2}(\Lambda_n) \leq 3n - 3$ ,  $C_{B_0}(\Lambda_n) \leq 4n - 4$ .

► Формулы (1.1) приводят к соотношениям  $C_{U_2}(\Lambda_n) \leq C_{U_2}(\Lambda_{n-1}) + 3$  и  $C_{B_0}(\Lambda_n) \leq C_{B_0}(\Lambda_{n-1}) + 4$ . Остается заметить, что также верно  $C_{B_0}(\bar{\Lambda}_n) \leq C_{B_0}(\Lambda_{n-1}) + 4$ , поскольку формулы (1.1) остаются в силе при инверсии входящих в них линейных функций. Соответствующие схемы изображены на рис. 1.1<sup>1</sup>). ■

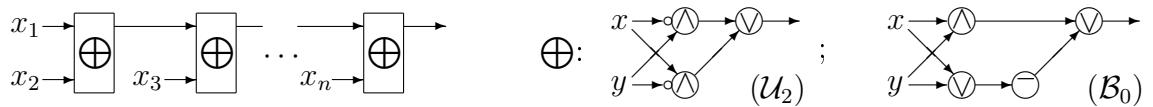


Рис. 1.1: Схемы для линейной функции

Простой метод вычисления на самом деле оказывается оптимальным. При этом доказательство нижних оценок двойственны доказательству верхних. Соответствующий способ рассуждения в теории нижних оценок сложности называется методом подстановки констант или методом исключения элементов. Приведем

<sup>1</sup> Схема в базисе  $B_0$  на рис. 1.1 реализует линейную функцию или ее отрицание в зависимости от четности  $n$ .

простой пример в качестве иллюстрации. Следующий результат принадлежит К. Шнорру [299].

**Теорема 1.2** ([299]).  $C_{\mathcal{U}_2}(\Lambda_n) \geq 3n - 3$ .

► Пусть  $n \geq 2$ . Рассмотрим произвольную минимальную схему, вычисляющую линейную функцию  $f = \Lambda_n \oplus \sigma$ , где  $\sigma \in \mathbb{B}$ . К некоторому элементу  $e_1$  схемы присоединены входы двух разных переменных, обозначим их  $x$  и  $y$ . По свойству функций базиса  $U_2$ , существуют константы  $\alpha, \beta \in \mathbb{B}$ , такие, что как при подстановке  $x = \alpha$ , так и при  $y = \beta$  выход элемента  $e_1$  обращается в константу. Заметим, что по крайней мере одна из переменных  $x, y$  (на самом деле, обе), скажем,  $x$ , присоединяется еще к какому-то элементу  $e_2$  (в противном случае подстановка  $x = \alpha$  устранила бы зависимость функции  $f$  от  $y$ , и наоборот).

Тогда при подстановке  $x = \alpha$  схема упрощается: из нее можно удалить, по меньшей мере, элементы  $e_1, e_2$  и некоторый элемент  $e_3$ , к которому присоединялся выход элемента  $e_1$ . Новая схема реализует линейную функцию  $n - 1$  переменных. Получаем  $C_{\mathcal{U}_2}(f) \geq \min\{C_{\mathcal{U}_2}(\Lambda_{n-1}), C_{\mathcal{U}_2}(\overline{\Lambda_{n-1}})\}$ , откуда немедленно следует требуемая оценка. ■

- Точность оценки теоремы 1.1 для базиса  $\mathcal{B}_0$  доказал Н. П. Редькин в [67] также методом подстановки констант, но доказательство требует рассмотрения нескольких случаев. Более того, Редькин [70] доказал, что в любом полном булевом базисе  $\mathcal{B}$  при  $n \geq 2$  выполнено  $C_{\mathcal{B}}(\Lambda_n) \leq 7(n - 1)$ , причем эта оценка достигается, например, в базисе  $\mathcal{B} = \{\wedge, \neg\}$ .

### Стандартная схема сумматора $\boxed{s}$

Через  $\Sigma_n$  будем обозначать булев  $(2n, n + 1)$ -оператор сложения  $n$ -разрядных чисел:  $\Sigma_n(A, B) = A + B$ . Пусть в двоичной записи

$$A = [a_{n-1}, a_{n-2}, \dots, a_0], \quad B = [b_{n-1}, b_{n-2}, \dots, b_0], \quad A + B = [z_n, z_{n-1}, \dots, z_0].$$

**Теорема 1.3.**  $C_{\mathcal{B}_2}(\Sigma_n) \leq 5n - 3$ .

► Результат дает схема, реализующая простой школьный метод сложения. Ее структура изображена на рис. 1.2.

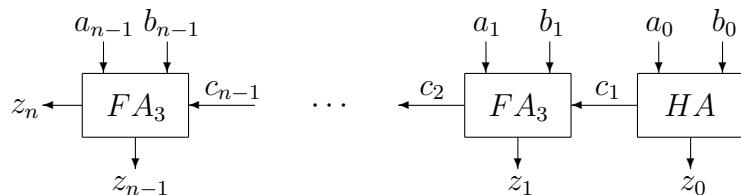


Рис. 1.2: Стандартная схема сумматора

Последовательно двигаясь от младших разрядов к старшим, всякий раз значение очередного разряда суммы и переноса в следующий разряд вычисляется

схемой сумматора трех битов, обозначаемой  $FA_3$  (сокращение от full adder), по формулам

$$x_i = a_i \oplus b_i, \quad y_i = a_i b_i, \quad z_i = x_i \oplus c_i, \quad c_{i+1} = y_i \oplus x_i c_i \quad (1.2)$$

со сложностью 5, см. рис. 1а). В младшем разряде переноса нет, и сложение выполняется более простой схемой  $HA$  (half adder — полусумматор):  $z_0 = x_0 = a_0 \oplus b_0$ ,  $c_1 = y_0 = a_0 b_0$ . ■

- Н. П. Редькин в работе [69] показал, что построенная схема минимальна, т.е.  $C_{B_2}(\Sigma_n) = 5n - 3$ . Это доказательство гораздо сложнее, чем для схемной сложности линейной функции.

Если базис не содержит линейных функций (случай  $\mathcal{B}_0$  или  $\mathcal{U}_2$ ), то переносы удобнее вычислять по формулам<sup>2)</sup>

$$c_{i+1} = y_i \vee x_i c_i, \quad x_i = a_i \vee b_i, \quad y_i = a_i b_i. \quad (1.3)$$

Например, так можно получить оценки  $C_{\mathcal{U}_2}(\Sigma_n) \leq 7n - 4$  и  $C_{\mathcal{B}_0}(\Sigma_n) \leq 9n - 5$ .

Аналогично строятся схемы для вычитания.

### Наибольший общий делитель. Бинарный алгоритм s

Наибольший общий делитель двух  $n$ -разрядных чисел  $\text{НОД}(a, b)$ , как известно, можно вычислить при помощи алгоритма Евклида за  $O(n)$  итераций вида<sup>3)</sup>  $(a, b) = (b, a \bmod b)$ .

При работе в двоичной арифметике несколько более естественным представляется бинарный алгоритм, предложенный Й. Стайном [313]. Он состоит в циклическом выполнении итераций:

1. Если  $a < b$ , то  $\text{НОД}(a, b) = \text{НОД}(b, a)$ .

2. Если  $b = 0$ , то  $\text{НОД}(a, b) = a$ .

Пусть  $a' = a \bmod 2$ ,  $b' = b \bmod 2$ .

3. Если  $a' = b' = 0$ , то  $\text{НОД}(a, b) = 2 \cdot \text{НОД}(a/2, b/2)$ ;

иначе, если  $a' = 0, b' = 1$ , то  $\text{НОД}(a, b) = \text{НОД}(a/2, b)$ ;

иначе, если  $a' = 1, b' = 0$ , то  $\text{НОД}(a, b) = \text{НОД}(a, b/2)$ ;

иначе,  $\text{НОД}(a, b) = \text{НОД}((a - b)/2, b)$ .

Таким образом, метод просматривает числа (двоичную запись) справа налево и попутно модифицирует их. Довольно очевидно, что на каждой итерации суммарная длина чисел  $a$  и  $b$  уменьшается по меньшей мере на 1, поэтому вычисление НОД  $n$ -разрядных чисел требует не более  $2n - 1$  таких итераций.

<sup>2)</sup>Различие между (1.2) и (1.3) проистекает из двух способов записи функции голосования трех переменных:  $\text{maj}_3(a, b, c) = ab \oplus ac \oplus bc = ab \vee ac \vee bc$ .

<sup>3)</sup>Как только на каком-то шаге получается  $(r, 0)$ , делается заключение  $\text{НОД}(a, b) = r$ .

**Теорема 1.4.**  $C(\text{НОД}_n) \preccurlyeq n^2$ .

► Как обычно, для представления алгоритма НОД в виде схемы требуется немногого дополнительной работы. Соединим последовательно  $2n - 1$  блоков, каждый из которых реализует очередную итерацию алгоритма. Более точно,  $i$ -й блок осуществляет преобразование  $(a_i, b_i, e_i, k_i) \rightarrow (a_{i+1}, b_{i+1}, e_{i+1}, k_{i+1}, r_i)$ , где через  $(a_i, b_i)$  и  $(a_{i+1}, b_{i+1})$  обозначается пара чисел  $(a, b)$  на входе и на выходе при  $a_0 = a$  и  $b_0 = b$ ,  $e_{i+1} = e_i \vee (\min\{a_i, b_i\} = 0)$  — признак выполнения условия шага 2 на какой-то из  $i$  первых итераций (полагаем  $e_0 = 0$ ),  $k_{i+1} = k_i + \overline{a'_i} \cdot \overline{b'_i}$  — накапливающийся при выполнении пункта 1 шага 3 показатель степени двойки в НОД (при  $k_0 = 0$ ),  $r_i = \min\{a_i, b_i\}$  — нечетная составляющая НОД( $a, b$ ) в случае выполнения условия на шаге 2.

Вычисления завершает подсхема выбора правильного значения. Номер итерации, на которой алгоритм завершает работу, определяется условием  $\overline{e_{i-1}} \cdot e_i = 1$ . Тогда  $\text{НОД}(a, b) = 2^{k_i} r_i$ .

Каждый из блоков в цепочке включает подсхемы сравнения, вычитания, сложения и выбора (мультиплексор) и поэтому имеет линейную сложность. Финальная схема включает подсхему выбора сложности  $O(n^2)$  (выбор можно выполнить по формуле<sup>4</sup>)  $[k, r] = \bigvee_i (\overline{e_{i-1}} \cdot e_i)[k_i, r_i]$  и подсхему сдвига, сложность которой оценивается как  $O(n \log n)$  при помощи следующей леммы (см. также в [46]).

Обозначим через  $SH_n(v, x)$  оператор сдвига  $n$ -разрядного числа  $x$  на  $v < n$  позиций влево:  $(v, x) \rightarrow 2^v x$ .

**Лемма 1.1** ([46]).  $C(SH_n) \preccurlyeq n \log n$ .

► Простая схема реализации сдвига строится последовательным методом. Воспользуемся двоичной записью величины сдвига:  $v = [v_k, \dots, v_0]$ . Положим  $x_0 = x$  и далее  $x_i = 2^{2^i v_i} x_{i-1}$  при  $i = 1, \dots, k$ . Тогда  $x_k = 2^v x$ .

Схема сдвига получается последовательным соединением  $k \sim \log_2 n$  подсхем  $(2, 1)$ -мультиплексоров, каждый из которых в зависимости от значения  $v_i$  выбирает либо  $x_{i-1}$ , либо  $2^{2^i} x_{i-1}$ .  $\square$

■

- Имеется не менее десятка вариаций как алгоритма Евклида, так и бинарного алгоритма: в частности, есть правосторонняя версия первого и левосторонняя версия второго. О теоретически более быстрых алгоритмах НОД речь идет в следующей главе, см. на стр. 30.

### Формулы для натуральных чисел $\boxed{s}$

Следующая задача относится к разряду занимательных. Записать натуральное число  $n$  формулой, используя операции сложения, умножения, скобки и возможно меньшее число единиц. Например, минимальная запись для числа 11 имеет

<sup>4</sup>Дизъюнкция выполняется поразрядно.

вид  $(1+1)(1+1+1+1+1)+1$ . Иначе говоря, речь идет об определении величины  $\Phi_{\mathcal{A}_+^N}(n)$ , которую далее будем кратко обозначать<sup>5)</sup>  $\Phi_+(n)$ .

Первое упоминание этой задачи находят в работе К. Малера и Я. Попкена 1953 г. [262]. Несмотря на интерес к проблеме и несколько частных результатов, долгое время были известны только простые общие оценки  $3 \log_3 n \leq \Phi_+(n) < 3 \log_2 n \approx 4.33 \ln n$  (при  $n \geq 2$ ). Верхнюю оценку здесь доставляет схема Горнера. Лишь в 2009 г. Дж. Зелински получил первую нетривиальную верхнюю оценку и позднее улучшил ее до  $\Phi_+(n) < 3.76 \ln n$  [335]. Мы предлагаем к рассмотрению упрощенный вариант его метода.

**Теорема 1.5** ([335]). *При всех  $n \geq 2$  справедливо  $\Phi_+(n) \leq 10 \log_{12} n \approx 4.02 \ln n$ .*

► Обозначим через  $v_p(n)$  число цифр  $p - 1$  на младшем конце  $p$ -ичной записи числа  $n$ . Иначе говоря, это наибольшее  $k$ , при котором  $n \equiv -1 \pmod{p^k}$ .

Также введем преобразование  $\left[ \frac{* - a}{b} \right] : n \rightarrow \frac{n - a}{b}$ . Предполагая  $b \mid (n - a)$ , заметим, что

$$\Phi(n) \leq \Phi\left(\left[\frac{* - a}{b}\right]n\right) + \Phi(a) + \Phi(b). \quad (1.4)$$

Докажем индукцией по  $n$ , что  $\Phi(n) \leq C \ln n$  при подходящей (возможно меньшей) константе  $C$  — ее значение будет установлено в ходе доказательства.

0) При  $2 \leq n \leq 6$  неравенство выполнено с константой  $C = 5/\ln 5 \approx 3.11$ . Теперь, полагая  $n \geq 7$ , докажем индуктивный переход от  $n - 1$  к  $n$ . Далее для краткости положим  $v_p = v_p(n)$ .

1) Пусть  $v_2 \leq 1$ , иначе говоря,  $n \not\equiv 3 \pmod{4}$ . Если младшие двоичные разряды  $n$  равны  $n_1$  и  $n_0$ , то перейдем от  $n$  к  $n' = \left[\frac{* - n_1}{2}\right] \left[\frac{* - n_0}{2}\right] n > 1$  и воспользуемся (1.4). Получаем  $\Phi(n) \leq \Phi(n') + 5$ . Как следствие, индуктивный переход доказывается при любом  $C \geq 5/\ln 4 \approx 3.61$ .

2) Пусть  $v_3 = 0$ . Тогда  $a = n \pmod{3} \leq 1$ . Переходим от  $n$  к  $n' = \left[\frac{* - a}{3}\right] n$ . Согласно (1.4), получаем  $\Phi(n) \leq \Phi(n') + 4$ . В этом случае индуктивный переход устанавливается при  $C \geq 4/\ln 3 \approx 3.64$ .

3) В оставшемся случае  $v_2 \geq 2$  и  $v_3 \geq 1$ . В частности,  $n \geq 11$ . Заметим, что  $n \equiv (2^{v_2} 3^{v_3} - 1) \pmod{2^{v_2+1} 3^{v_3}}$ .

3.1) Рассмотрим  $n' = \left[\frac{* - 1}{2}\right]^{v_2} n$ . Легко проверить, что  $2 \mid n'$  и  $v_3(n') = v_3(n)$ . Следовательно, для  $n'' = n'/2$  выполнено  $n'' \equiv \frac{3^{v_3}-1}{2} \pmod{3^{v_3}}$ . Таким образом, троичная запись числа  $n''$  оканчивается на  $v_3$  единиц. Поэтому при  $n'' \neq \frac{3^{v_3}-1}{2}$  возможен переход к  $n''' = \left[\frac{* - 1}{3}\right]^{v_3} n''$ , в противном случае — к  $1 = \left[\frac{* - 1}{3}\right]^{v_3-1} n''$ . В общем случае получаем

$$\Phi(n) \leq \Phi(n''') + 3v_2 + 2 + 4v_3 \leq C \ln(n/(2^{v_2+1} 3^{v_3})) + 3v_2 + 4v_3 + 2.$$

Как следствие, для обоснования индуктивного перехода требуется

$$3v_2 + 4v_3 + 2 \leq C((v_2 + 1) \ln 2 + v_3 \ln 3).$$

---

<sup>5)</sup>Обычно эту величину обозначают  $\|n\|$ .

При  $C \geq 2/\ln 2 \approx 2.88$  это неравенство следует из более простого:

$$3v_2 + 4v_3 \leq C(v_2 \ln 2 + v_3 \ln 3). \quad (1.5)$$

В частном случае  $n'' = \frac{3^{v_3}-1}{2}$  получаем вычисление  $n = 2^{v_2}3^{v_3} - 1$  со сложностью  $3v_2 + 4v_3 - 2$ . Условие (1.5) является достаточным для обоснования перехода в силу  $\ln(n+1) \leq \ln n + \frac{1}{n}$  при  $n \geq 11$ .

3.2) Пусть  $v_3 < v_2/2$ . Положим  $n' = [\frac{*-2}{3}]^{v_3} n$ . Несложно проверить, что  $a = (n' \bmod 3) \neq 2$  и  $v_2(n') = v_2(n)$ . Тогда  $n'$  имеет вид

$$n' = 2^{v_2+1}(3m - \delta) + 2^{v_2} - 1, \quad \delta = \begin{cases} 0, & a = (v_2 \bmod 2) \\ 1, & a \neq (v_2 \bmod 2) \end{cases}.$$

Рассмотрим  $n'' = [\frac{*-a}{3}]n'$ . Тогда

$$(n'' \bmod 2^{v_2+1}) \in \left\{ \frac{2^{v_2}-1}{3}, \frac{2^{v_2}-2}{3}, \frac{5 \cdot 2^{v_2}-2}{3}, \frac{5 \cdot 2^{v_2}-1}{3} \right\}.$$

В системе с основанием 4 эти остатки имеют вид соответственно

$$\begin{aligned} 4^{\frac{v_2}{2}-1} + \dots + 4 + 1, & \quad 2(4^{\frac{v_2-1}{2}-1} + \dots + 4 + 1), \\ 4^{\frac{v_2}{2}} + 2(4^{\frac{v_2}{2}-1} + \dots + 4 + 1), & \quad 3 \cdot 4^{\frac{v_2-1}{2}} + (4^{\frac{v_2-1}{2}-1} + \dots + 4 + 1). \end{aligned}$$

Поэтому при  $n'' \neq \frac{2^{v_2}-1}{3}$  возможен переход к  $n''' = [\frac{*-d}{4}]^{\lfloor v_2/2 \rfloor} n''$ , где  $d \in \{1, 2\}$ . Раскладывая, как в п. 1, преобразование  $[\frac{*-d}{4}]$  в композицию  $[\frac{*-n_1}{2}] [\frac{*-n_0}{2}]$ , получаем

$$\Phi(n) \leq \Phi(n''') + 5v_3 + 4 + 5\lfloor v_2/2 \rfloor \leq C \ln(n/(3^{v_3+1}4^{\lfloor v_2/2 \rfloor})) + 5\lfloor v_2/2 \rfloor + 5v_3 + 4.$$

В этом случае для обоснования индуктивного перехода необходимо

$$5\lfloor v_2/2 \rfloor + 5v_3 + 4 \leq C(2\lfloor v_2/2 \rfloor \ln 2 + (v_3 + 1) \ln 3).$$

При  $C \geq 4/\ln 3$  это неравенство вытекает из

$$5\lfloor v_2/2 \rfloor + 5v_3 \leq C(2\lfloor v_2/2 \rfloor \ln 2 + v_3 \ln 3). \quad (1.6)$$

В оставшемся случае  $n'' = \frac{2^{v_2}-1}{3}$  (при этом  $2 \mid v_2$  и  $a = 0$ ) выполняется переход к  $1 = [\frac{*-1}{4}]^{(v_2/2)-1} n''$ . Так получаем вычисление  $n = 2^{v_2}3^{v_3} - 1$  со сложностью  $5v_3 + 5\lfloor v_2/2 \rfloor - 2$ . Как и в п. 3.1 выше, условие (1.6) является достаточным для обоснования индуктивного перехода.

3.3) Осталось определить минимальную константу  $C$ , с которой проходит доказательство. Ее величина определяется соотношениями (1.5) и (1.6). Для любых  $v_2, v_3$  должно быть выполнено

$$C \geq \min \left\{ \frac{3v_2 + 4v_3}{v_2 \ln 2 + v_3 \ln 3}, \frac{5\lfloor v_2/2 \rfloor + 5v_3}{2\lfloor v_2/2 \rfloor \ln 2 + v_3 \ln 3} \right\}.$$

При  $v_3 \geq v_2/2$  используем первую оценку, ее максимум достигается при  $v_3 = v_2/2$ , а при  $v_3 \leq \lfloor v_2/2 \rfloor$  — вторую, максимум которой достигается при  $v_3 = \lfloor v_2/2 \rfloor$ . В обоих случаях максимальные значения равны  $10/\ln 12$ . ■

- Более сильную оценку  $\Phi_+(n) < 3.76 \ln n$  Зелински получил, анализируя разложения по большому числу простых оснований. Предполагается, что максимум отношения  $\Phi_+(n)/\ln n$  достигается на числе 1439 и равен  $26/\ln 1439 \approx 3.58$ . Разумеется, это не исключает возможности получения оценок вида  $(C + o(1)) \ln n$  с гораздо меньшими константами  $C$ . К. Амано [126] с опорой на компьютерные расчеты доказал, что для почти всех  $n$  справедливо  $\Phi_+(n) < 3.24 \ln n$ .

### Алгоритмы динамического программирования. Вычисление функции проводимости и гамильтониана s

Сопоставим ребрам полного ориентированного графа  $K_n$  на  $n$  вершинах булевые переменные  $X = \{x_e\}$ . Значения переменных определяют произвольный граф  $G \subset K_n$ :  $x_e = 1$  означает, что  $e \in G$ ;  $x_e = 0$  — что  $e \notin G$ . Функция  $(s, t)$ -проводимости графа определяет наличие ориентированного пути, соединяющего вершины  $s$  и  $t$ . Ее можно задать формулой

$$CONN_n(X) = \bigvee_{P \text{ — } (s, t)\text{-путь в } K_n} \bigwedge_{e \in P} x_e.$$



Ричард Эрнест  
Беллман

Университет Южной  
Каролины, с 1965 по 1984

Несмотря на то, что число путей, соединяющих две вершины в графе, может быть экспоненциально велико, все их можно «перебрать» простым алгоритмом полиномиальной сложности. Этот алгоритм был независимо обнаружен Л. Фордом [193], Э. Муром [269] и Р. Беллманом [133] в 1950-х гг.

**Теорема 1.6** ([133, 193, 269]).  $C_{B_M}(CONN_n) \preccurlyeq n^3$ .

► Пусть вершины графа нумеруются натуральными числами из от 1 до  $n$ . Для удобства положим  $s = 1$  и  $t = n$ .

Обозначим через  $y_j^{(l)}$  функцию, характеризующую наличие пути длины  $l$  из вершины 1 в вершину  $j$ . Все  $y_j^{(l)}$  вычисляются по рекуррентным формулам

$$y_j^{(1)} = x_{1,j}, \quad y_j^{(l+1)} = \bigvee_{i=2}^{n-1} y_i^{(l)} \cdot x_{i,j} \quad (1.7)$$

со сложностью  $\preccurlyeq n^3$ . Окончательно,  $CONN_n = y_n^{(1)} \vee y_n^{(2)} \vee \dots \vee y_n^{(n-1)}$ . ■

Принцип поступательного движения к решению задачи через решение подобных подзадач известен под названием *динамическое программирование* (термин

предложен Беллманом). На его базе строятся эффективные алгоритмы решения многих оптимизационных проблем (построение дерева Штейнера, поиск максимального независимого множества в графе и т.п.).

- Алгоритм Беллмана—Форда—Мура работает не только в булевом, но и во многих других арифметических полукольцах. Наибольшее прикладное значение он имеет для тропических полукольц (  $\mathbb{R}, \min, +$  ). Булевой задаче проводимости соответствует задача о поиске кратчайшего пути между двумя вершинами в графе: переменные  $x_e$  принимают вещественные значения и интерпретируются как веса, приписанные ребрам, вычисляемая функция имеет вид  $\min_P \sum_{e \in P} x_e$ , где  $P$  пробегает множество путей. Алгоритм существенно опирается на идемпотентность аддитивной операции полукольца (формулы (1.7) включают пути с циклами). Обычный вещественный многочлен — аналог функции  $CONN_n$  — имеет уже сверхполиномиальную монотонную арифметическую сложность  $\asymp n^2 2^n$ , как показали М. Джеррум и М. Шнир [227].

Пусть  $\Pi(j_1, \dots, j_k)$  обозначает множество циклических перестановок чисел  $j_1, \dots, j_k$ . Гамильтониан порядка  $n$  — это многочлен

$$HAM_n(X) = \sum_{(i_1, \dots, i_{n-1}) \in \Pi(1, 2, \dots, n-1)} x_{0,i_1} x_{i_1, i_2} \cdots x_{i_{n-2}, i_{n-1}} x_{i_{n-1}, 0},$$

в котором каждый моном соответствует гамильтонову циклу в полном ориентированном графе на  $n$  вершинах  $0, 1, \dots, n-1$ . При подстановке вместо переменных признаков наличия ребер в произвольном графе  $G$  (как выше) многочлен принимает значение числа гамильтоновых циклов в  $G$ .

Алгоритм динамического программирования Р. Беллмана [134], М. Хелда и Р. Карпа [218] позволяет построить оптимальную монотонную арифметическую схему для гамильтониана.

**Теорема 1.7** ([134, 218]).  $C_{\mathcal{A}_+}(HAM_n) \asymp n^2 2^n$ .

- Обозначим через  $H_{s,J}$ , где  $J \subset \llbracket n \rrbracket \setminus \{0, s\}$  и  $|J| = k$ , многочлен

$$H_{s,J} = \sum_{(i_1, \dots, i_k) \in \Pi(J)} x_{s,i_1} x_{i_1, i_2} \cdots x_{i_{k-1}, i_k} x_{i_k, 0},$$

перечисляющий гамильтоновы пути из вершины  $s$  в вершину  $0$  в полном графе на множестве вершин  $J \cup \{0, s\}$ .

Положим  $H_{s,\emptyset} = x_{s,0}$ . В общем случае имеем

$$H_{s,J} = \sum_{j \in J} x_{s,j} \cdot H_{j, J \setminus \{j\}} \quad \text{и} \quad HAM_n = \sum_{j=1}^{n-1} x_{0,j} \cdot H_{j, \llbracket n \rrbracket \setminus \{0,j\}}. \quad (1.8)$$

Если все многочлены  $H_{s,J}$  при  $1 \leq s \leq n-1$  и  $|J| = k-1$  вычислены, то вычисление многочленов  $H_{s,J}$ ,  $|J| = k$ , по формуле (1.8) требует не более  $2knC_{n-2}^k$  операций. Следовательно,

$$C(HAM_n) \leq 2n + \sum_{k=1}^{n-2} 2knC_{n-2}^k \asymp n^2 2^n.$$

■

- Джеррум и Шнир [227] показали, что оценка теоремы 1.7 по порядку точна не только для вещественного арифметического, но и для тропического полукольца  $(\mathbb{R}, \min, +)$ , в котором гамильтониан соответствует оптимизационной задаче коммивояжера — поиску кратчайшего обхода всех вершин графа. При этом мультипликативная сложность алгоритма теоремы 1.7, которая, если считать аккуратно, составляет  $(n-1)((n-2)2^{n-3}+1)$ , — минимально возможная.

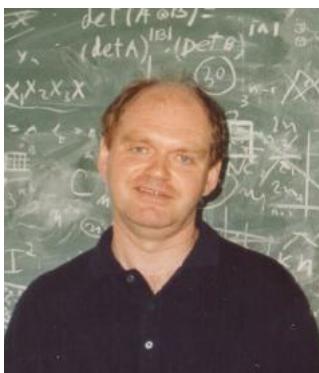
## Монотонные схемы для многочлена Кирхгофа s

Идея последовательного вычисления позволяет получать в том числе и весьма нетривиальные результаты, в чем мы убедимся на примере вычисления многочленов Кирхгофа.

Рассмотрим связный неориентированный граф  $G$ , ребрам которого приписаны символы вещественных переменных  $x_e$  (веса). Напомним, что *остовным деревом* графа называется подграф, который является деревом, связывающим все вершины графа. *Многочлен Кирхгофа* (многочлен оставных деревьев) графа  $G$  определяется как

$$ST_G(X) = \sum_{T \text{ — ост. дер. в } G} \prod_{e \in T} x_e.$$

Многочлен Кирхгофа несвязного графа удобно определить как произведение многочленов Кирхгофа связных компонент. Допускается наличие кратных ребер.



Дмитрий Юрьевич

Григорьев

Национальный центр научных  
исследований Франции,  
Лилль, с 1998

Монотонная арифметическая сложность многочлена Кирхгофа велика: С. Юкна и Х. Сайверт [232] доказали<sup>6</sup>, что  $C_{\mathcal{A}_{+}^{\mathbb{R}}}(ST_{K_n}) = 2^{\Omega(\sqrt{n})}$ . Ситуация меняется при расширении базиса. Так, недавно С. Фомин, Д. Григорьев и Г. Кошевой [192] обнаружили существование простых монотонных схем с делением для этой задачи<sup>7</sup>). Метод основан на последовательном добавлении вершин в граф.

**Теорема 1.8** ([192]). Для любого графа  $G$  на  $n$  вершинах  $C_{\mathcal{A}_{D+}^{\mathbb{R}}}(ST_G) \leq n^3$ , где  $\mathcal{A}_{D+} = \{+, *, /\}$ .

► Достаточно построить схему для полного графа  $K_n$ . Схема для произвольного графа получается подстановкой нулей вместо некоторых переменных.

Пусть вершины графа нумеруются числами от 1 до  $n$ , ребра — парами чисел.

<sup>6</sup>Результат справедлив не только в арифметическом, но и в тропическом полукольце  $(\mathbb{R}, \min, +)$ .

<sup>7</sup>В определенном смысле, схемы с делением не совсем монотонны, т.к. они позволяют вычислять некоторые многочлены с отрицательными коэффициентами, например,  $x^2 - xy + y^2$  как  $(x^3 + y^3)/(x + y)$ .

Схема строится индуктивно. Обозначим  $w_n = x_{1,n} + x_{2,n} + \dots + x_{n-1,n}$ . Удалим из графа вершину  $n$  и добавим новые ребра  $e_{i,j}$ , соединяющие всевозможные пары  $i, j$  оставшихся вершин<sup>8)</sup>. Ребру  $e_{i,j}$  припишем вес  $x'_{i,j} = x_{i,n} \cdot x_{j,n} / w_n$ . Новый граф обозначим через  $K'_{n-1}$ .

Наша цель — доказать соотношение

$$ST_{K_n}(X) = w_n \cdot ST_{K'_{n-1}}(X, X'). \quad (1.9)$$

Если выделить в произвольном оствовном дереве графа  $K_n$  ребра, инцидентные вершине  $n$ , получим следующий способ перечисления всех оствовных деревьев. Пусть  $J = \{J_1, \dots, J_s\}$  — некоторое разбиение множества вершин  $1, \dots, n-1$  на подмножества. В каждом полном графе на вершинах  $J_i$  выберем оствовное дерево и соединим его ребром с вершиной  $n$ , см. рис. 1.3а.

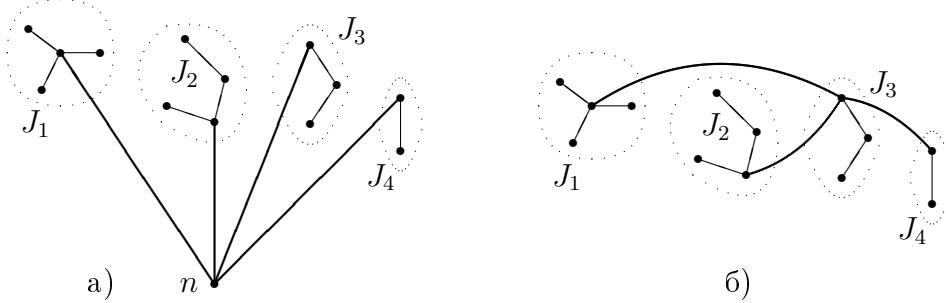


Рис. 1.3: Структура оствовных деревьев графов  $K_n$  (а) и  $K'_{n-1}$  (б)

Пусть  $K_A$  — полный подграф графа  $K_n$  на множестве вершин  $A$ . Обозначим  $w_A = \sum_{i \in A} x_{i,n}$ . Тогда

$$ST_{K_n} = \sum_J \prod_{A \in J} w_A \cdot ST_{K_A}. \quad (1.10)$$

Аналогично, выделяя из оствовного дерева графа  $K'_{n-1}$  множество дополнительных ребер  $e_{i,j}$ , получим способ перечисления всех оствовных деревьев в  $K'_{n-1}$ . Снова пусть  $J = \{J_1, \dots, J_s\}$  — разбиение множества вершин  $1, \dots, n-1$ . В каждом графе  $K_{J_i}$  выберем оствовное дерево. Эти деревья соединим ребрами  $e_{i,j}$ , руководствуясь (внешним) оствовным деревом полного графа, построенного на множествах  $J_i$  как на вершинах, см. рис. 1.3б. При любых  $1 \leq k, l \leq s$  положим

$$X'_{k,l} = \sum_{i \in J_k, j \in J_l} x'_{i,j} = w_{J_k} \cdot w_{J_l} / w_n. \quad (1.11)$$

Обозначим через  $K_J$  полный граф на вершинах  $J_i$  с весами ребер  $X'_{k,l}$ . Получаем

$$ST_{K'_{n-1}} = \sum_J ST_{K_J} \prod_{A \in J} ST_{K_A}. \quad (1.12)$$

<sup>8)</sup>По ходу доказательства мы допускаем наличие в графе нескольких ребер, соединяющих одну и ту же пару вершин.

Теперь (1.9) следует из (1.10) и (1.12), если выполнено  $ST_{K_J} = (\prod_{A \in J} w_A) / w_n$ . Проверим это.

Рассмотрим вспомогательную задачу. Пусть в полном графе  $K_s$  на  $s$  вершинах вес ребра  $(i, j)$  равен  $x_i x_j$  (фактически теперь мы приписываем веса вершинам). Следующая лемма является вариантом теоремы А. Кэли [162].

**Лемма 1.2** ([162]).  $ST_{K_s} = x_1 x_2 \cdot \dots \cdot x_s (x_1 + x_2 + \dots + x_s)^{s-2}$ .

▷ Доказательство основано на известном способе подсчета оствовых деревьев.

По условию, вклад каждого оствового дерева в многочлен  $ST_{K_s}$  имеет вид  $\prod_{i=1}^s x_i^{d_i}$ , где  $d_i$  — степень вершины  $i$  в дереве.

Оствовое дерево однозначно определяется кодом  $[a_1, a_2, \dots, a_{s-2}]$ ,  $1 \leq a_i \leq s$ , так:  $a_1$  — номер вершины, с которой соединена висячая вершина с минимальным номером; удалим эту висячую вершину вместе с инцидентным ей ребром, после чего  $a_2$  определяется аналогично, и так до тех пор, пока в дереве не останется только одна вершина<sup>9)</sup>.

Осталось заметить, что дерево с кодом  $[a_1, a_2, \dots, a_{s-2}]$  входит в многочлен  $ST_{K_s}$  в виде монома  $x_1 x_2 \cdot \dots \cdot x_s \prod_{i=1}^{s-2} x_{a_i}$ .  $\square$

Согласно (1.11), в лемме 1.2 следует положить  $x_i = w_{J_i} / \sqrt{w_n}$ , чтобы с учетом  $w_{J_1} + \dots + w_{J_s} = w_n$  получить  $ST_{K_J} = (\prod_{A \in J} w_A) / w_n$ , что доказывает (1.9).

Наконец, несложно видеть, что любой набор кратных ребер (в графе  $K'_{n-1}$ ) можно заменить одним, вес которого равен сумме весов этих ребер — многочлен Кирхгофа при этом не изменится. Поэтому формула (1.9) влечет соотношение

$$\mathsf{C}(ST_{K_n}) \leq \mathsf{C}(ST_{K_{n-1}}) + O(n^2),$$

откуда следует оценка теоремы.

В заключение заметим, что при переходе от графа  $K_n$  к произвольному связному графу  $G$  в схеме не возникает делений на ноль, поскольку  $w_n \neq 0$  во всех расчетных формулах. В случае несвязного графа достаточно вычислить многочлены Кирхгофа всех связных компонент.  $\blacksquare$

- Булева версия многочлена Кирхгофа  $\bigvee_T \bigwedge_{e \in T} x_e$  (сумма берется по оствовым деревьям  $T \subset G$ ) определяет связность графа  $G$ . Она тоже имеет монотонную булеву сложность  $\preccurlyeq n^3$ . Это можно проверить, используя алгоритм динамического программирования Б. Роя [295], Р. Флойда [191] и С. Уоршелла [328] определения наличия путей между всеми парами вершин в графике. Тропический вариант задачи имеет сверхполиномиальную сложность  $2^{\Omega(\sqrt{n})}$  [232].

---

<sup>9)</sup>Такой способ перечисления оствовых деревьев предложен Х. Прюфером [289].

## Глава 2

### Деление пополам

/2

Деление задачи на подобные части, две или более, — один из самых продуктивных приемов теории быстрых вычислений, позволяющий строить эффективные рекурсивные алгоритмы в разнообразных ситуациях.

#### Сложность формул для линейной функции /2

Короткие формулы для линейной булевой функции  $\Lambda_n(X)$  в базисе  $\mathcal{B}_0$  строятся, следуя обобщающему (1.1) правилу

$$\Lambda_n(X) = \Lambda_{n_1}(X^1) \cdot \overline{\Lambda_{n_2}(X^2)} \vee \overline{\Lambda_{n_1}(X^1)} \cdot \Lambda_{n_2}(X^2), \quad (2.1)$$

где  $X = (X^1, X^2)$ ,  $|X^i| = n_i$  и  $n = n_1 + n_2$ . Следующий результат получен С. В. Яблонским [114] еще в 1950-х гг.

**Теорема 2.1** ([114]). *При любом  $n$*

$$\Phi_{\mathcal{B}_0}(\Lambda_n) \leq n^2 + (n - 2^{\lfloor \log_2 n \rfloor}) (2^{\lceil \log_2 n \rceil} - n) < (9/8)n^2. \quad (2.2)$$

*В частности, при  $n = 2^m$  справедливо  $\Phi_{\mathcal{B}_0}(\Lambda_n) \leq n^2$ .*

► Требуемая оценка доказывается по индукции при помощи (2.1) с разбиением множества переменных на равные части:  $n_1 = \lfloor n/2 \rfloor$  и  $n_2 = \lceil n/2 \rceil$ . ■

Теорема 2.1 дает правильный порядок сложности линейной функции в силу известной нижней оценки В. М. Храпченко [104]  $\Phi_{\mathcal{B}_0}(\Lambda_n) \geq n^2$ , а в случае  $n = 2^m$  построенные формулы просто минимальны.

● Вопрос о точности оценки (2.2) при любом  $n$  (проблема С. В. Яблонского [114]) остается открытым. Усилиями К. Л. Рычкова [73, 74] и Д. Ю. Черухина [113] точность теоремы 2.1 установлена при всех  $n \leq 7$ .

Отметим, что нижняя оценка сложности формул доказывается с помощью приема, в определенном смысле двойственного к доказательству верхней оценки. А именно, на множестве функций вводится подходящий функционал  $\mu(f)$ ,

который при движении от простых функций к сложным растет примерно также, как функционал сложности, но при этом проще вычисляется для конкретных функций.

- Такой функционал называется *формальной мерой сложности*. Если он удовлетворяет условиям

$$\mu(0) = \mu(1) = 0, \quad \mu(x), \mu(\bar{x}) \leq 1, \quad \mu(f \vee g) \leq \mu(f) + \mu(g), \quad \mu(f \cdot g) \leq \mu(f) + \mu(g),$$

то автоматически  $\Phi_{\mathcal{B}_0}(f) \geq \mu(f)$ . Методу Храпченко [104] соответствует мера

$$\mu(f) = \max_{N \subset f^{-1}(0), P \subset f^{-1}(1)} \frac{|R(N, P)|^2}{|N| \cdot |P|},$$

где  $R(N, P)$  — множество пар соседних наборов из  $N$  и  $P$ , т. е. отличающихся в одной координате. Подробнее см., например, в [330, 231].

### Умножение чисел. Метод Карацубы /2



Анатолий Алексеевич  
Карацуба

Московский университет,  
с 1959 по 2008

Применим тот же прием в задаче умножения чисел. Разобьем  $2n$ -разрядные числа  $X$  и  $Y$  на блоки по  $n$  разрядов:  $X = X_1 2^n + X_0$ ,  $Y = Y_1 2^n + Y_0$ . Теперь умножение исходных чисел может быть выполнено при помощи четырех умножений «половинок» и нескольких сложений по формуле

$$XY = X_1 Y_1 2^{2n} + (X_0 Y_1 + X_1 Y_0) 2^n + X_0 Y_0.$$

Этот способ приводит к квадратичной оценке сложности  $C(M_n) \preccurlyeq n^2$ , как и прямой метод умножения столбиком, где  $M_n$  обозначает булев  $(2n, 2n)$ -оператор умножения  $n$ -разрядных чисел.

Неожиданно<sup>1)</sup> в 1960 г. А. А. Карацуба [20] обнаружил более экономную формулу, использующую всего три умножения половинного размера:

$$XY = X_1 Y_1 2^{2n} + ((X_0 + X_1)(Y_0 + Y_1) - X_1 Y_1 - X_0 Y_0) 2^n + X_0 Y_0. \quad (2.3)$$

**Теорема 2.2** ([20]).  $C(M_n) \preccurlyeq n^{\log_2 3}$ .

- Заметим, что  $C(M_{n+1}) \leq C(M_n) + O(n)$ . Тогда формула (2.3) ведет к рекуррентному соотношению

$$C(M_{2n}) \leq C(M_{n+1}) + 2C(M_n) + O(n) = 3C(M_n) + O(n),$$

которое разрешается как  $C(M_n) \preccurlyeq n^{\log_2 3} \prec n^{1.59}$ . ■

---

<sup>1</sup>В то время как многие считали, что следует сосредоточить усилия на доказательстве оценки  $C(M_n) \asymp n^2$ .

Несмотря на кажущуюся простоту, в начале 1960-х метод Карацубы произвел переворот в теории быстрых вычислений и открыл путь к построению быстрых алгоритмов для множества других задач.

- Метод Карацубы широко применяется на практике. Аккуратная оценка сложности метода при  $n = 2^k$  имеет вид  $C_{B_2}(M_n) < 25 \frac{83}{405} \cdot 3^k$  [87]. Еще успешнее метод применяется к умножению многочленов. Например, в наиболее интересном случае двоичных многочленов сложность умножения оптимизированным методом Карацубы оценивается как  $C_{A^F_2}(M_n^{\mathbb{F}_2}) < 5 \frac{13}{18} \cdot 3^k$  [140].

А. Л. Тоом [98] обобщил метод Карацубы: умножение длинных чисел, если разбить их на  $k$  блоков, сводится к  $2k - 1$  умножениям коротких чисел (их длина соответствует размеру блока). При подходящем выборе параметра  $k$  получается оценка  $C(M_n) \leq 2^{O(\sqrt{\log n})} n$ .

### Умножение матриц. Метод Штрассена /2

Перейдем к задаче вычисления произведения квадратных матриц  $Z = XY$  над некоторым полукольцом  $R$  арифметическими схемами. Умножение  $2n \times 2n$  матриц сводится к умножению матриц размера  $n \times n$ :

$$\begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \cdot \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} X_{11}Y_{11} + X_{12}Y_{21} & X_{11}Y_{12} + X_{12}Y_{22} \\ X_{21}Y_{11} + X_{22}Y_{21} & X_{21}Y_{12} + X_{22}Y_{22} \end{bmatrix}. \quad (2.4)$$

Прямое вычисление по формулам (2.4) приводит к алгоритму кубической сложности,  $C_{A^R}(MM_n) \leq n^3$ , где через  $MM_n^R$  обозначается оператор умножения  $n \times n$  матриц над  $R$ .

Ф. Штрассен [316] заметил, что одно умножение подматриц можно сэкономить, если выполнять вычисления по формулам

$$\begin{aligned} Z_{11} &= U_1 + U_2 - U_3 + (X_{12} - X_{22})(Y_{21} + Y_{22}), & Z_{12} &= U_3 + U_5, \\ Z_{21} &= U_2 + U_4, & Z_{22} &= U_1 - U_4 + U_5 + (X_{21} - X_{11})(Y_{11} + Y_{12}), \\ U_1 &= (X_{11} + X_{22})(Y_{11} + Y_{22}), & U_2 &= X_{22}(Y_{21} - Y_{11}), \\ U_3 &= (X_{11} + X_{12})Y_{22}, & U_4 &= (X_{21} + X_{22})Y_{11}, & U_5 &= X_{11}(Y_{12} - Y_{22}). \end{aligned} \quad (2.5)$$

Использование вычитаний подразумевает, что  $R$  является кольцом.

**Теорема 2.3** ([316]). *Если  $R$  — кольцо, то  $C_{A^R}(MM_n) \leq n^{\log_2 7}$ .*

- Формулы (2.5) приводят к соотношению

$$C_{A^R}(MM_{2n}) \leq 7C_{A^R}(MM_n) + O(n^2),$$

которое разрешается как  $C_{A^R}(MM_n) \leq n^{\log_2 7} \prec n^{2.81}$ . ■

- Если вычисления выполняются в монотонном базисе<sup>2)</sup>, то кубическая оценка не может быть улучшена,  $C_{A^R_+}(MM_n) \geq n^3$  [239]. Уже для булева полукольца  $(\mathbb{B}, \vee, \wedge)$  выполнено  $C_{B_M}(MM_n) = 2n^3 - n^2$  [276].

Формулы (2.5), помимо 7 умножений, используют 18 аддитивных операций с подматрицами. Ш. Виноград сократил число сложений/вычитаний до 15 (см., например, [24]), а в работе [331] он показал, что мультиплекативная сложность умножения  $2 \times 2$  матриц действительно равна 7. Позже было доказано, что если  $R$  — поле, то 15 аддитивных операций необходимы в любой схеме умножения  $2 \times 2$  матриц с мультиплекативной сложностью 7 [288, 161].

<sup>2)</sup>Так будет, например, в полукольце  $R$  с необратимой операцией сложения.

## Быстрое преобразование Фурье $\boxed{/_2}$

Пусть  $\zeta$  — примитивный корень степени  $N$  в коммутативном ассоциативном кольце  $R$  с единицей. *Дискретным преобразованием Фурье* ( $\mathcal{D}\Phi$ ) порядка  $N$  называется линейный  $(N, N)$ -оператор над  $R$

$$\mathcal{D}\Phi_{N, \zeta}[R](x_0, \dots, x_{N-1}) \rightarrow (x_0^*, \dots, x_{N-1}^*), \quad x_j^* = \sum_{i=0}^{N-1} \zeta^{ij} x_i. \quad (2.6)$$

- *Основные свойства  $\mathcal{D}\Phi$ .* Обратное к  $\mathcal{D}\Phi$  преобразование совпадает с исходным с точностью до замены примитивного корня и нормировки:

$$\mathcal{D}\Phi_{N, \zeta}^{-1} = N^{-1} \cdot \mathcal{D}\Phi_{N, \zeta^{-1}}.$$

В полиномиальной интерпретации  $\mathcal{D}\Phi$  вычисляется значения многочлена в точках  $\zeta^i$ ,  $i = 0, \dots, N-1$ . Пусть  $\Gamma(t) = x_0 + x_1 t + \dots + x_{N-1} t^{N-1}$ . Тогда

$$\mathcal{D}\Phi_{N, \zeta}(x_0, \dots, x_{N-1}) = (\Gamma(\zeta^0), \dots, \Gamma(\zeta^{N-1})).$$

Матрица  $\mathcal{D}\Phi$  — это матрица Вандермонда, построенная на степенях примитивного корня  $\zeta^0, \zeta^1, \dots, \zeta^{N-1}$ .

В центре теории быстрого преобразования Фурье лежит прием декомпозиции  $\mathcal{D}\Phi$  составного порядка из работы Дж. Кули и Дж. Тьюки [173].

**Лемма 2.1** ([173]). *Пусть  $\zeta$  — примитивный корень степени  $P \cdot Q$ . Тогда*

$$\mathsf{C}_{\mathcal{A}_L}(\mathcal{D}\Phi_{PQ, \zeta}) \leq P \cdot \mathsf{C}_{\mathcal{A}_L}(\mathcal{D}\Phi_{Q, \zeta^P}) + Q \cdot \mathsf{C}_{\mathcal{A}_L}(\mathcal{D}\Phi_{P, \zeta^Q}) + (P-1)(Q-1).$$

▷ При любых  $p = 0, \dots, P-1$  и  $q = 0, \dots, Q-1$  запишем

$$\begin{aligned} x_{pQ+q}^* &= \sum_{I=0}^{PQ-1} \zeta^{I(pQ+q)} x_I = \sum_{i=0}^{Q-1} \sum_{j=0}^{P-1} \zeta^{(iP+j)(pQ+q)} x_{iP+j} = \\ &= \sum_{i=0}^{Q-1} \sum_{j=0}^{P-1} \zeta^{iqP+jpQ+jq} x_{iP+j} = \sum_{j=0}^{P-1} (\zeta^Q)^{jp} \cdot \zeta^{jq} \cdot \sum_{i=0}^{Q-1} (\zeta^P)^{iq} x_{iP+j}. \end{aligned} \quad (2.7)$$

Внутренние суммы вычисляются при помощи  $\mathcal{D}\Phi$  порядка  $Q$ . Результаты умножаются на степень примитивного корня  $\zeta$  (среди  $PQ$  умножений  $P+Q-1$  умножений выполняется на единицу, при  $j=0$  или  $q=0$ ). Наконец, внешние суммы находятся при помощи  $\mathcal{D}\Phi$  порядка  $P$ .  $\square$

- Если  $\text{НОД}(P, Q) = 1$ , то возможен более экономный способ вычисления, не требующий дополнительных умножений на степени  $\zeta$ . Он был найден И. Гудом [204] в конце 1950-х гг.

Для  $I = 0, \dots, PQ-1$  обозначим  $x_I = x_{i,j}$ , где  $i = I \bmod Q$ , а  $j = I \bmod P$ . Пусть  $a, b$  — коэффициенты Безу из равенства  $aP + bQ = 1$ . Заметим, что  $I = (iaP + jbQ) \bmod PQ$ .

Для произвольного  $K = 0, \dots, PQ-1$  положим  $s = bK \bmod P$  и  $t = aK \bmod Q$ . Легко видеть, что  $K = (sQ + tP) \bmod PQ$ .

Наконец, заметим, что если  $\zeta$  — примитивный корень порядка  $PQ$ , то  $\zeta^{bQ^2} = \zeta^{Q(1-aP)} = \zeta^Q$  и аналогично  $\zeta^{aP^2} = \zeta^P$ . Теперь легко проверяется тождество

$$\begin{aligned} x_K^* &= \sum_{I=0}^{PQ-1} \zeta^{IK} x_I = \sum_{j=0}^{P-1} \sum_{i=0}^{Q-1} \zeta^{(iaP+jbQ)(sQ+tP)} x_{i,j} = \\ &= \sum_{j=0}^{P-1} \sum_{i=0}^{Q-1} \zeta^{(iatP^2+jbsQ^2)} x_{i,j} = \sum_{j=0}^{P-1} \sum_{i=0}^{Q-1} (\zeta^P)^{it} (\zeta^Q)^{js} x_{i,j} = \sum_{j=0}^{P-1} (\zeta^Q)^{js} \sum_{i=0}^{Q-1} (\zeta^P)^{it} x_{i,j}. \end{aligned}$$

Вычисления по этим формулам требуют только  $Q$  ДПФ порядка  $P$  и  $P$  ДПФ порядка  $Q$ . Однако область применения метода Гуда уже, чем для метода Кули—Тьюки.

Рекурсивным применением леммы 2.1 с учетом  $C_{\mathcal{A}_L}(\text{ДПФ}_2) = 2$  доказывается<sup>3)</sup>

**Теорема 2.4** ([173]).  $C_{\mathcal{A}_L}(\text{ДПФ}_{2^k}) \leq 3k2^{k-1} - 2^k + 1$ .

- Схема теоремы 2.4 состоит из  $k2^k$  сложений/вычитаний и  $(k-2)2^{k-1} + 1$  скалярных умножений на степени примивного корня, отличные от  $\pm 1$ . Первая оценка пока не улучшена, а скалярная мультипликативная сложность ДПФ порядка  $n$  равна  $O(n)$  [332, 217], но такая оценка достигается ценой существенного увеличения аддитивной сложности.

## Параллельные префиксные схемы $\boxed{/2}$

Рассмотрим систему префиксных сумм

$$x_1 \circ x_2 \circ \dots \circ x_i, \quad i = 1, \dots, n, \quad (2.8)$$

над некоторой полугруппой  $(G, \circ)$  с ассоциативной, но не обязательно коммутативной операцией сложения. Схемы над базисом из единственной операции  $\{\circ\}$ , реализующие эту систему, обычно называют *префиксными схемами*.

Сложность системы (2.8) очевидно равна  $n-1$ , однако глубина минимальной схемы тоже равна  $n-1$ . Еще в 1960-е годы в связи с потребностями ряда приложений возникла необходимость строить параллельные префиксные схемы, т.е. схемы глубины  $O(\log n)$ . Вопрос: какова может быть сложность таких схем. Более конкретный вопрос: можно ли построить префиксную схему минимально возможной глубины  $\lceil \log_2 n \rceil$  и сложности  $O(n)$ . Увердительный ответ дан в работе Р. Ладнера и М. Фишера [251].

В [251] строится семейство префиксных схем  $\Pi_k(n)$ . Схема  $\Pi_k(n)$  удовлетворяет двум условиям:

- (\*) реализует систему (2.8) с глубиной  $\lceil \log_2 n \rceil + k$ ;
- (\*\*) реализует максимальную префиксную сумму  $x_1 \circ x_2 \circ \dots \circ x_n$  с глубиной  $\lceil \log_2 n \rceil$ .



Майкл Джон Фишер  
Йельский университет, с 1981

<sup>3)</sup>Строго говоря, в [173] указана более слабая оценка сложности  $2k2^k$ .

Метод Ладнера—Фишера комбинирует два варианта применения принципа деления пополам: разбиение множества переменных на группы со старшими и младшими индексами, и разбиение на группы с четными и нечетными индексами.

Конструкция схемы  $\Pi_0(n)$  следует первому варианту. Схема устроена так:

- а) суммы  $\sigma_i = x_1 \circ \dots \circ x_i$ ,  $1 \leq i \leq \lceil n/2 \rceil$ , вычисляются схемой  $\Pi_1(\lceil n/2 \rceil)$ ;
- б) суммы  $\tau_i = x_{\lceil n/2 \rceil + 1} \circ \dots \circ x_i$ ,  $\lceil n/2 \rceil < i \leq n$ , вычисляются схемой  $\Pi_0(\lfloor n/2 \rfloor)$ ;
- в) недостающие суммы  $x_1 \circ \dots \circ x_i$ ,  $\lceil n/2 \rceil < i \leq n$ , вычисляются как  $\sigma_{\lceil n/2 \rceil} \circ \tau_i$ .

При  $k \geq 1$  конструкции схем  $\Pi_k(n)$  получаются из четно-нечетного разбиения:

- а) вычисляются суммы  $z_i = x_{2i-1} \circ x_{2i}$ ,  $1 \leq i \leq \lfloor n/2 \rfloor$ . При нечетном  $n$  дополнительно положим  $z_{\lceil n/2 \rceil} = x_n$ ;
- б) вычисляются суммы  $\sigma_i = x_1 \circ \dots \circ x_{2i}$ ,  $1 \leq i \leq \lfloor n/2 \rfloor$ , и сумма  $x_1 \circ \dots \circ x_n$  как  $z_1 \circ \dots \circ z_j$  схемой  $\Pi_{k-1}(\lceil n/2 \rceil)$ ;
- в) недостающие суммы  $x_1 \circ \dots \circ x_{2i-1}$ ,  $2 \leq i \leq \lfloor n/2 \rfloor$ , вычисляются как  $\sigma_{i-1} \circ x_{2i-1}$ .

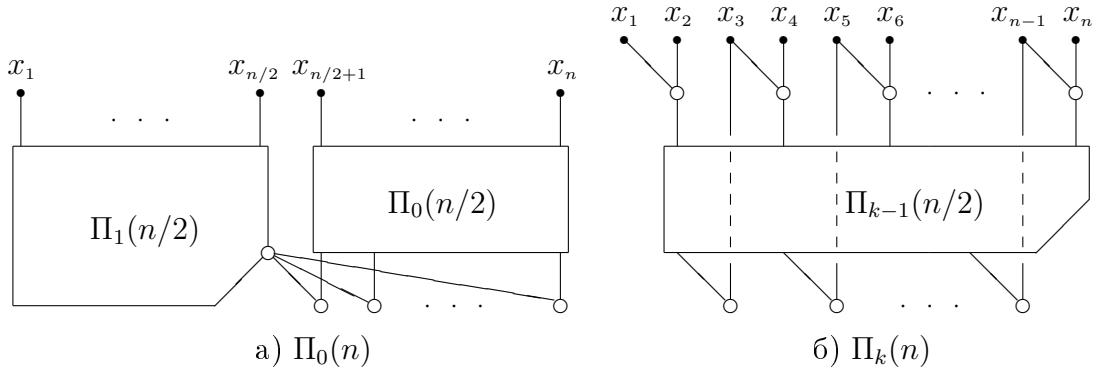


Рис. 2.1: Префиксные схемы Ладнера—Фишера

Описанные конструкции схем изображены на рис. 2.1 (округления в индексах и аргументах опущены). Несложно проверить, что схемы  $\Pi_k(n)$  действительно удовлетворяют условиям (\*), (\*\*).

Обозначим через  $\{\Phi_k\}$  последовательность чисел Фибоначчи:  $\Phi_1 = \Phi_2 = 1$ ,  $\Phi_{k+1} = \Phi_k + \Phi_{k-1}$ . Пусть  $P_k(n)$  означает минимальную сложность префиксной схемы на  $n$  входах при ограничении глубины  $\lceil \log_2 n \rceil + k$ .

**Теорема 2.5 ([251]).** При  $k \leq m$  выполнено

$$C_{\{o\}}(\Pi_k(2^m)) = 2(1 + 2^{-k})2^m - \Phi_{5+m-k} + 1 - k. \quad (2.9)$$

Следовательно,  $P_k(n) \leq 2(1 + 2^{-k})n$ .

► По построению,

$$C(\Pi_0(n)) = C(\Pi_0(\lfloor n/2 \rfloor)) + C(\Pi_1(\lceil n/2 \rceil)) + \lfloor n/2 \rfloor, \quad (2.10)$$

$$C(\Pi_k(n)) = C(\Pi_{k-1}(\lceil n/2 \rceil)) + 2\lfloor n/2 \rfloor - 1, \quad (2.11)$$

откуда по индукции легко выводятся требуемые соотношения. ■

В частности, в наиболее интересном случае  $k = 0$  получаем

**Следствие 2.1.**  $P_0(2^m) \leq 4 \cdot 2^m - \Phi_{m+5} + 1 \sim 4 \cdot 2^m$ .

- Автором в [84, 309] установлена точная оценка

$$P_0(2^m) \leq 3.5 \cdot 2^m - (8.5 + 3.5(m \bmod 2))2^{\lfloor m/2 \rfloor} + m + 5, \quad (2.12)$$

которая достигается в некоторых полугруппах  $(G, \circ)$ . Как следствие (из конструкции),  $P_0(n) \leq 3.5n$  при любом  $n$ . Также в [84, 309] точно установлено минимально возможное значение сложности универсальных префиксных схем, удовлетворяющих условиям (\*), (\*\*) в случае  $n = 2^m$ . Оказывается, что способ построения схем  $\Pi_k(2^m)$  в методе Ладнера—Фишера оптимален при всех  $k$  за исключением  $k = 1$ . Открытым остается вопрос о существовании более экономных коммутативных префиксных схем<sup>4</sup>). В [84, 309] также показано, что в частном случае группы  $(\mathbb{B}, \oplus)$  верхняя оценка (2.12) может быть понижена до  $36n/11$ .

Параллельные префиксные схемы имеют многочисленные приложения (см., например, обзор [146]), самым известным из которых является конструкция префиксного сумматора. Этот способ восходит к работе Ю. П. Офмана [60]. Преимуществом такого подхода является возможность построения сумматоров с различными полезными свойствами в зависимости от типа выбранной префиксной схемы (которых известно множество). В частности, легко добиться логарифмической глубины, сохраняя линейный порядок сложности (глубина стандартного сумматора линейна).

Напомним, что центральной частью сумматора  $n$ -разрядных чисел является вычисление системы функций (переносов)

$$c_i = F_i(X, Y) = y_{i-1} + x_{i-1}(y_{i-2} + \dots + x_2(y_1 + x_1 y_0) \dots), \quad i = 1, \dots, n, \quad (2.13)$$

см. (1.2) и (1.3).

Введем операцию  $\circ$  на векторах высоты 2:

$$\begin{pmatrix} f_1 \\ p_1 \end{pmatrix} \circ \begin{pmatrix} f_2 \\ p_2 \end{pmatrix} = \begin{pmatrix} f_2 + p_2 f_1 \\ p_2 p_1 \end{pmatrix}. \quad (2.14)$$

Легко убедиться, что эта операция является ассоциативной:

$$\begin{pmatrix} f_2 + p_2 f_1 \\ p_2 p_1 \end{pmatrix} \circ \begin{pmatrix} f_3 \\ p_3 \end{pmatrix} = \begin{pmatrix} f_3 + p_3 f_2 + p_3 p_2 f_1 \\ p_3 p_2 p_1 \end{pmatrix} = \begin{pmatrix} f_1 \\ p_1 \end{pmatrix} \circ \begin{pmatrix} f_3 + p_3 f_2 \\ p_3 p_2 \end{pmatrix}.$$

Теперь формулу (2.13) для функции переноса можно записать как

$$\begin{pmatrix} F_i \\ x_{i-1} \dots x_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} y_1 \\ x_1 \end{pmatrix} \circ \dots \circ \begin{pmatrix} y_{i-1} \\ x_{i-1} \end{pmatrix}. \quad (2.15)$$

Таким образом, задача вычисления всех переносов сводится к реализации системы префиксных сумм вида (2.8).

Одна операция  $\circ$  реализуется схемой над  $B_2$  сложности 3 и глубины 2. Поэтому на основе схемы, вычисляющей  $n$  префиксных сумм со сложностью  $C$  и глубиной  $D$ , можно построить схему сумматора сложности  $3(C+n)-1$  и глубины  $2D+2$ .

<sup>4</sup>Т.е. над группами  $(G, \circ)$  с коммутативной операцией  $\circ$ .

## Параллельные схемы сумматоров. Метод Храпченко /2



Валерий Михайлович  
Храпченко

Институт прикладной  
математики АН СССР/РАН,  
Москва, с 1966 по 2019

ЦИЯ ВИДА

Глубина префиксного  $n$ -разрядного сумматора не может быть меньше  $2 \log_2 n$ . Более аккуратное исследование выражений (2.13) привело В. М. Храпченко [103] к построению сумматора асимптотически оптимальной глубины  $\sim \log_2 n$ . Для общности будем полагать, что вычисления проводятся над монотонным арифметическим базисом  $\mathcal{A}_+$ .

**Теорема 2.6** ([103]). Для функции  $F_n$  из (2.13) справедливо  $D_{\mathcal{A}_+}(F_n) \leq \log_2 n + \sqrt{2 \log_2 n} + O(1)$ .

► Добавим в рассмотрение функции

$$F_{r,m}(X, Y) = x_{m+r-1} \cdot \dots \cdot x_{m+1} \cdot x_m \cdot F_m(X, Y). \quad (2.16)$$

В частности,  $F_{0,m} = F_m$ . Из (2.13) вытекает декомпозиция вида

$$F_{r,m+n} = F_{r,m} + F_{r+m,n} \quad (2.17)$$

(аргументы функций мы здесь для краткости опускаем). Обозначим  $d(s, k) = D(F_{s \cdot 2^k, 2^k})$ . Очевидно,  $d(s_1, k) \leq d(s_2, k)$  при  $s_1 \leq s_2$ . Поэтому, как следствие из (2.16) и (2.17),

$$d(2^l, k) \leq \max\{k + l, d(0, k)\} + 1, \quad (2.18)$$

$$d(s, k) \leq d(2s + 1, k - 1) + 1. \quad (2.19)$$

**Лемма 2.2.** При  $C_l^2 < m \leq C_{l+1}^2$  справедливо  $d(0, m) \leq m + l + 1$ .

► Неравенство доказывается индукцией по  $m$  с базой  $m = 1$ . Для доказательства индуктивного перехода от  $m - 1$  к  $m = C_l^2 + r$ , применяя  $r$  раз неравенство (2.19), а затем (2.18), получаем

$$d(0, m) \leq d(2^r - 1, C_l^2) + r \leq C_{l+1}^2 + r + 1 = m + l + 1.$$

□

Для завершения доказательства теоремы остается подставить в лемму 2.2 значение  $m = \lceil \log_2 n \rceil$  и заметить, что условие  $C_l^2 < m \leq C_{l+1}^2$  означает  $l = \lceil (\sqrt{1 + 8m} - 1)/2 \rceil$ . ■

При реализации переносов по правилам (1.3) получаем

**Следствие 2.2.**  $D_{\mathcal{B}_0}(\Sigma_n) \leq \log_2 n + \sqrt{2 \log_2 n} + O(1)$ .

- В [246] С. Р. Косараю доказал нижнюю оценку  $D_{\mathcal{A}_+}(F_n) \geq \log_2 n + \sqrt{(2 - o(1)) \log_2 n}$ . Следовательно, результат теоремы 2.6 не может быть существенно улучшен без дополнительных предположений относительно свойств полукольца, в котором производятся вычисления. М. И. Гринчук показал, что в булевом полукольце  $\{\mathbb{B}, \vee, \wedge\}$  оценка может быть понижена [15], см. далее на стр. 96. Открыт вопрос о возможности уточнения оценки в поле  $\mathbb{F}_2$ .

Прямое применение формул теоремы 2.6 приводит к сумматору нелинейной сложности. Несколько модифицировав способ вычисления, Храпченко [103] ценой увеличения глубины в пределах  $\log_2 n + O(\sqrt{\log n})$  построил  $n$ -разрядный сумматор линейной сложности  $(12 + o(1))n$  в базисе  $\mathcal{B}_0$ . В работе [11] показано, что сумматор такой сложности можно реализовать с глубиной  $\log_2 n + \sqrt{(2 + o(1)) \log_2 n}$ . В базисе  $\mathcal{B}_2$  сложность сумматоров понижается до  $(8 + o(1))n$ .

## Другие приложения

Область применения метода в теории быстрых вычислений настолько обширна, что сколько-нибудь подробное описание всевозможных приложений потянет на целую книгу, а то и не на одну. Руководствуясь принципом уважения к информации<sup>5</sup>, только перечислим кратко еще несколько важных задач, где метод работает эффективно.

**Переход между системами счисления.** Простой и асимптотически быстрый алгоритм перевода записи числа между системами счисления с разными основаниями предложен А. Шёнхаге<sup>6</sup>.

Для перевода числа  $A$  из  $b$ -ичной системы в двоичную оно разбивается пополам,  $A = A_1 b^k + A_0$ , «половинки»  $A_0$  и  $A_1$  переводятся к двоичному представлению рекурсивным вызовом алгоритма, для числа  $b^k$  двоичное представление можно считать предвычисленным; остается выполнить умножение и сложение двоичных чисел. В обратную сторону,  $n$ -разрядное двоичное число  $A$  делится с остатком на степень  $b^k \asymp 2^{n/2}$ , получаем  $A = A_1 b^k + A_0$ , остается перевести в  $b$ -ичное представление половинки  $A_0$  и  $A_1$ .

Таким образом, для сложности оператора  $T_{n,b}$  перевода числа размера  $< 2^n$  из  $b$ -ичного представления в двоичное, и для сложности обратного оператора имеют место рекуррентные соотношения

$$\mathsf{C}(T_{n,b}) \leq 2\mathsf{C}(T_{n/2,b}) + \mathsf{C}(M_{n/2}) + O(n), \quad \mathsf{C}(T_{n,b}^{-1}) \leq 2\mathsf{C}(T_{n/2,b}^{-1}) + \mathsf{C}(QR_{n,n/2}),$$

где  $QR_{n,m}$  — оператор деления с остатком  $n$ -разрядного числа на  $m$ -разрядное.

В терминах сложности умножения<sup>7</sup>  $M(n)$  соотношения разрешаются как  $\mathsf{C}(T_{n,b}), \mathsf{C}(T_{n,b}^{-1}) \preccurlyeq M(n) \log n$ , поскольку  $\mathsf{C}(QR_{2n,n}) \preccurlyeq M(n)$  (доказано С. Куком [172]; см. далее на стр. 53). В силу результата Д. Харви и Ж. ван дер Хувена [211],  $M(n) \preccurlyeq n \log n$ .

**Быстрое вычисление наибольшего общего делителя.** Идея деления пополам позволила радикально понизить сложность классических евклидовых алгоритмов вычисления НОД. Усовершенствовав исходный метод Д. Кнута [244], А. Шёнхаге в [301] доказал оценку<sup>8</sup>  $\mathsf{C}(\text{НОД}_n) \preccurlyeq M(n) \log n$ .

Идея метода заключается в следующем. Вводится оператор  $\text{ПНОД}_n(A, B) = (a, b, M)$ , который по двум  $n$ -разрядным числам  $A, B$  строит пару чисел  $a, b$  длины  $\approx n/2$  с сохранением НОД, а также матрицу перехода  $M$ , элементы которой также имеют размер  $\approx n/2$ . Например, можно требовать выполнения условий

$$\text{НОД}(a, b) = \text{НОД}(A, B), \quad a \geq 2^{n/2} > b, \quad \begin{pmatrix} a \\ b \end{pmatrix} = M \begin{pmatrix} A \\ B \end{pmatrix}.$$

---

<sup>5</sup>Сокращать объем изложения там, где материала много, и воспроизводить материал целиком там, где его мало (сформулирован А. Т. Фоменко применительно к анализу исторических хроник).

<sup>6</sup>По-видимому, метод так и не был опубликован автором. Цитируется по [24].

<sup>7</sup>Удовлетворяющей условиям  $M(n) \geq \mathsf{C}(M_n)$  и  $M(x+y) \geq M(x) + M(y)$  при всех  $x, y$ .

<sup>8</sup>Строго говоря, алгоритмы НОД обычно формулируются для модели программ с ветвлени-ями. Переход к схемной реализации выполняется подобно тому, как это сделано для бинарного алгоритма НОД выше, и не влияет на порядок сложности.

Так вычисление  $\text{НОД}(A, B)$  сводится к выполнению операции  $(a, b, M) = \text{ПНОД}_n(A, B)$  и вычислению  $\text{НОД}(b, a \bmod b)$  чисел половинного размера. Поэтому

$$\mathcal{C}(\text{НОД}_n) \leq \mathcal{C}(\text{НОД}_{n/2}) + \mathcal{C}(\text{ПНОД}_n) + \mathcal{C}(QR_{2n,n}) + O(n \log n), \quad (2.20)$$

где последнее слагаемое учитывает специфику реализации алгоритма схемами (как в доказательстве теоремы 1.4).

Центральной частью метода является вычисление оператора ПНОД. В основе лежит наблюдение Д. Лехмера [254]: первые шаги алгоритма Евклида определяются старшими разрядами чисел  $A$  и  $B$ . Примерная последовательность действий при вычислении  $\text{ПНОД}_n(A, B)$  такова. Пусть  $A = A_1 2^{n/2} + A_0$  и  $B = B_1 2^{n/2} + B_0$ .

1. Вычисляем  $(a', b', M_1) = \text{ПНОД}_{n/2}(A_1, B_1)$ . Используя полученную матрицу перехода, можно получить новую пару чисел  $A', B'$ , которые удовлетворяют условию  $\text{НОД}(A', B') = \text{НОД}(A, B)$  и имеют длину в пределах  $3n/4$  разрядов. В первом приближении  $\begin{pmatrix} A' \\ B' \end{pmatrix} = M_1 \begin{pmatrix} A \\ B \end{pmatrix}$ , но может потребоваться коррекция на несколько делений с остатком вперед или даже назад.

2. Записывая  $A' = A_3 2^{n/4} + A_2$  и  $B' = B_3 2^{n/4} + B_2$ , выполняем еще один рекурсивный вызов операции ПНОД. Вычисляется  $(a'', b'', M_2) = \text{ПНОД}_{n/2}(A_3, B_3)$ . При помощи матрицы перехода  $M_2$  из  $A', B'$  получается искомая пара  $a, b$  длины  $\approx n/2$  разрядов, а также матрица перехода  $M$ . С точностью до необходимой коррекции,  $\begin{pmatrix} a \\ b \end{pmatrix} = M_2 \begin{pmatrix} A' \\ B' \end{pmatrix}$  и  $M = M_2 M_1$ .

Необходимость коррекции результатов на каждом шаге и дополнительные проверки существенно затрудняют аккуратное изложение метода. Принципиально, нужно следить, чтобы промежуточные шаги алгоритма не порождали слишком маленькие числа. Это бы приводило к невозможности применения правила Лехмера: результат зависел бы от тех младших разрядов, которые в расчетах не учитывались.

В итоге для сложности получается рекуррентное соотношение

$$\mathcal{C}(\text{ПНОД}_n) \leq 2\mathcal{C}(\text{ПНОД}_{n/2}) + O(\mathcal{C}(M_n) + \mathcal{C}(QR_{2n,n}) + n \log n),$$

которое приводит к  $\mathcal{C}(\text{ПНОД}_n) \leq M(n) \log n$  и вкупе с (2.20) — к  $\mathcal{C}(\text{НОД}_n) \leq M(n) \log n$ .

К настоящему времени метод получил ряд модификаций. Сам Шёнхаге в [304] предложил использовать для вычисления НОД итерации деления с выбором наибольшего остатка, что позволило упростить контроль скорости сходимости и проверку корректности метода. Этот алгоритм вместе с предложенным Н. Мёллером более эффективным ПНОД-вариантом описан в [267]. Д. Штеле и П. Циммерман [312] построили быстрый алгоритм НОД на базе правостороннего деления с остатком (в этом случае контроль параметров метода также выполняется проще). Наконец, метод Д. Бернштейна и Б.-Й. Янга [141] по сути базируется непосредственно на итерациях бинарного алгоритма НОД (см. стр. 13). Метод [141] имеет приоритет перед перечисленными выше ввиду малого числа ветвлений и максимально простого контроля сходимости и корректности.

Переложение быстрого алгоритма НОД для кольца многочленов выполнено Р. Монком в [266], более современный вариант метода описан в [141], а существенно оптимизированная версия — в [222]. Анализ полиномиального алгоритма несколько проще ввиду отсутствия переносов при сложении (проще коррекция промежуточных результатов). Схемы для НОД многочленов над кольцами общего вида неизбежно должны включать не только арифметические операции, но также, например, операции сравнения.

**Сортировка и монотонные схемы для пороговых функций.** Монотонные пороговые булевы функции  $n$  переменных в совокупности образуют оператор *сортировки* (булевы набора)  $\text{SORT}_n = (T_n^1, T_n^2, \dots, T_n^n)$ , где  $T_n^k = (x_1 + \dots + x_n \geq k)$  — монотонная симметрическая функция  $n$  переменных с порогом  $k$ . Поэтому задачи вычисления пороговых функций и сортировки (или выбора, если речь идет об одной компоненте оператора) оказываются тесно связанными.

Популярной моделью для реализации сортировки являются *схемы компараторов*. Компарататор — это подсхема, упорядочивающая два входных числа; в булевом случае преобразование

имеет вид  $(x, y) \rightarrow (x \wedge y, x \vee y)$ . В схемах компараторов запрещены ветвления выходов компараторов<sup>9)</sup>, также см. ниже на стр. 61.

В полном базисе тривиально выполняется  $C(SORT_n) \asymp n$ . Оценки сложности монотонных схем получаются переносом известных результатов о сложности схем компараторов. Наивный метод (сравнить все элементы попарно) приводит к оценке  $C_{B_M}(SORT_n) \asymp n^2$ . Существенно более эффективный способ сортировки с элегантным применением принципа деления пополам предложил К. Бэтчер [128]. Метод заключается в следующем. Сортируемый набор делится пополам, выполняется сортировка в каждой из половин, затем выполняется слияние упорядоченных поднаборов. Таким образом,

$$C_{B_M}(SORT_{2n}) \leq 2C_{B_M}(SORT_{2n}) + C_{B_M}(СЛ_{n,n}),$$

где через  $СЛ_{m,n}$  обозначается оператор слияния<sup>10)</sup> упорядоченных наборов длины  $m$  и  $n$ .

Слияние двух наборов тоже выполняется рекурсивно методом деления пополам. Отдельно выполняется слияние всех четных элементов обоих наборов и всех нечетных. Для совместной сортировки полученных двух списков оказывается достаточным всего  $n-1$  сравнений. Поэтому

$$C_{B_M}(СЛ_{n,n}) \leq 2C_{B_M}(СЛ_{n/2,n/2}) + 2(n-1),$$

откуда  $C_{B_M}(СЛ_{n,n}) \asymp n \log n$ , следовательно,  $C_{B_M}(SORT_n) \asymp n \log^2 n$ . Подробное изложение метода см. в [25, 110].

Впоследствии М. Айтаи, Я. Комлош и Э. Семереди [119, 120] доказали, что на самом деле  $C_{B_M}(SORT_n) \asymp n \log n$  (доказывается в теореме 5.6 далее), но при разумных значениях  $n$  метод Бэтчера имеет приоритет.

При вычислении отдельных функций  $T_n^k$  с небольшими порогами  $k$  хорошо работает асимптотически оптимальный в модели схем компараторов метод Э. Яо [333]. В нем элементы разбиваются на пары, которые упорядочиваются. Заметим, что среди младших элементов пар содержится не более  $k/2$  из  $k$  наибольших элементов входного набора. Поэтому выбираем  $k$  наибольших элементов среди старших элементов пар и  $\lfloor k/2 \rfloor$  наибольших элементов среди младших элементов пар, и затем находим среди них  $k$  наибольших элементов. Можно записать рекуррентное соотношение

$$C_{B_M}(T_n^1, \dots, T_n^k) \leq n + C_{B_M}(T_{n/2}^1, \dots, T_{n/2}^k) + C_{B_M}(T_{n/2}^1, \dots, T_{n/2}^{k/2}) + C_{B_M}(СЛ_{k,k/2}).$$

Выполненный автором [93] аккуратный перенос метода Яо на модель монотонных схем<sup>11)</sup> приводит к справедливой при постоянном или медленно растущем  $k$  оценке

$$C_{B_M}(T_n^k) \lesssim (\lfloor \log_2 k \rfloor + \lfloor \log_2(4k/3) \rfloor)n.$$

Объединение результатов В. Е. Алексеева [2] или Яо [333] (конструкции схем компараторов для пороговых функций) с результатом [119, 120], дает общую оценку  $C_{B_M}(T_n^k) \asymp n \log k$ . Наилучшая универсальная оценка, справедливая при любых значениях порога,  $C_{B_M}(T_n^k) \lesssim 6n \log_3 n$  получена С. Джимбо и А. Марукой в [228].

**Мультипликативная сложность многочленов.** Рассмотрим задачу вычисления значения произвольного многочлена  $f(x) = a_n x^n + \dots + a_1 x + a_0$  схемами в полном арифметическом базисе  $\mathcal{A}^{\mathbb{R}}$  или  $\mathcal{A}^{\mathbb{C}}$ . Простейший способ вычисления многочлена — это схема Горнера:

$$f(x) = (\dots ((a_n x + a_{n-1}) x + \dots) x + a_0.$$

<sup>9)</sup>Схему компараторов можно изобразить как последовательность  $n$ -элементных наборов, удовлетворяющую условиям: 1) начинается с входного вектора; 2) каждый следующий набор получается из предыдущего упорядочением одной пары элементов; 3) заканчивается линейно упорядоченным набором (либо, в более общем случае, содержит требуемый частичный порядок).

<sup>10)</sup>Это частично определенный булев оператор.

<sup>11)</sup>При этом дополнительно используется специальный метод синтеза схем для функции  $T_n^2$ , см. теорему 11.1.

Она использует  $n$  сложений и  $n$  умножений. Первая оценка (для числа аддитивных операций) в общем случае неулучшаема, как показали Э. Г. Белага [5] и В. Я. Пан [61, 62]. Они же (для полей  $\mathbb{C}$  и  $\mathbb{R}$  соответственно) доказали, что можно всегда ограничиться  $n/2 + O(1)$  умножениями. Более того, обе оценки с точностью до  $O(1)$  достигаются на одной схеме.

Методы Пана и Белаги нетривиальны и требуют довольно сложной предварительной подготовки коэффициентов<sup>12</sup>). Изящный способ вычисления, приводящий лишь к немногим худшей оценке  $n/2 + O(\log n)$  умножений, предложен М. Рабиным и Ш. Виноградом [290], см. также [280]. В основе метода лежит наблюдение о том, что нормированный (т. е. со старшим коэффициентом 1) многочлен  $f(x)$  степени  $2^{k+1} - 1$  может быть представлен в виде

$$f(x) = (x^{2^k} + a)f_1(x) + f_0(x), \quad (2.21)$$

где  $f_0, f_1$  — нормированные многочлены степени  $2^k - 1$ . Поэтому, если отдельно вычислены все степени  $x^2, x^{2^2}, \dots, x^{2^k}$ , то любой промежуточный многочлен степени  $2^m - 1$  вычисляется по формулам (2.21) при помощи  $2^{m-1} - 1$  умножений.

Число нескаллярных умножений, необходимых для вычисления многочлена степени  $n$ , в общем случае оценивается снизу как  $\sqrt{n}$  [280]. Наилучшая верхняя оценка, полученная М. Паттерсоном и Л. Стокмайером [280], имеет вид  $\sqrt{2n} + O(\log n)$ . Метод идеально близок к методу [290] и также эксплуатирует идею деления пополам. Нормированный многочлен  $f(x)$  степени  $p(2^{k+1} - 1)$  может быть представлен в виде

$$f(x) = (x^{p2^k} + c(x))f_1(x) + f_0(x), \quad (2.22)$$

где  $f_0, f_1$  — нормированные многочлены степени  $p(2^k - 1)$  и  $\deg c < p$ . Если вычислены все степени  $x^2, x^3, \dots, x^p$  и  $x^{2p}, x^{2^2p}, \dots, x^{2^kp}$ , то каждый промежуточный многочлен степени  $p(2^m - 1)$  вычисляется по формулам (2.22) при помощи  $2^{m-1} - 1$  умножений. Требуемую оценку получаем при  $p \approx \sqrt{n/2}$ .

---

<sup>12</sup>Если коэффициенты многочлена тоже являются входами схемы, иначе говоря, предварительная обработка невозможна, то схема Горнера оптимальна, как показал Пан в работе [63].

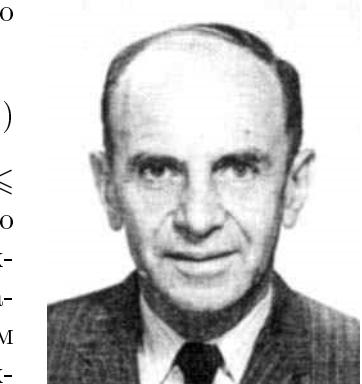
# Глава 3

## Метод общей части

[C]

Суть метода — в выделении фрагментов, которые могут быть использованы многократно. Удивительно, что одна лишь эта идея ведет к ряду нетривиальных асимптотически точных результатов.

### Аддитивные цепочки. Метод Брауэра



Альфред Теодор

Брауэр

Университет Сев. Каролины,  
с 1942 по 1964

*Аддитивные цепочки* — это (универсальные) аддитивные схемы с одним входом и одним выходом. Они реализуют преобразования  $x \rightarrow nx$ , но традиционно входом цепочки считается константа 1, тогда на выходе получается натуральное число  $n$ .

Отталкиваясь от двоичного представления числа, аддитивную цепочку для  $n = [n_k, n_{k-1}, \dots, n_0]_2$  можно построить, следуя формуле

$$n = (\dots (2n_k + n_{k-1})2 + \dots + n_1)2 + n_0. \quad (3.1)$$

Это бинарный метод — он приводит к оценке  $L(n) \leq \log_2 n + \nu(n) - 1 < 2 \log_2 n$ , где  $\nu(n)$  — вес числа  $n$  (число единиц в двоичной записи). При этом очевидная нижняя оценка имеет вид  $L(n) \geq \log_2 n$  (на каждом шаге размер вычисленного числа увеличивается максимум вдвое). А. Брауэр [155] заметил, что на самом деле нижняя оценка асимптотически точна.

**Теорема 3.1** ([155]).  $L(n) \leq \log_2 n + (1 + o(1)) \frac{\log_2 n}{\log_2 \log n}$ .

► Пусть  $n < 2^{tk}$ . Организовав разряды числа  $n$  в матрицу  $A$  размера  $t \times k$ , можно записать

$$n = (1, 2^k, 2^{2k}, \dots, 2^{(t-1)k}) \cdot \begin{pmatrix} n_0 & n_1 & \cdots & n_{k-1} \\ n_k & n_{k+1} & \cdots & n_{2k-1} \\ \vdots & \vdots & \ddots & \vdots \\ n_{(t-1)k} & n_{(t-1)k+1} & \cdots & n_{tk-1} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ \vdots \\ 2^{k-1} \end{pmatrix}. \quad (3.2)$$

Центральным местом метода является выбор параметров с условием  $t > 2^k$ . Таким образом, в матрице  $A$  оказывается много повторяющихся строк.

**Лемма 3.1.** Для произвольной булевой  $t \times k$  матрицы  $A$  выполнено  $L(A) \leq 2^k - k - 1$ .

▷ Имеется всего  $2^k - k - 1$  различных строк длины  $k$  и веса  $\geq 2$ . Несложно построить процедуру вычисления всех таких строк, используя по одной операции на каждую.  $\square$

Выполняя вычисления по формуле (3.2) справа налево и используя результат леммы 3.1, получаем

$$L(n) \leq (k-1) + (2^k - k - 1) + (k+1)(t-1),$$

подразумевая, что для последнего умножения используется  $2^k$ -арный вариант формулы (3.1). Остается выбрать  $k \approx \log_2 \log_2 n - 2 \log_2 \log_2 \log_2 n$ .  $\blacksquare$

- П. Эрдёш [186] показал, что для почти всех  $n$  выполнено

$$L(n) \geq \log_2 n + (1 - o(1)) \frac{\log_2 n}{\log_2 \log n}.$$

Более того, как установили В. В. и Д. В. Кочергини [31], в качестве  $o(1)$  и в верхней оценке теоремы 3.1, и в нижней оценке Эрдёша можно указать величину  $(2 + o(1)) \frac{\log_2 \log \log n}{\log_2 \log n}$ . Наилучшая абсолютная нижняя оценка доказана А. Шёнхаге в [302] и имеет вид  $L(n) \geq \log_2 n + \log_2 \nu(n) - 2.13$  (точность этого результата подчеркивает легко проверяемый факт: оценка  $\lfloor \log_2 n \rfloor + \lceil \log_2 \nu(n) \rceil$  достигается при любых значениях  $\nu(n)$ ).

### Асимптотически минимальные схемы для булевых функций c

Произвольную булеву функцию  $p$  переменных можно задать таблицей ее значений размера  $2^n$  бит, и в общем случае более короткого кода не существует<sup>1</sup>). Известный результат К. Шеннона [311] показывает, что для кодирования функций (контактными) схемами достаточно  $O(2^n/n)$  элементов — таким образом, основной объем информации заключен в топологии схем. Д. Маллер [272] распространял метод Шеннона на схемы из функциональных элементов, а вскоре О. Б. Лупанов разработал асимптотически оптимальный метод синтеза [42]. Метод Лупанова произвел сенсацию: трудно было рассчитывать, что на первый взгляд грубая мощностная нижняя оценка вида  $\asymp 2^n/n$  окажется настолько точной и конструктивно достижимой в асимптотическом смысле. В основе метода лежит простой результат Лупанова об аддитивной сложности булевых матриц [41].

**Лемма 3.2** ([41]). Пусть  $q \succsim \log p$ . Тогда для произвольной булевой  $p \times q$  матрицы  $A$  справедливо

$$L(A) \leq \left(1 + O\left(\frac{\log \log p}{\log p}\right)\right) \frac{pq}{\log_2 p}.$$

<sup>1</sup>Поскольку различных функций  $n$  переменных —  $2^{2^n}$ .

► Разделим матрицу на вертикальные полосы ширины  $k$ :  $A = (A_1 | A_2 | \dots | A_s)$ , где  $s = \lceil q/k \rceil$ . Используя оценку леммы 3.1, получаем

$$\mathsf{L}(A) \leq \mathsf{L}(A_1) + \dots + \mathsf{L}(A_s) + (s-1)p \leq s2^k + pq/k.$$

Осталось выбрать  $k \approx \log_2 p - \log_2 \log_2 p$ . □

Заметим, что лемма использует тот же прием, что и в теореме 3.1, — сведение к вычислению сверхузких матриц. Так использование повторяющихся фрагментов сумм в разных строках позволяет уйти от тривиальной оценки  $\mathsf{L}(A) \leq p(q-1)$ .

- В работе [41] оценка сложности доказывается для вентильных схем (глубины 2), но доказательство для аддитивных схем то же самое. Легко проверить, что оценка леммы 3.2 асимптотически точна при  $\log q \prec \log p$  [41] (т.е. когда матрица  $A$  достаточно узкая).

**Теорема 3.2 ([42]).** Для произвольной булевой функции  $f$  от  $n$  переменных

$$\mathsf{C}_{\mathcal{B}_2}(f) \leq \left(1 + O\left(\frac{\log n}{n}\right)\right) \frac{2^n}{n}.$$

► Разделим множество  $n$  переменных на две группы  $X, Y$ , где  $|X| = k$  и  $|Y| = n - k$ . Обозначим через  $X_J = \prod_{j \in J} x_j$ ,  $J \subset \{1, \dots, k\}$ , произведения (мономы) переменных первой группы, а через  $Y_I = \prod_{i \in I} y_i$ ,  $I \subset \{1, \dots, n-k\}$ , — мономы переменных второй группы. Любую булеву функцию  $f(X, Y)$  можно представить в виде многочлена Жегалкина

$$f(X, Y) = \bigoplus_I Y_I \bigoplus_J f_{I,J} X_J, \quad (3.3)$$

где  $F = (f_{I,J})$  — булева матрица размера  $2^{n-k} \times 2^k$  (отметим, что на самом деле формулы (3.2) и (3.3) аналогичны).

Вычисление всех  $X_J$  выполняется линейным оператором размера  $2^k \times k$  над  $(\mathbb{B}, \wedge)$  — согласно лемме 3.1 он реализуется схемой сложности  $2^k$ . Аналогично, сложность вычисления всех  $Y_I$  не превосходит  $2^{n-k}$ . Сложность преобразования с матрицей  $F$  (над  $(\mathbb{B}, \oplus)$ ) оценим при помощи леммы 3.2. Еще  $2 \cdot 2^{n-k}$  операций требуется для умножений промежуточных сумм на мономы  $Y_I$  и для вычисления внешней суммы в (3.3). Таким образом,

$$\mathsf{C}_{\mathcal{B}_2}(f) \leq 3 \cdot 2^{n-k} + 2^k + \left(1 + O\left(\frac{\log(n-k)}{n-k}\right)\right) \frac{2^n}{n-k}.$$

При выборе  $k \approx 2 \log_2 n$  получаем утверждение теоремы. ■



Олег Борисович  
Лупанов

Московский университет,  
с 1959 по 2006

- В базисе  $\mathcal{B}_0$  получается такая же оценка, если вместо (3.3) использовать получаемую из ДНФ (дизъюнктивной нормальной формы) формулу  $f = \bigvee_I Y_I \bigvee_J f_{I,J} X_J$ , где  $X_J$  и  $Y_I$  — элементарные конъюнкции (произведения переменных и их отрицаний). Все элементарные конъюнкции  $k$  переменных вычисляются со сложностью  $\sim 2^k$  (см. лемму 3.5 ниже).

На самом деле, в работе [42] Лупанов получил более общий результат  $C_{\mathcal{B}}(f) \sim 2^n / ((s-1)n)$  для почти всех функций  $n$  переменных, где  $s$  — максимальное число существенных переменных у функций полного конечного базиса  $\mathcal{B}$ . Асимптотические оценки сложности высокой точности получил С. А. Ложкин [35, 38]: например, для базиса  $\mathcal{B} \in \{\mathcal{B}_0, \mathcal{B}_2\}$  оценка принимает вид

$$\left(1 + \frac{\log_2 n - O(1)}{n}\right) \frac{2^n}{n} \lesssim C_{\mathcal{B}}(\mathcal{P}_n) \lesssim \left(1 + \frac{\log_2 n + \log_2 \log n + O(1)}{n}\right) \frac{2^n}{n}.$$

Лемма 3.2, устанавливающая сложность класса булевых  $p \times q$  матриц, представляет самостоятельную ценность. Например, она позволяет обобщить результат теоремы 3.1 на случай вычисления нескольких чисел. Пусть  $n = \max_i n_i$  и  $s \log n \rightarrow \infty$ . Тогда

$$L(n_1, \dots, n_s) \leq \log_2 n + (1 + o(1)) \frac{s \log_2 n}{\log_2(s \log n)} + O(s). \quad (3.4)$$

▷ Пусть  $n < 2^{tk}$ . Транспонируя (3.2), можно записать

$$n_i = b A_i C^T, \quad b = (1, 2, 2^2, \dots, 2^{k-1}), \quad C = (1, 2^k, 2^{2k}, \dots, 2^{(t-1)k}),$$

где  $A_i$  — булева  $k \times t$  матрица, составленная из разрядов числа  $n_i$ . Тогда

$$\begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_s \end{pmatrix} = BAC^T, \quad B = \begin{pmatrix} b & 0 \dots 0 & \cdots & 0 \dots 0 \\ 0 \dots 0 & b & \cdots & 0 \dots 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 \dots 0 & 0 \dots 0 & \cdots & b \end{pmatrix}, \quad A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_s \end{pmatrix}.$$

Как следствие,

$$L(n_1, \dots, n_s) \leq L(B) + L(A) + L(C^T) \leq 2s(k-1) + L(A) + (t-1)k.$$

Если  $\log s \leq \log \log n$ , полагаем  $t \approx \log_2^2 \log_2 n$ . Иначе (т.е. при большом  $s$ ) выбираем  $k = 1$ . Далее сложность матрицы  $A$  (размера  $sk \times t$ ) оценивается с помощью леммы 3.2. □

Оценка (3.4) является частным случаем более общего результата Н. Пиппенджера [285] о сложности целочисленных матриц. Мощностным методом несложно проверить, что в классе всех наборов из  $s$  чисел величины  $\leq n$  указанная оценка неулучшаема [285]. В случае индивидуально заданных ограничений  $n_i \leq r_i$  С. Б. Гашков и В. В. Кочергин [12] уточнили оценку (3.4) до

$$L(n_1, \dots, n_s) \leq \log_2 n + (1 + o(1)) \frac{R}{\log_2 R} + O(s), \quad R = \log_2(r_1 \cdot \dots \cdot r_s). \quad (3.5)$$

Она также неулучшаема в своем классе. Впоследствии Кочергин уточнил остаточные члены в обеих оценках (3.4) и (3.5), см. [30].

## Схемы для линейных булевых операторов c

Лемма 3.2 устанавливает асимптотику сложности для класса достаточно узких булевых матриц. В наиболее интересном случае квадратных матриц асимптотически точный результат получается более тонким применением метода общей части, которое предложил Э. И. Нечипорук [55] (также см. [57]).

**Теорема 3.3** ([55]). Для произвольной булевой  $n \times n$  матрицы  $A$  справедливо

$$\mathsf{L}(A) \lesssim \frac{n^2}{2 \log_2 n}.$$

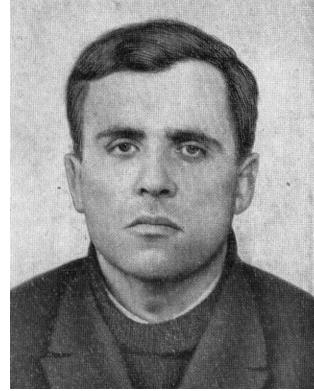
► Сначала докажем вспомогательную оценку.

**Лемма 3.3.** Для произвольной булевой  $p \times q$  матрицы  $A$  справедливо

$$\mathsf{L}(A) \leq |A|/2 + q + p^2.$$

▷ Единицы в каждом столбце матрицы разобьем на пары. Так произвольная строка  $\sigma_i$  представляется в виде суммы подстрок  $\sigma_{\{i,j\}}$ , общих с другими строками  $\sigma_j$ ,  $j \neq i$ , и непарной подстроки  $\sigma'_i$ . Обозначим через  $A_1$  суммарный вес подстрок  $\sigma_{\{i,j\}}$ ,  $i \neq j$ , и через  $A_2$  — суммарный вес подстрок  $\sigma'_i$ . Тогда  $2A_1 + A_2 = |A|$  и  $A_2 \leq q$ .

Вычислим все подстроки  $\sigma_{\{i,j\}}$  и  $\sigma'_i$ , а затем суммы в каждой строке. Это требует не более  $A_1 + A_2 + p^2 \leq |A|/2 + q + p^2$  операций.  $\square$



Эдуард Иванович  
Нечипорук  
Ленинградский университет,  
с 1960-х по 1970 г.

- Чуть более простое доказательство леммы получается применением принципа транспонирования (лемма 8.1).

Доказательство теоремы начинается так же, как в лемме 3.2. Разделим матрицу на вертикальные полосы ширины  $k$ :  $A = (A_1|A_2|\dots|A_s)$ , где  $s = n/k$  (для простоты полагаем  $k|n$ ). Вычисление сумм в полосах соответствует представлению матриц  $A_i$  в виде произведений  $B_i U_k$  матриц  $B_i$  размера  $n \times 2^k$ , имеющих по одной единице в каждой строке, и  $2^k \times k$  матриц  $U_k$ , составленных из всевозможных строк длины  $k$ :

$$A = (B_1|B_2|\dots|B_s) \cdot \begin{pmatrix} U_k & 0 & \cdots & 0 \\ 0 & U_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & U_k \end{pmatrix}.$$

Матрицу  $B = (B_1|B_2|\dots|B_s)$  разобьем на горизонтальные полосы высоты  $p$  и вычисление сумм в каждой полосе (это матрица размера  $p \times s2^k$  и веса  $ps$ ) выполним независимо, используя оценку леммы 3.3. В результате получаем

$$\mathsf{L}(A) \leq s\mathsf{L}(U_k) + |B|/2 + \lceil n/p \rceil (s2^k + p^2) = ns/2 + O(ns2^k/p + np).$$

Требуемая оценка следует при выборе  $k \approx \log_2 n - 3 \log_2 \log_2 n$  и  $p \asymp n/\log^2 n$ . ■

- На самом деле, Нечипорук [55] получил более общий результат, показав, что асимптотически точная оценка

$$\mathsf{L}(A) \leq (1 + \varepsilon) \frac{mn}{\log_2(mn)}, \quad \varepsilon = o(1), \tag{3.6}$$

сложности булевых  $m \times n$  матриц справедлива при  $\log_2 n \sim \alpha \log_2 m$ , где  $\alpha = r - 1 + r/s$  при  $r, s \in \mathbb{N}$ . Нижняя оценка  $L(A) \geq (1-\delta)mn/\log_2(mn)$ ,  $\delta \asymp \frac{\log \log n}{\log n}$ , почти для всех  $m \times n$  матриц  $A$  устанавливается простым мощностным рассуждением.

Н. Пиппенджер [284] снял ограничения на  $m, n$ , доказав справедливость (3.6) для любых  $m, n$ , где  $\log m \asymp \log n$ , при  $\varepsilon \asymp \sqrt{\frac{\log \log n}{\log n}}$ . Автор в [90] показал, что при  $m \in n^\mu \log^{\pm O(1)} n$ ,  $\mu \in \mathbb{Q}$ , можно полагать  $\varepsilon \asymp \frac{\log \log n}{\log n}$ , как и в нижней оценке.

Результат для булевых матриц влечет наиболее общую форму оценки сложности целочисленных матриц [285]. Пусть  $A$  —  $m \times n$  матрица с элементами  $\leq q$ , и  $mn \log q \rightarrow \infty$ . Тогда

$$L(A) \leq m \log_2 q + (1 + o(1)) \frac{mn \log_2 q}{\log_2(mn \log q)} + O(n). \quad (3.7)$$

Доказывается аналогично оценке сложности векторов (3.4), см. также [90].

### Сложность монотонных схем. Принцип локального кодирования c e

Проблема асимптотической сложности класса  $\mathcal{M}^n$  монотонных функций  $n$  переменных опирается на их эффективное перечисление (кодирование). Подходящую процедуру впервые построил Д. Клейтман в 1969 г. [242], установив, что  $|\mathcal{M}^n| = 2^{(1+o(1))C_n^{n/2}}$ . После этого стала возможной разработка оптимальных методов синтеза монотонных функций. Для схем в полных базисах соответствующий результат получил А. Б. Угольников [99]: в частности,

$$C_{B_0}(\mathcal{M}^n) \sim \frac{C_n^{n/2}}{n} \sim \frac{2^n}{n^{3/2} \sqrt{\pi/2}}.$$

Доказательство такой же асимптотической оценки в классе схем над монотонным базисом  $B_M$  далось значительно труднее. Эту задачу решил А. Е. Андреев [3].

- Асимптотика числа монотонных функций была найдена А. Д. Коршуновым в [28] очень трудным способом. Еще одно доказательство предложил А. А. Сапоженко в [76]. Подробнее об истории вопроса см. в обзоре [29].

Мы приведем более простой, хотя и нетривиальный, результат о порядке сложности монотонных схем, который был установлен Н. П. Редькиным в [68]. Упрощенное доказательство предложено А. В. Чашкиным в [111].

Метод доказательства иллюстрирует общий подход к построению оптимальных схем, предложенный О. Б. Лупановым [46] и названный им *принципом локального кодирования*. Идея состоит в представлении функции таким двоичным кодом, что значение функции на каждом конкретном наборе определяется только локальным фрагментом кода. Вычисление функции, таким образом, заключается (1) в определении нужного фрагмента и (2) в декодировании этого фрагмента. Оба этапа должны допускать простую реализацию.

Для кодирования монотонных функций используются разбиения булева куба на монотонные цепи и тот факт, что функция может только один раз поменять значение на каждой цепи.

**Теорема 3.4 ([68]).**  $C_{B_M}(\mathcal{M}^n) \leq C_n^{n/2}/n$ .

► Пусть  $n = 2k + m$ . Напомним, что булев куб  $\mathbb{B}^k$  можно разбить на  $C_k^{k/2}$  монотонных цепей (следствие из теоремы Дилуорса, см., например, [230]). Запишем

$$\mathbb{B}^{2k} = \mathbb{B}^k \times \mathbb{B}^k = \left( \bigcup_{\alpha} \alpha \right) \times \left( \bigcup_{\beta} \beta \right) = \bigcup_{\alpha \subset \mathbb{B}^k} \bigcup_{\beta \subset \mathbb{B}^k} (\alpha \times \beta),$$

где  $\alpha, \beta$  — монотонные цепи из (выбранного) оптимального разбиения  $\mathbb{B}^k$ .

**Лемма 3.4** ([68]). Число различных монотонных булевых функций, определенных на произведении монотонных цепей  $\alpha \times \beta$ , равно  $C_{|\alpha|+|\beta|}^{|\alpha|}$ .

► Пусть  $\alpha = (\alpha_1 \prec \dots \prec \alpha_p)$  и  $\beta = (\beta_1 \prec \dots \prec \beta_q)$ . Любой монотонной функции  $\varphi : \alpha \times \beta \rightarrow \mathbb{B}$  однозначно соответствует набор чисел  $0 \leq k_1 \leq \dots \leq k_p \leq q$ , где  $k_i$  — число элементов  $x \in \beta$ , для которых  $\varphi(\alpha_i, x) = 1$ . Число таких наборов равно  $C_{p+q}^p$ .  $\square$

Зафиксируем разбиение  $\mathbb{B}^{2k} = \bigcup I_j$ , где любое множество  $I_j$  (кроме, быть может, одного) является объединением произведений  $\alpha \times \beta$  с условием  $k + 2 \leq \sum(|\alpha| + |\beta|) \leq 2k + 2$ . Из леммы 3.4 следует, что число монотонных функций, определенных на  $I_j$ , не превосходит  $2^{2k+2}$ . Поскольку  $\sum_{\alpha, \beta \subset \mathbb{B}^k} (|\alpha| + |\beta|) = 2^{k+1} C_k^{k/2}$ , общее число  $t$  множеств  $I_j$  в разбиении не превосходит  $2^{k+1} C_k^{k/2} / (k + 2)$ .

Пусть  $|X| = 2k$ ,  $|Y| = m$ . Отталкиваясь от монотонной ДНФ, произвольную функцию  $f \in \mathcal{M}_n$  представим в виде

$$f(X, Y) = \bigvee_{\sigma \in \mathbb{B}^m} f_{\sigma}(X) \cdot Y^{\sigma} = \bigvee_{\sigma \in \mathbb{B}^m} Y^{\sigma} \cdot \bigvee_{j=1}^t f_{\sigma, j}(X), \quad Y^{\sigma} = \prod_{\sigma_i=1} y_i,$$

где монотонная функция  $f_{\sigma, j}$  совпадает с  $f_{\sigma}$  на множестве  $I_j$  и равна 0 везде, где это не противоречит монотонности. Следовательно,

$$f(X, Y) = \bigvee_{j=1}^t \bigvee_s h_{j, s}(X) \cdot \bigvee_{\sigma: f_{\sigma, j}=h_{j, s}} Y^{\sigma}, \tag{3.8}$$

где  $h_{j, s}$  пробегает множество различных функций  $f_{\sigma, j}$  (напомним, их не более  $2^{2k+2}$  при любом  $j$ ).

Любую функцию  $h_{j, s}$  можно реализовать при помощи монотонной ДНФ как дизъюнкцию не более чем  $k + 1$  мономов переменных  $X$ . Окончательно, схема, построенная по формуле (3.8), содержит  $2^{2k} + 2^m$  конъюнкторов для вычисления мономов переменных  $X$  и  $Y$  (лемма 3.1), не более  $k2^{2k+2}t < 2^{4k+3}$  дизъюнкторов для вычисления функций  $h_{j, s}$ , не более  $2^m t$  элементов для вычисления внутренних дизъюнкций в (3.8), еще по  $2^{2k+2}t$  внутренних конъюнкторов и внешних дизъюнкторов для завершения вычислений. При выборе  $k = n/4 - \log_2 n$  получаем

$$C_{\mathcal{B}_M}(\mathcal{M}^n) \leq 2^m t + (k + 2)2^{2k+2}t + 2^{2k} + 2^m \lesssim \frac{2^{m+k+1} C_k^{k/2}}{k} \sim 16 \cdot \frac{C_n^{n/2}}{n}. \tag{3.9}$$

■

В данном случае монотонная функция определяется независимо для каждого из множеств  $I_j \times \mathbb{B}^m$ , основная сложность приходится на процедуру декодирования. Теорема 3.2 тоже может служить иллюстрацией принципа локального кодирования, но применяемое в ней кодирование функции таблицей значений тривиально.

- Оценка сложности (3.9) весьма грубая. Например, поскольку почти все цепи  $\alpha$  в разбиении куба  $\mathbb{B}^k$  имеют длину  $O(\sqrt{k})$ , для почти всех множеств  $I_j$  можно обеспечить выполнение условия на сумму длин составляющих их цепей  $\sum(|\alpha| + |\beta|) = 2k - O(\sqrt{k})$ . Тогда оценка числа  $t$  снизится примерно вдвое, до  $t \lesssim 2^k C_k^{k/2}/k$ . Еще немного уменьшить множитель в оценке (3.9) можно, применяя лемму 3.3 для вычисления внутренних дизъюнкций в (3.8). Однако для получения правильной асимптотики сложности [3] требуется гораздо более изощренный метод.

Разнообразные приложения принципа локального кодирования приведены О. Б. Лупановым в [46].

### Сложность схем для мультиплексорной функции $\boxed{c}[\boxed{\_}/_2]$

Идея выделения общей части полезна не только в универсальных методах синтеза, но и при вычислении конкретных функций. Простой иллюстрацией служат оптимальные схемы для мультиплексорной функции. *Мультиплексорная функция* порядка  $n$  определяется как  $\mu_n(X; Y) = y_X$ , где  $X$  — булев вектор  $n$  адресных переменных, интерпретируемый также как число  $X \in \llbracket 2^n \rrbracket$ , а  $Y$  — вектор  $2^n$  информационных переменных  $y_i$ .

Простейший способ построить схему мультиплексора дает метод каскадов: раскладывая по первой переменной, получаем

$$\mu_n(X; Y) = \overline{x_1} \cdot \mu_{n-1}(X'; Y_0) \vee x_1 \cdot \mu_{n-1}(X'; Y_1),$$

где  $X = (x_1, X')$ ,  $Y = (Y_0, Y_1)$ . Продолжая рекурсивно, приходим к оценке<sup>2</sup>  $C_{B_0}(\mu_n) \leqslant 3 \cdot (2^n - 1)$ . Эта оценка оказывается неоптимальной. Причина в том, что независимо вычисляемые на разных «ветвях» каскадной схемы множители — элементарные конъюнкции переменных  $x_1, x_2, \dots$  — пересекаются, и вычисления частично дублируются. В оптимальном методе синтеза, предложенном П. Клейном и М. Патерсоном [241], этот недостаток устраняется.

**Теорема 3.5** ([241]).  $C_{B_0}(\mu_n) \lesssim 2^{n+1}$ .

- Нам понадобится простая лемма о сложности системы  $\{X^\sigma \mid \sigma \in \mathbb{B}^n\}$  всех элементарных конъюнкций  $n$  переменных,  $X^\sigma = \prod x_i^{\sigma_i}$ .

**Лемма 3.5.**  $C_{B_0}(\{X^\sigma \mid \sigma \in \mathbb{B}^n\}) \sim 2^n$ .

▷ Верхняя оценка получается тривиально делением множества переменных пополам. □

---

<sup>2</sup>Или, в общем виде,  $C_B(\mu_n) \leqslant C_B(\mu_1) \cdot (2^n - 1)$  с учетом  $\mu_n = \mu_1(x_1; \mu_{n-1}, \mu_{n-1})$ .

Пусть  $X = (X_0, X_1)$ ,  $|X_0| = q$ ,  $|X_1| = n - q$ . Разложим функцию  $\mu_n$  по переменным  $X_0$ :

$$\mu_n(X; Y) = \bigvee_{\tau \in \mathbb{B}^q} X_0^\tau \cdot \mu_{n-q}(X_1; Y_\tau), \quad (3.10)$$

где  $Y_\tau$  — независимые группы из  $2^{n-q}$  переменных  $Y$ . Каждую из функций  $\mu_{n-q}$  можно вычислить как

$$\mu_{n-q}(X_1; Y_\tau) = \bigvee_{\sigma \in \mathbb{B}^{n-q}} y_{\tau, \sigma} \cdot X_1^\sigma, \quad (3.11)$$

где мы подразумеваем введение нумерации с двумя индексами на множестве переменных  $y_i$ .

Все элементарные конъюнкции групп переменных  $X_0$  и  $X_1$  вычисляются со сложностью порядка  $2^q + 2^{n-q}$  согласно лемме 3.5. Еще  $2^{n+1} + 2^q$  операций достаточно для завершения вычислений по формулам (3.10), (3.11). Остается выбрать  $q \approx n/2$ . ■

На самом деле, задача синтеза мультиплексоров близка к задаче синтеза произвольных функций, поскольку мультиплексорная функция содержит в качестве подфункций все возможные булевые функции  $n$  переменных (и даже  $n + 1$  переменных, если помимо подстановки констант разрешается отождествлять переменные вместе с их отрицаниями).

- Как видим, метод теоремы 3.5 дает оценку  $C_{\mathcal{B}_0}(\mu_n) \leq 2^{n+1} + O(2^{n/2})$ . П. В. Румянцев в [72] анонсировал неулучшаемость этой оценки, что значит  $C_{\mathcal{B}_0}(\mu_n) = 2^{n+1} + \Theta(2^{n/2})$ . В более широком базисе  $C_{\mathcal{B}_2}(\mu_n) \sim 2^{n+1}$  следует из нижней оценки В. Пауля [283]. Аналогичная оценка справедлива и для формульной сложности функции. Разница лишь в остаточном члене. На самом деле, результат теоремы 3.5 прямо следует уже из метода О. Б. Лупанова синтеза контактных схем [43] в форме  $\Phi_{\mathcal{B}_0}(\mu_n) \leq 2^{n+1} + O(2^n/n)$ . С. А. Ложкин и Н. В. Власов [39] доказали  $\Phi_{\mathcal{B}_0}(\mu_n) = 2^{n+1} + (1 \pm o(1))2^n/n$ . Для более узких базисов,  $\mathcal{B} = \{\vee, \neg\}$  или  $\mathcal{B} = \{\wedge, \neg\}$ , как показал В. В. Коровин [27], метод теоремы 3.5 также дает наилучший результат,  $C_{\mathcal{B}}(\mu_n) \sim 3 \cdot 2^n$  (нужно просто выразить недостающую операцию конъюнкции или дизъюнкции через функции базиса).

В отношении сложности системы всех элементарных конъюнкций (лемма 3.5) легко показать, что  $C_{\mathcal{B}}(\{X^\sigma \mid \sigma \in \mathbb{B}^n\}) = 2^n + \Theta(2^{n/2})$  для  $\mathcal{B} \in \{\mathcal{B}_0, \mathcal{B}_2\}$ .

## Глава 4

# Метод потенциалов

$U$

Метод потенциалов — это не столько самостоятельный метод синтеза, сколько метод контроля за ключевыми параметрами схемы, позволяющий выбрать из некоторого семейства схем наилучшую. Идея в том, чтобы приписать выражениям, возникающим в процессе вычислений, просто определяемую числовую характеристику (потенциал), которая оказывается подходящим образом связана со сложностью.

### Глубина схем для сложения по модулю 3 $U[/_2]$

Рассмотрим задачу вычисления суммы  $n$  булевых переменных по модулю 3 в базисе  $\mathcal{B}_0$ . Булеву функцию, проверяющую равенство суммы  $n$  булевых переменных числу  $r$  по модулю  $m$ , будем обозначать через

$$\text{MOD}_n^{m,r}(X) = (x_1 + \dots + x_n \equiv r \pmod{m}).$$

Оператор сложения  $n$  переменных по модулю  $m$  определяется как  $\text{MOD}_n^m = (\text{MOD}_n^{m,0}, \dots, \text{MOD}_n^{m,m-1})$ .

Пусть  $X = (X^1, X^2)$ ,  $|X| = n$ ,  $|X^i| = n_i$ . Рекурсивное применение простых формул [47]

$$\text{MOD}_{n_1+n_2}^{m,r}(X) = \bigvee_{k=0}^{m-1} \text{MOD}_{n_1}^{m,k}(X^1) \cdot \text{MOD}_{n_2}^{m,r-k}(X^2) \quad (4.1)$$

методом деления пополам приводит к оценке  $D_{\mathcal{B}_0}(\text{MOD}_n^m) \leq (\lceil \log_2 m \rceil + 1) \log_2 n$ . В частности, при  $m = 3$  получаем  $D_{\mathcal{B}_0}(\text{MOD}_n^3) \leq 3 \log_2 n$ . Однако Э. Чин [165] нашел более экономный способ вычисления, использующий разбиение множества переменных на неравные части (более простое доказательство предложено автором в [89]).

**Теорема 4.1** ([165]).  $D_{\mathcal{B}_0}(\text{MOD}_n^3) \lesssim 2.89 \log_2 n$ .

► Напомним, что функция  $f$  и ее отрицание  $\bar{f}$  реализуются в базисе  $\mathcal{B}_0$  с одинаковой глубиной. Теперь заметим, что формула (4.1) имеет альтернативу:

$$\text{MOD}_{n_1+n_2}^{m,r}(X) = \bigwedge_{k=0}^{m-1} \left( \text{MOD}_{n_1}^{m,k}(X^1) \vee \overline{\text{MOD}_{n_2}^{m,r-k}(X^2)} \right). \quad (4.2)$$

Обозначим через  $N_k$  максимальное  $n$ , такое, что функции  $\text{MOD}_n^{3,r}$  представимы как конъюнкциями, так и дизъюнкциями формул глубины  $k$  и  $k - 1$ . Тогда из (4.1), (4.2) следует  $N_{k+2} \geq N_k + N_{k-2}$ , откуда сразу вытекает

$$D_{\mathcal{B}_0}(\text{MOD}_n^3) \leq 2 \log_\varphi n + O(1) < 2.89 \log_2 n + O(1),$$

где  $\varphi = \frac{1+\sqrt{5}}{2}$ . ■

В доказательстве, хоть и не вполне явно, используется потенциальная функция  $d \rightarrow \varphi^{d/2}$ , указывающая примерное число слагаемых в сумме, которую можно вычислить с глубиной  $d$ .

- В работе [165] теорема 4.1 доказана более сложным способом в терминах коммуникационной сложности. Для глубины оператора сложения по модулю 5 метод дает оценку  $D_{\mathcal{B}_0}(\text{MOD}_n^5) \lesssim 3.48 \log_2 n$  [165]. Использование специальных формул с разбиением множества переменных на три группы позволило автору [89] усилить оценку теоремы 4.1 до  $D_{\mathcal{B}_0}(\text{MOD}_n^3) \lesssim 2.8 \log_2 n$ . В работе [89] также предложен общий метод, приводящий к понижению глубины сложения по малым простым модулям, см. далее на стр. 89.

### Формульная сложность линейной функции в тернарном базисе $\boxed{U}$

Рассмотрим задачу минимизации размера формулы для линейной функции  $\Lambda_n = x_1 \oplus x_2 \oplus \dots \oplus x_n$  в базисе  $\mathcal{B}_3 = \{\text{maj}_3(x, y, z), \bar{x}, 1\}$ <sup>1</sup>.

Поскольку любая функция из  $\mathcal{U}_2$  бесповторно выражима в  $\mathcal{B}_3$ , тривиально выполняется  $\Phi_{\mathcal{B}_3}(\Lambda_n) \leq \Phi_{\mathcal{U}_2}(\Lambda_n) \preccurlyeq n^2$ . Иллюстрацией более сильных выражительных возможностей базиса  $\mathcal{B}_3$  служит формула [168]

$$x \oplus y \oplus z = \text{maj}_3(x, \text{maj}_3(\bar{x}, y, z), \text{maj}_3(\bar{x}, \bar{y}, \bar{z})). \quad (4.3)$$

Вместо переменных в нее можно подставить некоторые линейные функции, что приведет при  $n = p + q + r$  к соотношению

$$\Phi_{\mathcal{B}_3}(\Lambda_n) \leq 3\Phi_{\mathcal{B}_3}(\Lambda_p) + 2\Phi_{\mathcal{B}_3}(\Lambda_q) + 2\Phi_{\mathcal{B}_3}(\Lambda_r). \quad (4.4)$$



Ури Цвик

Тель-Авивский университет,  
с конца 1980-х

<sup>1</sup> В общем виде, рассматривается базис  $\mathcal{U}_k$ , который служит обобщением базиса  $\mathcal{U}_2$  на множество  $k$ -местных булевых функций. Базис  $\mathcal{U}_k$  включает функции, по каждой переменной монотонно невозрастающие или монотонно неубывающие, т.е. ровно те функции, из которых нельзя получить линейную функцию двух переменных путем подстановки констант, инверсий и отождествлений переменных, подробнее см. в [112, 168]. Несложно проверить, что базис  $\mathcal{B}_3$  эквивалентен базису  $\mathcal{U}_3$ , что значит  $\Phi_{\mathcal{B}_3}(f) = \Phi_{\mathcal{U}_3}(f)$  для любой булевой функции  $f$ .

При выборе  $p = q = r$  получаем  $\Phi_{\mathcal{B}_3}(\Lambda_{3n}) \leq 7\Phi_{\mathcal{B}_3}(\Lambda_n)$ , откуда  $\Phi_{\mathcal{B}_3}(\Lambda_n) \preccurlyeq n^{\log_3 7} < n^{1.78}$ . Но можно сделать еще лучше, воспользовавшись неравномерной зависимостью формулы (4.3) от переменных.

**Теорема 4.2** ([168]).  $\Phi_{\mathcal{B}_3}(\Lambda_n) \prec n^{1.74}$ .

► Будем искать оценку сложности в виде

$$\Phi_{\mathcal{B}_3}(\Lambda_n) \leq c n^\mu, \quad (4.5)$$

применяя (4.4) рекурсивно с выбором  $q = r = \gamma n$  и  $p = (1 - 2\gamma)n$  (здесь  $\gamma, \mu$  — неизвестные параметры). О целочисленности  $p, q, r$  пока беспокоиться не будем. Для вывода (4.5) требуется база индукции (малые значения  $n$ ), которая обеспечивается выбором константы  $c$  и индуктивный переход, который обеспечивается при подстановке оценок (4.5) в (4.4) неравенством

$$3(1 - 2\gamma)^\mu + 4\gamma^\mu \leq 1. \quad (4.6)$$

С целью минимизации показателя  $\mu$  выбираем  $\gamma \approx 0.4$  и окончательно находим  $\Phi_{\mathcal{B}_3}(\Lambda_n) \prec n^{1.74}$ . ■

- Учет условия  $p, q, r \in \mathbb{Z}$  не меняет ничего по сути, лишь приводит к более громоздкой выкладке. Например, можно искать оценку для  $n \geq n_0$  в виде

$$\Phi_{\mathcal{B}_3}(\Lambda_n) \leq c_1 n^\mu - c_2 n. \quad (4.7)$$

► Пусть  $q = r = \lfloor \gamma n \rfloor$ . Тогда  $|q - \gamma n| \leq 1/2$  и  $|p - (1 - 2\gamma)n| \leq 1$ . Заметим, что при любых  $a > 0$ ,  $\mu \geq 1$  и  $x \in [0, 1]$  справедливо неравенство  $(a + x)^\mu \leq a^\mu + \mu(a + 1)x$ . Пусть  $x = 1/n$ . Теперь индуктивный переход обеспечивается неравенством

$$\begin{aligned} n^{-\mu}(3\Phi_{\mathcal{B}_3}(\Lambda_p) + 4\Phi_{\mathcal{B}_3}(\Lambda_q)) &\leq \\ &3c_1(1 - 2\gamma + x)^\mu + 4c_1(\gamma + x/2)^\mu - c_2(3p + 4q)n^{-\mu} \leq \\ &c_1(3(1 - 2\gamma)^\mu + 4\gamma^\mu + 3(2 - 2\gamma)\mu x + 2(1 + \gamma)\mu x) - c_2((3 - 2\gamma)n^{1-\mu} - n^{-\mu}) \leq \\ &c_1 - c_2(3 - 2\gamma - x)n^{1-\mu} + c_1(8 - 4\gamma)\mu x \leq c_1 - c_2 n^{1-\mu}, \end{aligned}$$

если положить  $c_2 = \beta c_1$ , где  $\beta = (8 - 4\gamma)\mu/(1 - 2\gamma)$ . Порог  $n_0$  выбирается так, чтобы выполнялось  $n_0^{\mu-1} \geq \beta + 1$  (тогда правая часть (4.7) не меньше  $c_1 n$ ). Окончательно, выбор параметра  $c_1$  обеспечивает основание индукции, а именно, выполнение (4.7) для всех  $n \in [n_0, n_1]$ , где  $n_1 = (n_0 + 1)/(1 - 2\gamma)$  (например, подойдет  $c_1 = 2n_1$ , поскольку заведомо  $\Phi_{\mathcal{B}_3}(\Lambda_n) \leq \Phi_{\mathcal{U}_2}(\Lambda_n) < 2n^2$ ). □

Приведенная оценка пока не улучшена. В [279] показано, что лучшей оценки, используя только (4.3), получить нельзя. Об этом речь пойдет далее (см. стр. 49). В [94] автором доказана нижняя оценка  $\Phi_{\mathcal{B}_3}(\Lambda_n) \succ n^{1.53}$ .

## Глубина схем для многократного сложения $\boxed{U}$

В школьном методе умножение  $n$ -разрядных чисел сводится к сложению  $n$  чисел (вообще говоря,  $2n$ -разрядных). Естественная мысль выполнить сложение деревом из обычных сумматоров, пусть даже параллельных, приводит к схеме лишь

глубины  $\Omega(\log^2 n)$ . Конструкция схем логарифмической глубины опирается на идею компрессоров, предложенную на рубеже 1950–60-х гг. независимо многими исследователями. Суть идеи в том, чтобы промежуточные суммы вычислять в определенном смысле «не до конца».

Рассмотрим задачу в общей постановке как вычисление суммы  $n$  элементов в некоторой коммутативной группе  $(G, +, 0)$ .  $(k, l)$ -компрессором называется схема, выполняющая преобразование  $(x_1, \dots, x_k) \rightarrow (y_1, \dots, y_l)$  с условием сохранения суммы:  $\sum_{i=1}^k x_i = \sum_{j=1}^l y_j$ , где  $k > l$ .

Дерево из  $(k, l)$ -компрессоров позволяет свести суммирование  $n$  элементов к суммированию  $l$  штук (предполагаем, что единица группы  $0 \in G$  доступна в качестве дополнительного входа схемы). Произвольный  $(k, l)$ -компрессор  $A$  можно охарактеризовать отображением  $\mathbf{a} \rightarrow \mathbf{b}$  вектора глубин<sup>2)</sup> входов  $\mathbf{a} = (a_1, \dots, a_k)$  в вектор глубин выходов  $\mathbf{b} = (b_1, \dots, b_l)$ . Исходя из физического смысла задачи, полагаем, что  $\max_i a_i < \max_j b_j$  при любом  $\mathbf{a}$  (т.е. никакой из входов компрессора не является одновременно и выходом).

Например, семейство параллельных копий сумматора  $FA_3$  образует числовой  $(3, 2)$ -компрессор. Если глубины входов компрессора равны 0, 0, 1, то выходы вычисляются на глубинах 2 и 3, см. рис. 1а). Пример дерева, построенного из таких компрессоров, показан на рис. 4.1 (компрессоры изображаются многоугольными блоками с тремя входами и двумя выходами; промежуточные суммы выстроены по шкале глубины).

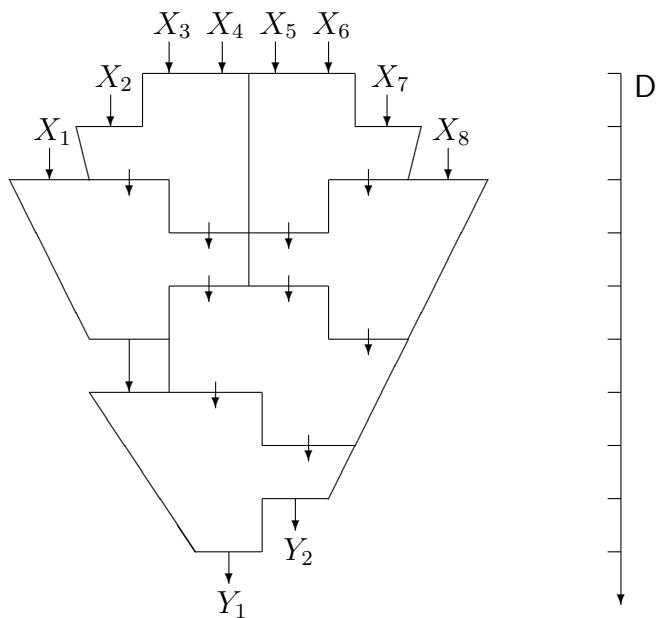


Рис. 4.1: Схема многократного сложения из  $(3, 2)$ -компрессоров [279]

Вопрос о том, как располагать компрессоры в схеме многократного сложения, чтобы минимизировать ее глубину, оказался нетривиальным. Только в начале

<sup>2)</sup>Термин «глубина» здесь достаточно условен. Числа  $a_i$  мы предполагаем целыми только для удобства рассуждений.

1990-х гг. М. Патерсон, Н. Пиппенджер и У. Цвик [279] нашли общий подход к решению этой задачи.



Майкл Стюарт  
Патерсон

Уорикский университет, с 1971

Обозначим минимально возможную глубину  $(n, l)$ -компрессора, составленного из  $(k, l)$ -компрессоров  $A$  через  $D_A(n)$ . Метод потенциалов позволяет оценить эту глубину снизу. Через  $g_A(\mathbf{a}; x)$  обозначим характеристический многочлен компрессора  $A$ :

$$g_A(\mathbf{a}; x) = \sum_{j=1}^l x^{b_j} - \sum_{i=1}^k x^{a_i}. \quad (4.8)$$

**Лемма 4.1** ([279]). *Пусть  $\lambda(\mathbf{a})$  — максимальный положительный корень многочлена  $g_A(\mathbf{a}; x)$ . Обозначим  $\lambda = \sup_{\mathbf{a} \in \mathbb{Z}^k} \lambda(\mathbf{a})$ . Тогда*

$$D_A(n) \geq \log_\lambda(n/l).$$

▷ Прежде всего заметим, что многочлен  $g_A(\mathbf{a}; x)$  имеет максимальный корень  $\lambda(\mathbf{a}) > 1$ , поскольку  $g_A(\mathbf{a}; 1) = l - k < 0$ , и  $g_A(\mathbf{a}; x) \rightarrow +\infty$  при  $x \rightarrow +\infty$ . Тогда и верхняя грань  $\lambda > 1$  существует в силу  $\lambda(\mathbf{a}) \leq k$ , поскольку при  $x \geq k$ :

$$g_A(\mathbf{a}; x) \geq x^{\max_j b_j} - k \cdot x^{\max_i a_i} \geq (x - k) \cdot x^{\max_i a_i} \geq 0.$$

Рассмотрим поэтапную процедуру построения схемы из компрессоров, от входов к выходам, и проследим за изменением множества глубин слагаемых промежуточных сумм. В начальный момент имеем  $n$  входов глубины 0. При добавлении очередного компрессора  $k$  чисел из списка (глубины входов компрессора) заменяются  $l$  другими числами (глубины выходов компрессора).

Промежуточному слагаемому, расположенному на глубине  $d$ , припишем числовую величину (потенциал)  $\lambda^d$ . В силу определения  $\lambda$  сумма потенциалов слагаемых не убывает при добавлении компрессоров. Поэтому  $n \leq l \cdot \lambda^{D_A(n)}$ . □

Если компрессор — не просто абстрактное устройство, а булева схема с глубиной, определяемой обычным образом, то верхняя грань  $\lambda$  в условии леммы 4.1 достигается при некотором  $\mathbf{a}$ , см. [279].

Доказательство нижней оценки служит руководством к получению верхней.

**Теорема 4.3** ([279]). *Пусть  $\lambda$  — максимальный положительный корень многочлена  $g_A(\mathbf{a}; x)$ . Тогда*

$$D_A(n) \lesssim \log_\lambda n.$$

► Не ограничивая общности, можно считать, что  $\min_i a_i = a_1 = 0$ . Положим  $b = \max_j b_j$ . Уровнем компрессора (в схеме из компрессоров) будем называть глубину его первого входа.

Составим схему из компрессоров  $A$  так, что при любом  $d$  она содержит

$$C_d = \begin{cases} \lceil n\lambda^{-d} + \frac{l}{k-l} \rceil, & -b < d \leq \log_\lambda n \\ 0, & \text{иначе} \end{cases}$$

компрессоров уровня  $d$ . Входами компрессоров могут быть входы схемы и выходы других компрессоров, вычисляемые на соответствующей глубине. Входы компрессоров, не поступающие с выходов других компрессоров, полагаются входами схемы. Выходы компрессоров, которые не присоединяются ко входам других компрессоров, полагаются выходами схемы.

Компрессор уровня  $d$  принимает входы на глубинах  $a_i + d$  и производит выходы на глубинах  $b_j + d$ . По построению, суммарное число входов глубины  $d$ , необходимых для компрессоров схемы, равно  $\sum_{i=1}^k C_{d-a_i}$ , а число выходов, производимых компрессорами на той же глубине, равно  $\sum_{j=1}^l C_{d-b_j}$ .

Установим несколько свойств схемы.

а) На глубине  $d$ ,  $0 \leq d \leq \log_\lambda n$ , схема не имеет выходов. Действительно, разница между числом входов и числом выходов компрессоров на глубине  $d$  положительна:

$$\begin{aligned} \sum_{i=1}^k C_{d-a_i} - \sum_{j=1}^l C_{d-b_j} > \\ \sum_{i=1}^k \left( n\lambda^{a_i-d} + \frac{l}{k-l} \right) - \sum_{j=1}^l \left( n\lambda^{b_j-d} + \frac{l}{k-l} + 1 \right) = -n\lambda^{-d} g_A(\mathbf{a}; \lambda) = 0. \end{aligned} \quad (4.9)$$

б) Схема имеет более  $n$  входов на глубине 0. Действительно, при  $d = 0$  ввиду  $C_{-b} = 0$ , в выкладке (4.9) разница между левой частью и правой не меньше  $n\lambda^b > n$ .

в) Общее число выходов схемы на глубине свыше  $\log_\lambda n$  равно  $O(1)$  (схема может также иметь выходы на отрицательной глубине). Действительно, это число не превосходит суммарного числа выходов у компрессоров уровней  $d > \log_\lambda n - b$ , т.е.

$$l \sum_{d>\log_\lambda n-b} C_d < lb \left( n\lambda^{b-\log_\lambda n} + \frac{l}{k-l} + 1 \right) = lb (\lambda^b + k).$$

Таким образом, если  $n$  переменных  $x_i$  присоединить ко входам схемы на глубине 0, а на остальные входы подать константу 0  $\in G$ , то построенная схема будет  $(n, m)$ -компрессором,  $m = O(1)$  (выходы на отрицательной глубине не в счет — они реализуют 0  $\in G$ ). Добавив в схему еще несколько компрессоров  $A$ , число выходов сократим до  $l$ . Общая глубина схемы равна  $\log_\lambda n + O(1)$ . ■

Характеристический многочлен рассмотренного выше числового  $(3, 2)$ -компрессора  $FA_3$  равен

$$g_{FA_3}(0, 0, 1; x) = x^3 + x^2 - x - 2. \quad (4.10)$$

Его единственный положительный корень —  $\lambda \approx 1.2056$ . Поэтому для глубины оператора  $\Sigma_{m,n}$  сложения  $n$  штук  $m$ -разрядных чисел с учетом следствия 2.2 получаем

**Следствие 4.1.**  $D_{B_2}(\Sigma_{m,n}) \leq 3.71 \log_2 n + (1 + o(1)) \log_2(m + \log_2 n)$ .

**Следствие 4.2.**  $D_{B_2}(M_n) \lesssim 4.71 \log_2 n$ .

- Используя несколько более сложную конструкцию  $(6, 3)$ -компрессора, Патерсон и Цвик [281] получили оценку  $D_{B_2}(\Sigma_{1,n}) \lesssim 3.57 \log_2 n$ , а Э. Гроув [209] при помощи специального  $(7, 3)$ -компрессора — оценку  $D_{B_0}(\Sigma_{1,n}) \lesssim 4.94 \log_2 n$ .

Авторы [279] также распространяли метод на сложность формул многократного сложения. Теорема 4.2 служит простым примером применения этого метода (формула (4.3) описывает  $(3, 1)$ -компрессор в  $(\mathbb{B}, \oplus)$ ). В общем случае используемый для построения формулы компрессор  $A$  характеризуется отображением  $\mathbf{a} \rightarrow \mathbf{b}$  вектора размеров входов  $\mathbf{a} = (a_1, \dots, a_k)$  в вектор размеров выходов  $\mathbf{b} = (b_1, \dots, b_l)$ . Характеристическая функция компрессора записывается как

$$g_A(\mathbf{a}; x) = \sum_{j=1}^l b_j^x - \sum_{i=1}^k a_i^x.$$

Максимальный положительный корень функции обозначается через  $\mu(\mathbf{a})$ . Пусть  $\mu = \sup_{\mathbf{a} \in \mathbb{Z}^k} \mu(\mathbf{a})$ . Тогда минимальная сложность  $(n, l)$ -компрессора, составленного из компрессоров  $A$  имеет величину  $(n/l)^{1/\mu+o(1)}$ . Метод синтеза аналогичен методу теоремы 4.3, только для классификации слагаемых используется не глубина, а округленный логарифм размера формулы  $\lceil \log_2 s \rceil$  (и округления приводят к чуть более громоздким выкладкам в оценках).

В работе [281] авторы при помощи специальных конструкций компрессоров получили этим методом оценки  $\Phi_{B_2}(\Sigma_{1,n}) \prec n^{3.13}$  и  $\Phi_{B_0}(\Sigma_{1,n}) \prec n^{4.57}$ . Более сильные оценки получаются с использованием перекодировки входов, см. ниже на стр. 93.

## Параллельное перестроение арифметических формул U



Ричард Пирс Брент  
Национальный университет  
Австралии, Канберра, с 1972

В представляющей прикладной интерес задаче о параллельном перестроении арифметического выражения требуется по данной формуле длины  $n$  построить эквивалентную<sup>3)</sup> формулу возможно меньшей глубины (в бинарном базисе желательно как можно ближе к  $\log_2 n$ ). Известно, что в любом полном булевом базисе и в арифметических базисах общего вида всегда находится альтернативная формула глубины  $O(\log n)$ , что следует соответственно из результатов В. М. Храпченко (анонсирован в [115]) и Р. Брента [156]. Более точный вид соотношения между глубиной и сложностью зависит от базиса. Поэтому вводится понятие<sup>4)</sup> константы равномерности  $c_{\mathcal{B}}$  базиса  $\mathcal{B}$ , определяемой как верхняя грань чисел  $c$ , при которых справедливо  $\Phi_{\mathcal{B}}(f) \leq \Phi_{\mathcal{B}}(f) \rightarrow \infty$ . В исследовании констант равномерности арифметических базисов полагаем полукольцо, в котором выполняются вычисления, коммутативным.

Для простоты ограничимся рассмотрением формул в монотонном арифметическом базисе  $\mathcal{A}_+$ . Равномерность этого базиса (т.е. существование константы  $c_{\mathcal{A}_+}$ ) доказана Р. Брентом, Д. Куком и К. Маруюмой в [158].

<sup>3)</sup>Т.е. вычисляющую ту же функцию.

<sup>4)</sup>Предложено В. М. Храпченко в [107].

**Лемма 4.2** ([158]). Пусть бесповторная формула сложности  $n$  над  $\mathcal{A}_+$  вычисляет функцию  $f(X, y)$ . Тогда  $f(X, y) = f_1(X) \cdot y + f_2(X)$ , где  $\Phi_{\mathcal{A}_+}(f_1), \Phi_{\mathcal{A}_+}(f_2) \leq n$ .

▷ Доказывается тривиально индукцией по сложности формулы.  $\square$

**Теорема 4.4** ([158]). Для любого монотонного многочлена  $f$  выполнено

$$\mathsf{D}_{\mathcal{A}_+}(f) < 2.47 \log_2 \Phi_{\mathcal{A}_+}(f) + O(1).$$

► Определим рекуррентную последовательность чисел

$$N_0 = 1, \quad N_1 = 2, \quad N_2 = 3, \quad N_k = N_{k-2} + N_{k-3}, \quad k \geq 3.$$

Докажем по индукции, что формула длины  $\leq N_k$  может быть перестроена в формулу глубины  $\leq k$ . При  $k \leq 2$  утверждение очевидно. Докажем индуктивный переход. Нам понадобится простая

**Лемма 4.3.** При любом  $t \leq n$  в произвольной бинарной формуле сложности  $n$  найдется подформула сложности  $\geq t$ , главные подформулы<sup>5)</sup> которой имеют сложность  $< t$ .

▷ Достаточно организовать просмотр дерева формулы в направлении от корня к листьям, двигаясь по подформулам сложности  $\geq t$  до тех пор, пока возможно. Последняя подформула в цепочке будет искомой.  $\square$

Без ограничения общности можем считать формулы бесповторными (случай повторяющихся входов сводится к рассматриваемому). Рассмотрим формулу  $F$  сложности  $N_k$  и, используя лемму 4.3, выделим в ней подформулу  $G$  сложности  $\geq N_{k-3} + 1$ , которая имеет вид  $G_1 \circ G_2$ , где  $\Phi(G_1), \Phi(G_2) \leq N_{k-3}$  и  $\circ \in \{+, \cdot\}$ , см. рис. 4.2.

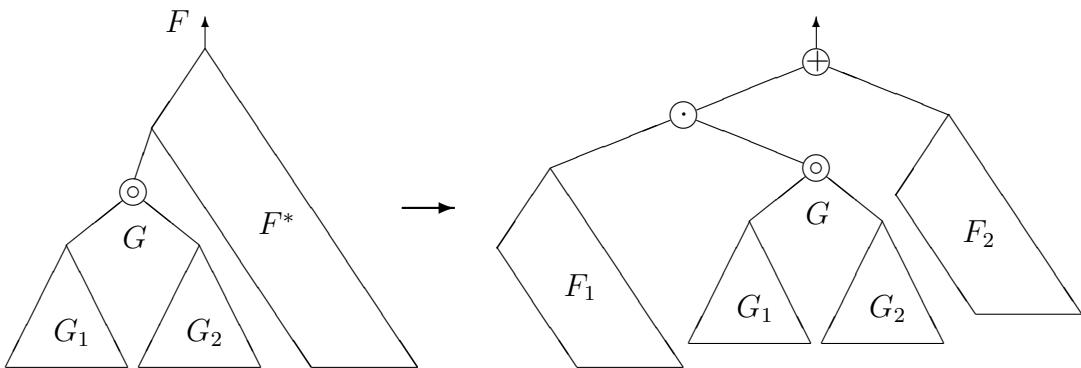


Рис. 4.2: Перестроение арифметической формулы

<sup>5)</sup>Напомним, что главными подформулами формулы  $F_1 \circ F_2$  считаются образующие ее формулы  $F_1, F_2$ .

Пусть формулы  $F, G_1, G_2$  реализуют функции  $f, g_1, g_2$  соответственно. Обозначим через  $f^*(X, y)$  функцию, в которую переходит  $f(X)$  при подстановке в формулу  $F$  вместо подформулы  $G$  новой переменной  $y$ . По построению,  $\Phi(f^*) \leq \Phi(F) - \Phi(G) + 1 \leq N_{k-2}$ . Согласно лемме 4.2, имеет место  $f^*(X, y) = f_1 \cdot y + f_2$ , где  $\Phi(f_1), \Phi(f_2) \leq N_{k-2}$ . Окончательно, функция  $f$  может быть вычислена как

$$f = f_1 \cdot (g_1 \circ g_2) + f_2$$

с глубиной  $\leq k$ , поскольку по индуктивному предположению  $D(g_i) \leq k - 3$  и  $D(f_i) \leq k - 2$ . Остается заметить, что  $N_k \asymp a^k$ , где  $a \approx 1.325$  — корень уравнения  $x^3 = x + 1$ . ■

- Ни добавление в базис вычитания, ни расширение его до полного не изменяют вывода теоремы 4.4 (например, потому что вычитания и умножения на константы всегда можно опустить на уровень входов переменных). Таким образом,  $c_A \leq c_{A+} < 2.47$ . Более изощренный способ декомпозиции формул с анализом нескольких случаев позволил С. Косараю [246] усилить оценку до  $c_{A+} \leq 2$ . При этом согласно результату Д. Копперсмита и Б. Шибера [176],  $c_{A+} \geq 1.5$ .



Дэвид Юджин  
Маллер

Иллинский университет,  
с 1953 по 1992

Метод перестроения арифметических выражений работает и для булевых монотонных формул. На самом деле,  $c_{B_0} \leq c_{B_M} \leq c_{A+}$  (базис  $B_M$  — частный случай арифметического базиса; неравенство для базиса  $B_0$  справедливо в силу правил де Моргана, позволяющих опустить отрицания на уровень входов). Однако взаимная дистрибутивность операций булева базиса существенно расширяет возможности преобразования булевых формул, в чем мы убедимся на примере результата Ф. Препараты и Д. Маллера [287].

**Теорема 4.5** ([287]). Для любой булевой функции  $f$  выполнено

$$D_{B_0}(f) < 1.82 \log_2 \Phi_{B_0}(f) + O(1). \quad (4.11)$$

- Ввиду возможности опускания отрицаний в формулах на уровень входов (правила де Моргана), достаточно доказать соотношение (4.11) в монотонном базисе  $B_M$ . Ключевым является следующее наблюдение, усиливающее лемму 4.2 для булева случая.

**Лемма 4.4** ([287]). Пусть бесповторная формула  $G$  сложности  $n$  над  $B_m$  вычисляет функцию  $f(X, y)$ . Тогда функцию  $f$  можно представить либо как  $f(X, y) = f_1(X) \cdot y \vee f_2(X)$ , либо как  $f(X, y) = (f_1(X) \vee y) \cdot f_2(X)$ , где  $\Phi_{B_M}(f_1) < n/2$  и  $\Phi_{B_M}(f_2) < n$ .

▷ В дереве, изображающем формулу  $G$ , выделим путь от переменной  $y$  к корню, см. рис. 4.3. Можно записать

$$G(X, y) = (\dots ((y \circ_1 G_1) \circ_2 G_2) \dots) \circ_k G_k,$$

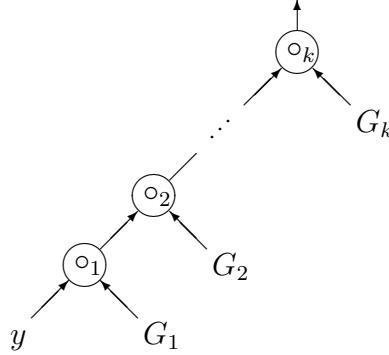


Рис. 4.3: Декомпозиция булевой формулы

где  $G_i$  — формулы, присоединяемые по пути.

Тогда функция  $f$  реализуется любой из двух формул

$$G^\wedge \cdot y \vee G(X, 0), \quad (G^\vee \vee y) \cdot G(X, 1), \quad \text{где } G^\wedge = \bigwedge_{\circ_i=\wedge} G_i, \quad G^\vee = \bigvee_{\circ_i=\vee} G_i.$$

Поскольку  $\Phi(G^\wedge) + \Phi(G^\vee) = n - 1$ , одна из формул удовлетворяет условиям леммы.  $\square$

Далее следуем доказательству теоремы 4.4. Теперь последовательность чисел определяется как

$$N_0 = 1, \quad N_1 = 2, \quad N_2 = 3, \quad N_k = N_{k-1} + N_{k-3}, \quad k \geq 3.$$

Докажем, что формула длины  $\leq N_k$  может быть перестроена в формулу глубины  $\leq k$  (осталось доказать индуктивный переход).

Пусть бесповторная формула  $F$  сложности  $N_k$  вычисляет функцию  $f(X)$ . При помощи леммы 4.3 выделим в ней подформулу  $G$  сложности  $\geq N_{k-3} + 1$ , которая имеет вид  $G_1 \circ G_2$ , где  $\Phi(G_1), \Phi(G_2) \leq N_{k-3}$  и  $\circ \in \{\vee, \wedge\}$ . Обозначим через  $f^*(X, y)$  функцию, в которую переходит  $f(X)$  при подстановке в формулу  $F$  вместо подформулы  $G$  новой переменной  $y$ . По построению,  $\Phi(f^*) \leq \Phi(F) - \Phi(G) + 1 \leq N_{k-1}$ . Выражая  $f^*$  при помощи леммы 4.4, окончательно получаем

$$f = (f_1 \circ_1 (g_1 \circ g_2)) \circ_2 f_2,$$

где  $\circ, \circ_1, \circ_2 \in \{\vee, \wedge\}$ ,  $\Phi(f_2) < N_{k-1}$  и  $\Phi(f_1) < N_{k-1}/2 \leq N_{k-2}$ . Таким образом, по предположению индукции  $D(g_i) \leq k-3$ ,  $D(f_1) \leq k-2$  и  $D(f_2) \leq k-1$ . Поэтому  $D(f) \leq k$ . Остается заметить, что  $N_k \asymp a^k$ , где  $a \approx 1.465$  — корень уравнения  $x^3 = x^2 + 1$ .  $\blacksquare$

- Оценка  $c_{\mathcal{B}_M} < 1.82$  теоремы [287] усилена В. М. Храпченко до  $c_{\mathcal{B}_M} < 1.73$  в [106]. Нижняя оценка  $c_{\mathcal{B}_M} > 1.06$  получена автором в [92]. Для полных базисов арифметического типа, например,  $\mathcal{A}, \mathcal{B}_0$ , нетривиальные нижние оценки констант равномерности неизвестны. Для неарифметических базисов такие оценки есть: в частности, для полного булева базиса  $\mathcal{B} = \{\setminus\}$  из одной операции «штрих Шеффера» оценка  $c_{\mathcal{B}} \geq 2$  доказана Храпченко в [103].

## Глава 5

# Метод приближений

$\boxed{\varepsilon}$

Метод приближений основан на следующем наблюдении: подзадачи, на которые разбивается исходная задача, зачастую выгодно решать не «до конца», а приблизительно, с контролируемой точностью. Из приближенных решений подзадач строится точное решение исходной задачи.

### Быстрое деление чисел $\boxed{\varepsilon} /_2$

В вычислительной практике уравнения  $f(x) = 0$  часто решаются методом Ньютона—Рафсона (методом касательных) посредством выполнения итераций  $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$ . Если начальное значение  $x_0$  выбрано удачно, и функция  $f(x)$  удовлетворяет простым ограничениям, то  $x_i \rightarrow x^*$ , где  $f(x^*) = 0$ , причем скорость сходимости метода в общем случае квадратична. Около 1966 г. С. Кук [172] заметил, что этот метод может быть адаптирован для быстрого деления чисел.



Стивен Артур Кук  
Университет Торонто, с 1970

Поскольку деление сводится к инвертированию и умножению, а методы быстрого умножения хорошо известны, достаточно ограничиться операцией инвертирования. Определим  $(n, n + 1)$ -оператор (приближенного) инвертирования  $I_n$  на множестве  $n$ -разрядных чисел из отрезка  $[1/2, 1]$  условием  $|I_n(a) - 1/a| \leq 2^{-n}$  (определение неоднозначно и на самом деле задает целое семейство операторов). Ограничивааясь отрезком  $[1/2, 1]$ , мы имеем в виду, что общий случай, когда  $a \notin [1/2, 1]$ , сводится к рассматриваемому при замене  $a$  на  $2^k a$  и последующем умножении результата на  $2^k$  (эти операции реализуются сдвигом позиции запятой, что в схемной модели выполняется бесплатно). Напомним, что  $M(n)$  обозначает сглаженную функцию сложности умножения.

**Теорема 5.1** ([172]).  $C(I_n) \leq M(n)$ .

► Доказательство следует близко к версии Й. Хостада [215]. Пусть  $a_{..k}$  обозначает число  $a$  с отсеченными разрядами после запятой младше  $k$ -го — это приближение к  $a$  с точностьюю  $2^{-k}$ . Определим последовательность  $r_i$  следующим образом<sup>1</sup>:

$$r_0 = 1, \quad \tilde{r}_{i+1} = 2r_i - a_{..4+2^i} \cdot r_i^2, \quad r_{i+1} = (\tilde{r}_{i+1})_{..4+2^{i+1}}. \quad (5.1)$$

**Лемма 5.1.**  $|1 - r_i a_{..4+2^i}| < 2^{-2^{i-1}-1/2}$ .

▷ Очевидно, неравенство справедливо при  $i = 0$ . Докажем индуктивный переход от  $i$  к  $i + 1$ .

По индуктивному предположению, справедливо

$$0 \leq 1 - \tilde{r}_{i+1} a_{..4+2^i} = (1 - r_i a_{..4+2^i})^2 \leq 2^{-2^{i-1}}.$$

Как следствие,  $0 < \tilde{r}_{i+1} \leq r_{i+1} \leq a_{..4+2^i}^{-1} \leq 2$ . Теперь оценка леммы следует из выкладки:

$$\begin{aligned} |1 - r_{i+1} a_{..4+2^{i+1}}| &\leq \\ |1 - \tilde{r}_{i+1} a_{..4+2^i}| + |r_{i+1} a_{..4+2^i} - \tilde{r}_{i+1} a_{..4+2^i}| + |r_{i+1} a_{..4+2^{i+1}} - r_{i+1} a_{..4+2^i}| &\leq \\ 2^{-2^{i-1}} + a_{..4+2^i} |r_{i+1} - \tilde{r}_{i+1}| + r_{i+1} |a_{..4+2^{i+1}} - a_{..4+2^i}| &\leq \\ 2^{-2^{i-1}} + 1 \cdot 2^{-2^{i+1}-4} + 2 \cdot 2^{-2^i-4} &< \frac{11}{16} \cdot 2^{-2^i} < 2^{-2^{i-1}-1/2}. \end{aligned}$$

□

Как следствие из леммы, получаем

$$|1 - r_i a| \leq |1 - r_i a_{..4+2^i}| + r_i |a - a_{..4+2^i}| < 2^{-2^{i-1}-1/2} + 2^{-2^i-4} < 2^{-2^{i-1}}.$$

Для вычисления  $a^{-1}$  с точностьюю  $2^{-n}$  достаточно определить  $r_i$  вплоть до  $i = \lceil \log_2 n \rceil + 1$ . Вычисление по формулам (5.1) приводит к оценке

$$\mathsf{C}(I_n) \leq \sum_{i=0}^{\lceil \log_2 n \rceil + 1} (2\mathsf{C}(M_{4+2^i}) + O(2^i)) \preccurlyeq \mathsf{M}(n).$$

■

- Аналогичным образом строятся схемы для некоторых других элементарных числовых функций, например, для квадратного корня, см., например, [157].

### Быстрое деление с остатком комплексных многочленов $\boxed{\varepsilon} \boxed{s}$

Метод последовательных приближений для многочленов работает еще лучше, чем для чисел, позволяя решать аналогичные задачи эффективнее.

---

<sup>1</sup>Метод касательных для поиска нуля функции  $f(x) = a - 1/x$ .

Изящный прием сведения операции деления многочленов с остатком к более простым операциям, делению по модулю  $x^n$  и умножению, предложил Ф. Штрассен в [317]. Пусть  $QR_{2n,n}^R$  и  $D_n^R$  означают операции с многочленами в  $R[x]$ : соответственно оператор вычисления частного и остатка от деления многочлена степени  $\leq 2n - 1$  на многочлен степени  $n$  и оператор деления многочленов по модулю  $x^n$ .

**Лемма 5.2** ([317]).  $C_A(QR_{2n,n}^R) \leq C_A(D_n^R) + C_A(M_n^R) + O(n)$ .

▷ Пусть  $q(x)$  и  $r(x)$  — соответственно частное и остаток от деления  $a(x)$  на  $b(x)$ , где  $\deg a \leq 2n - 1$ ,  $\deg b = n$  и  $\deg q, r < n$ . Тогда

$$a(x) = q(x)b(x) + r(x). \quad (5.2)$$

Через  $\tilde{a}(x)$ ,  $\tilde{q}(x)$ ,  $\tilde{b}(x)$ ,  $\tilde{r}(x)$  обозначим соответственно многочлены  $x^{2n-1}a(1/x)$ ,  $x^{n-1}q(1/x)$ ,  $x^n b(1/x)$  и  $x^{n-1}r(1/x)$ . Подставляя в (5.2)  $1/x$  вместо  $x$  и домножая на  $x^{2n-1}$ , получаем

$$\tilde{a}(x) = \tilde{q}(x)\tilde{b}(x) + x^n\tilde{r}(x). \quad (5.3)$$

Тогда

$$\tilde{q} = \tilde{a}/\tilde{b} \bmod x^n, \quad \tilde{r} = (\tilde{a} - \tilde{q}\tilde{b})/x^n. \quad (5.4)$$

Вычисления по формулам (5.4) сводятся к делению по модулю  $x^n$ , умножению и вычитанию многочленов.  $\square$



Жорис ван дер Хувен

Политехническая школа,  
Париж, с 2009

Деление можно выполнить при помощи инвертирования по модулю  $x^n$  и умножения. Инвертирование выполняется полиномиальным аналогом описанного в предыдущем пункте числового метода. В действительности, практически и теоретически более выгодно использовать метод приближений непосредственно для операции деления. В наиболее интересном случае поля  $\mathbb{C}$  (а также  $\mathbb{R}$ ), учитывая что умножение комплексных многочленов выполняется при помощи ДПФ, сложность метода удобно выражать через сложность ДПФ. Опишем самый быстрый известный метод, принадлежащий Ж. ван дер Хувену [221].

**Теорема 5.2** ([221]). Пусть константа  $c_F$  удовлетворяет условию  $C_A(\text{ДПФ}_N) \leq c_F N \log_2 N$  при любом  $N = 2^k$ . Тогда

$$C_A(QR_{2n,n}) \lesssim 12c_F \cdot n \log_2 n.$$

Заметим, что сложность умножения тривиально оценивается как  $C_A(M_n) \leq 6c_F \cdot n \log_2 n$  (на близость  $n$  к степени двойки можно не обращать внимания). Таким образом, сложность деления с остатком оказывается примерно равной сложности двух умножений. Согласно теореме 2.4, для базиса  $\mathcal{A}^\mathbb{C}$  можно выбрать  $c_F = 1.5$ .

► Сохраняем обозначения леммы 5.2. Пусть  $(k+1)m > n \geq km$ . Рассматривающие многочлены разобьем на блоки длины  $m$ :

$$\tilde{a} = a_0 + a_1x^m + a_2x^{2m} + \dots, \quad \tilde{b} = b_0 + b_1x^m + b_2x^{2m} + \dots, \quad \tilde{q} = q_0 + q_1x^m + q_2x^{2m} + \dots$$

Далее пусть  $(a)^*$  обозначает вектор<sup>2)</sup>  $\text{ДПФ}_{2m}(a)$ .

Положим  $b_0^{-1} = 1/b_0 \pmod{x^m}$ . Из условия  $\tilde{b}\tilde{q} = \tilde{a} \pmod{x^n}$  получается рекуррентная формула для выражения блоков частного  $\tilde{q}$ . При  $0 \leq j < k$  имеем

$$a_j = \left( q_j b_0 + \sum_{i=0}^{j-1} (q_i b_{j-i} + \lfloor q_i b_{j-i-1}/x^m \rfloor) \right) \pmod{x^m}. \quad (5.5)$$

Полагая  $b_{-1} = 0$  и вводя обозначение<sup>3)</sup>  $\beta_i = b_{i-1} + b_i x^m$  для  $0 \leq i \leq k+1$ , получаем

$$q_j = b_0^{-1} \left( a_j - \sum_{i=0}^{j-1} \lfloor q_i \beta_{j-i}/x^m \rfloor \right) \pmod{x^m}, \quad 0 \leq j < k. \quad (5.6)$$

При  $j = k$  формулы (5.5), (5.6) справедливы по модулю  $x^{n-km}$ .

Воспользуемся следующим алгоритмом:

I. Найдем  $q_0$ . Для этого вычислим  $b_0^{-1}$ , затем  $(b_0^{-1})^*$  и  $(a_0)^*$ , далее  $(a_0 b_0^{-1})^*$ , и наконец (посредством обратного ДПФ) восстановим  $q_0 = a_0 b_0^{-1} \pmod{x^m}$ .

II. Далее последовательно при помощи формулы (5.6) определяются блоки  $q_j$ .  
При каждом  $j = 1, \dots, k$ :

1) Вычисляются  $(\beta_j)^*$  и  $(q_{j-1})^*$ . Обозначим

$$\alpha = q_0 \beta_j + q_1 \beta_{j-1} + \dots + q_{j-1} \beta_1. \quad (5.7)$$

2) Вычисляется вектор  $(\alpha)^*$  по известным  $(\beta_i)^*$  и  $(q_i)^*$ , следуя (5.7).

3) При помощи обратного ДПФ вычисляем многочлен  $\alpha \pmod{(x^{2m} - 1)}$ . Заметим, что его старшие  $m$  коэффициентов совпадают со средними  $m$  коэффициентами  $\alpha$ . Обозначим

$$\gamma = a_j - \lfloor \alpha/x^m \rfloor \pmod{x^m}.$$

4) Последовательно вычисляя  $\gamma$ ,  $(\gamma)^*$  и  $(b_0^{-1}\gamma)^*$ , окончательно находим  $q_j = b_0^{-1}\gamma \pmod{x^m}$  при  $j < k$  и  $q_k = b_0^{-1}\gamma \pmod{x^{n-km}}$ .

III. Таким образом, найдено частное  $\tilde{q} = \sum_{i=0}^k q_i x^{im} \pmod{x^n}$ ; остается найти остаток  $\tilde{r}$ . Согласно (5.4), для этого достаточно вычислить недостающую часть произведения  $\tilde{q}\tilde{b}$ . Запишем

$$\tilde{q}\tilde{b} = c_0 + c_1 x^m + c_2 x^{2m} + \dots + c_{2k+1} x^{(2k+1)m},$$

где  $\deg c_i < m$ . По построению,  $c_j = a_j$  для  $0 \leq j < k$  и  $c_k \equiv a_k \pmod{x^{n-km}}$ . Остальные  $c_j$  вычисляются по формулам

$$c_j = \left\lfloor \sum_i q_i \beta_{j-i}/x^m \right\rfloor \pmod{x^m}. \quad (5.8)$$

<sup>2)</sup>Имеется в виду полиномиальная интерпретация ДПФ, когда преобразование применяется к вектору коэффициентов многочлена.

<sup>3)</sup>Многочлены  $\beta_i$  вводятся только для удобства записи.

Для пограничного блока  $c_k$  формула (5.8) дает недостающие  $(k+1)m-n$  старших коэффициентов.

Для вычислений по формулам (5.8) требуется вычислить  $(\beta_{k+1})^*$ ,  $(q_k)^*$ , при всех  $j = k, \dots, 2k+1$  найти  $(\sum_i q_i \beta_{j-i})^*$ , восстановить суммы  $\sum_i q_i \beta_{j-i} \pmod{(x^{2m}-1)}$ , из которых извлекаются блоки  $c_j$ . Окончательно находим  $\tilde{r} = (\tilde{a} - \tilde{q}\tilde{b})/x^n$ .

Оценим сложность алгоритма. Он начинается с инвертирования по модулю  $x^m$ . Далее на всех этапах выполняется  $3k+4$  ДПФ и  $3k+3$  обратных ДПФ порядка  $2m$ . Сложность остальных операций, среди которых доминируют вычисления Фурье-образов сумм (5.7) и (5.8), оценивается как  $O(mk^2)$ .

Целесообразен выбор параметров  $m = 2^t$  и  $k \asymp \sqrt{\log n}$ . Стартовое инвертирование можно выполнить любым алгоритмом сложности  $O(m \log m)$ , например, полиномиальным аналогом метода Кука из предыдущего пункта. В итоге получаем

$$\begin{aligned} C_{\mathcal{A}}(QR_{2n,n}) &\leqslant (6k + 7)c_F \cdot 2m \log_2(2m) + O(m(k^2 + \log m)) = \\ &= 12c_F \cdot n \log_2 n + O(n \sqrt{\log n}). \end{aligned}$$

■

Отметим, что в отличие от метода теоремы 5.1, где на каждом шаге точность удваивалась, метод теоремы 5.2 основан на итерациях (5.6) с фиксированным приращением точности для того, чтобы снизить удельный вес операции инвертирования.

- Отдельно для операции инвертирования по модулю  $x^n$  наилучшая известная оценка сложности  $(7.5 + o(1))c_F \cdot n \log_2 n$  получена автором в [83].

### Умножение матриц. Границный ранг $\boxed{\varepsilon}$



Дарио Андреа Бини  
Университет Пизы, с 1990

Теорема 2.3 опирается на специальный способ умножения матриц размера  $2 \times 2$  и легко обобщается. *Рангом*  $\text{rk}^R T$  системы билинейных форм  $T(X, Y)$  над полукольцом  $R$  называется минимальное число  $r$ , при котором возможно представление

$$T = \sum_{l=1}^r C_l X_l Y_l, \quad (5.9)$$

где  $C_l$  — вектор с компонентами из  $R$ , а  $X_l$  и  $Y_l$  — линейные над  $R$  комбинации переменных  $x_{ij}$  и  $y_{jk}$  соответственно. Например,  $\text{rk}^F MM_2 = 7$  в любом поле  $F$  [331]. Теперь теорема 2.3 может быть расширена как

**Теорема 5.3.** *Пусть  $r = \text{rk } MM_m$  в кольце  $R$ . Тогда*

$$C_{\mathcal{A}^R}(MM_n) \leqslant r \left( 1 + \frac{6rm^2}{r-m^2} \right) n^{\log_m r}.$$

► Рассмотрим некоторое представление вида (5.9) для оператора  $MM_m$ , использующее  $r$  нескалярных умножений и  $s$  линейных операций. Заведомо выполнено  $s \leq 6rm^2$ , поскольку любая сумма  $X_l$  или  $Y_l$  требует не более  $2m^2$  линейных операций, а любой элемент произведения матриц является линейной комбинацией  $r$  произведений  $X_l Y_l$ , следовательно, вычисляется за  $2r$  линейных операций.

По индукции легко проверяется оценка

$$\mathsf{C}(MM_{m^h}) \leq \left(1 + \frac{s}{r - m^2}\right) r^h - \frac{s}{r - m^2} m^{2h} \quad (5.10)$$

(разбиваем  $m^h \times m^h$  матрицу на подматрицы размера  $m^{h-1} \times m^{h-1}$ ). Наконец, при  $m^{h-1} < n \leq m^h$  получаем

$$\mathsf{C}(MM_n) \leq \mathsf{C}(MM_{m^h}) \leq \left(1 + \frac{s}{r - m^2}\right) r^h \leq r \left(1 + \frac{s}{r - m^2}\right) n^{\log_m r}. \quad \blacksquare$$

*Границым рангом*  $\underline{\text{rk}} T$  называется минимальное число  $r$ , при котором возможно представление

$$u^d T = \sum_{l=1}^r C_l(u) X_l(u) Y_l(u) \bmod u^{d+1}, \quad (5.11)$$

где  $d \in \mathbb{N}_0$ ,  $C_l(u) \in R[u]^{\dim T}$ , и  $X_l$  и  $Y_l$  — линейные над  $R[u]$  комбинации переменных  $x_{ij}$  и  $y_{jk}$  соответственно,  $\dim T$  — число форм в системе  $T$ . Если представить, что  $u \rightarrow 0$ , то формулы (5.11) при делении на  $u^d$  задают приближенное матричное умножение: собственно, приближенное вычисление и было целью работы итальянских математиков [144], в которой фактически возникает понятие граничного ранга. Но Д. Бини [143] заметил, что от формул (5.11) можно эффективно перейти к точному умножению матриц.

Далее формулы типа (5.11) будем называть  $(d, r)$ -представлениями.

**Теорема 5.4** ([143]). *Пусть  $R$  — кольцо и  $m = \text{const}$ . Тогда*

$$\mathsf{C}_{\mathcal{A}^R}(MM_n) \leq n^{\log_m(\underline{\text{rk}} MM_m) + o(1)}.$$

► Ключевым пунктом доказательства является лемма Бини (мы доказываем ее в ослабленной форме).

**Лемма 5.3.** *Если система  $T$  имеет  $(d, r)$ -представление, то  $\text{rk } T \leq C_{d+2}^2 \cdot r$ .*

► Пусть  $T$  вычисляется по формулам (5.11). Запишем

$$C_l(u) = \sum_{i=0}^d C_{l,i} u^i, \quad X_l(u) = \sum_{i=0}^d X_{l,i} u^i, \quad Y_l(u) = \sum_{i=0}^d Y_{l,i} u^i,$$

где  $C_{l,i}$  — векторы над  $R$ , а  $X_{l,i}$  и  $Y_{l,i}$  — линейные над  $R$  комбинации переменных. Тогда из (5.11) получаем

$$T = \sum_{l=1}^r \sum_{a+b+c=d} C_{l,a} X_{l,b} Y_{l,c}.$$

Осталось заметить, что внутренняя сумма содержит  $C_{d+2}^2$  членов.  $\square$

Нам еще понадобится простая лемма о композиции  $(d, r)$ -представлений. *Тензорное произведение* билинейных систем  $T_1$  и  $T_2$  определяется следующим образом. Если  $T_1 = \sum C_{i,j} x_i y_j$ , где  $C_{i,j} \in R^{\dim T_1}$ , то

$$T_1 \otimes T_2 = \sum C_{i,j} \otimes T_2(X^i, Y^j), \quad (5.12)$$

где  $X^i, Y^j$  — независимые группы переменных<sup>4)</sup>. В частности,

$$MM_{m,p,q} \otimes MM_{m',p',q'} = MM_{mm',pp',qq'}, \quad (5.13)$$

где  $MM_{m,p,q}$  — оператор умножения матриц размера  $m \times p$  и  $p \times q$  (над  $R$ ).

**Лемма 5.4.** *Если система  $T_1$  имеет  $(d_1, r_1)$ -представление, а система  $T_2$  имеет  $(d_2, r_2)$ -представление, то  $T_1 \otimes T_2$  имеет  $(d_1 + d_2, r_1 r_2)$ -представление.*

▷ Используя представление для  $T_1$ , запишем

$$u^{d_1}(T_1 \otimes T_2) = \sum_{l=1}^{r_1} C_l(u) \otimes T_2(X_l(u), Y_l(u)) \bmod u^{d_1+1}, \quad (5.14)$$

где  $C_l(u) \in R[u]^{\dim T_1}$ , а  $X_l(u)$  и  $Y_l(u)$  — линейные комбинации<sup>5)</sup> векторов переменных  $X^i$  и  $Y^j$ . Далее, подставляя в (5.14) представления для  $T_2$  и домножая на  $u^{d_2}$ , получаем

$$\begin{aligned} u^{d_1+d_2}(T_1 \otimes T_2) &= \sum_{l=1}^{r_1} C_l(u) \otimes \left( \sum_{s=1}^{r_2} C'_s(u) X_{l,s}(u) Y_{l,s}(u) \right) \bmod u^{d_1+d_2+1} = \\ &= \sum_{l=1}^{r_1} \sum_{s=1}^{r_2} (C_l(u) \otimes C'_s(u)) X_{l,s}(u) Y_{l,s}(u) \bmod u^{d_1+d_2+1}, \end{aligned}$$

где  $C'_s(u) \in R[u]^{\dim T_2}$ , а  $X_{l,s}(u)$  и  $Y_{l,s}(u)$  — линейные комбинации элементов векторов  $X_l(u)$  и  $Y_l(u)$ , т.е. в конечном счете просто линейные комбинации переменных.  $\square$

<sup>4)</sup> Символ  $\otimes$  в правой части (5.12) имеет смысл кронекерова произведения матриц (в данном случае, векторов). *Кронекерово произведение*  $m \times n$  матрицы  $A = (a_{i,j})$  и  $p \times q$  матрицы  $B$  определяется как  $mp \times nq$  матрица, получающаяся подстановкой в матрицу  $A$  блоков  $a_{i,j}B$  вместо  $a_{i,j}$ .

<sup>5)</sup> Напомним, что ввиду билинейности  $T_2(aX^1 + bX^2, Y) = aT_2(X^1, Y) + bT_2(X^2, Y)$  и  $T_2(X, aY^1 + bY^2) = aT_2(X, Y^1) + bT_2(X, Y^2)$ .

Завершаем доказательство теоремы. Положим  $r = \underline{\text{rk}} MM_m$ . Применяя  $h$  раз лемму 5.4 к  $(d, r)$ -представлению для  $MM_m$ , получаем  $(hd, r^h)$ -представление для  $MM_{m^h}$ . Тогда из леммы 5.3 выводим  $\text{rk } MM_{m^h} \leq (hd + 2)^2 r^h$ . При помощи теоремы 5.3 окончательно получаем

$$\mathsf{C}(MM_n) \leq 7(hd + 2)^4 r^{2h} m^{2h} n^{\log_m r + \frac{2}{h} \cdot \log_m (hd + 2)}.$$

Подходящим выбором параметра является  $h \asymp \sqrt{\log n \log \log n}$ . ■

Простой иллюстрацией преимущества, которое дает переход к граничному рангу, служит следующий пример А. Шёнхаге [303].

**Теорема 5.5** ([303]). *Если  $R$  — кольцо, то  $\mathsf{C}_{\mathcal{A}^R}(MM_n) \asymp n^{\log_3 21+o(1)} \prec n^{2.772}$ .*

► Покажем, что  $\underline{\text{rk}} MM_3 \leq 21$ . Элементы  $z_{ij}$  произведения  $Z = XY$  матриц размера  $3 \times 3$  можно вычислить по формулам

$$\begin{aligned} u^2 z_{jj} &= (x_{j1} + u^2 x_{j2})(y_{2j} + u^2 y_{1j}) + v_{jj} - w_j \mod u^3, \\ u^2 z_{ij} &= (x_{j1} + u^2 x_{i2})(y_{2j} - uy_{1i}) + v_{ij} - w_j + u(v_{ji} - v_{ii}) \mod u^3, \quad i \neq j, \\ v_{jj} &= (x_{j1} + u^2 x_{j3})y_{3j}, \quad v_{ij} = (x_{j1} + u^2 x_{i3})(y_{3j} + uy_{1i}), \quad w_j = x_{j1}(y_{2j} + y_{3j}), \end{aligned}$$

используя 21 нескалярное умножение. Остается применить теорему 5.4. ■

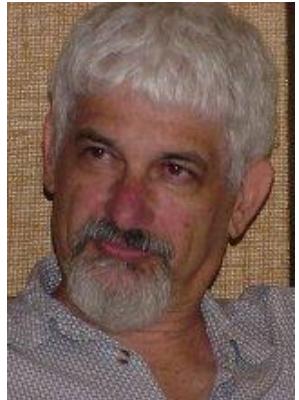
- Рассмотрение  $2 \times 2$  матриц не приводит к усилению оценок теоремы 2.3 ввиду  $\underline{\text{rk}} MM_2 = \text{rk } MM_2 = 7$  [253]. С другой стороны, для ранга умножения  $3 \times 3$  матриц пока известна лишь оценка  $\text{rk } MM_3 \leq 23$  [250]. А. В. Смирнов [96] показал, что  $\underline{\text{rk}} MM_3 \leq 20$  (но соответствующее представление труднее для описания) — при подстановке этой оценки в теорему 5.5 получится  $\mathsf{C}_{\mathcal{A}^R}(MM_n) \prec n^{2.727}$ .

Оценка леммы 5.3 уточняется до  $\text{rk } T \leq (2d + 1)r$  в поле  $F$  порядка не менее  $2d + 2$  [143].

### Монотонные схемы для сортировки $\boxed{\varepsilon} \boxed{P} \boxed{\ddots}$

Вскоре после работы К. Бэтчера [128], предложившего изящный способ построения монотонных схем для сортировки  $n$  входов со сложностью  $O(n \log^2 n)$ , Э. Ламанья и Дж. Сэвидж [252] доказали нижнюю оценку  $\mathsf{C}_{\mathcal{B}_M}(\text{SORT}_n) \geq n \log n$ . Наконец, спустя еще 10 лет М. Айтаян, Я. Комлош и Э. Семереди [119, 120] получили окончательный результат  $\mathsf{C}_{\mathcal{B}_M}(\text{SORT}_n) \asymp n \log n$  и  $\mathsf{D}_{\mathcal{B}_M}(\text{SORT}_n) \asymp \log n$  при помощи конструкции, известной теперь как AKS-схемы. Метод эксплуатирует сразу несколько идей, центральной из которых следует признать идею приближенных вычислений. Схемы сортировки оказалось выгодным строить из подсхем, выполняющих сортировку приближенно, с контролируемым числом ошибок.

Оригинальный метод [119, 120], как и большинство его модификаций, сложен для описания и для анализа. Мы изложим сравнительно



Янош Комлош  
Ратгерский университет,  
с 1988

простой вариант метода, построенный Ж. Сейферасом [307] в развитие подхода М. Патерсона [277].

Рассматриваемые далее схемы относятся к частной модели *схем компараторов* (в терминологии [25], сортирующие сети). Схема компараторов принимает на вход  $n$  элементов линейно упорядоченного множества и при помощи операций парного сравнения  $x, y \rightarrow \min(x, y), \max(x, y)$  (в булевом случае,  $x, y \rightarrow xy, x \vee y$ ) воспроизводит на выходах перестановку входов в соответствии с некоторым частичным порядком. Выходы внутренних подсхем-компараторов не ветвятся. В каком-то смысле, схема компараторов — это почти формула. Схема естественным образом распадается на слои из независимых по входам компараторов. Подробнее см. в [25].



Эндре Семереди  
Институт математики  
Венгерской академии наук,  
Будапешт, с 1965

Введем концепцию приближенной сортировки. Пусть  $0 < \varepsilon \leq 1$  и  $0 < \lambda \leq 1/2$ . Схема компараторов на  $n$  входах называется  $(\lambda, \varepsilon)$ -сепаратором, если при любом  $t \leq \lambda n$  схема размещает не менее  $(1 - \varepsilon)t$  из  $t$  наибольших элементов среди  $\lambda n$  правых выходов и не менее  $(1 - \varepsilon)t$  из  $t$  наименьших элементов — среди  $\lambda n$  левых выходов. Следующие две леммы устанавливают существование сепараторов постоянной глубины (и, следовательно, линейной сложности).

**Лемма 5.5** ([120]). *При любом  $\varepsilon > 0$  и любом  $n$  существует  $(1/2, \varepsilon)$ -сепаратор<sup>6</sup> на  $2n$  входах глубины  $O(1/\varepsilon^3)$  при  $\varepsilon \rightarrow 0$ .*

▷ Если  $n$  мало, скажем,  $n < 64/\varepsilon^3$ , то воспользуемся любой схемой сортировки. Поэтому далее полагаем  $n \geq 64/\varepsilon^3$ .

Покажем, что требуемую схему можно построить из нескольких слоев компараторов, где на каждом слое сравниваются между собой элементы из младшей и старшей половин согласно случайному выбранному паросочетанию. Пример схемы показан на рис. 5.1а.

Сопоставим такой схеме двудольный  $(n, n)$ -граф: вершины одной доли соответствуют позициям элементов из младшей половины списка, а другой — из старшей. Две вершины графа соединяются ребром, если на каком-то слое схемы выполнялось сравнение соответствующих элементов, см. рис. 5.1б. Другими словами, граф является композицией паросочетаний, задающих слои схемы.

I. Сначала проверим, что схема является  $(1/2, \varepsilon)$ -сепаратором, если график является  $(\varepsilon, \alpha)$ -расширителем,  $\alpha = 1/\varepsilon - 1/2$ , что значит: для каждого  $t \leq \varepsilon n$  любое множество из  $t$  вершин одной доли соединено ребрами не менее чем с  $\alpha t$  вершинами другой доли.

Прежде заметим, что если две вершины соединены ребром в графе, то функции, вычисляемые на соответствующих выходах схемы, связаны отношением  $\leq$  (одна доля собирает только минимумы упорядоченных пар, а другая — только максимумы).

<sup>6</sup> $(1/2, \varepsilon)$ -сепараторы также называются  $\varepsilon$ -халверами.

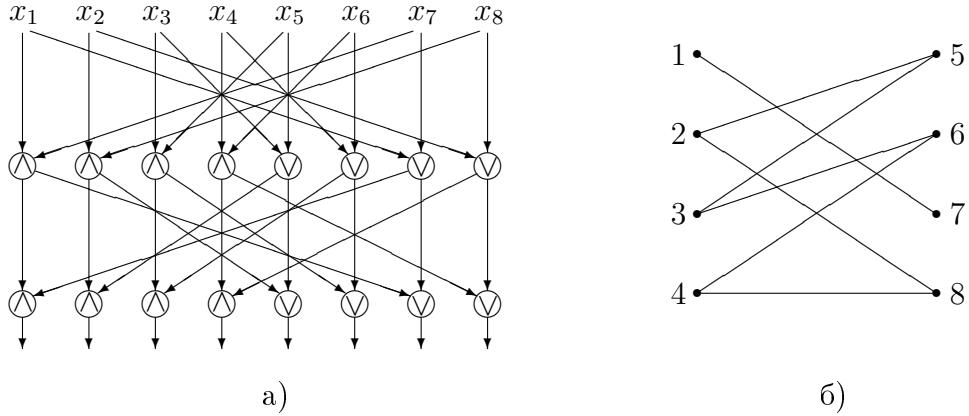


Рис. 5.1: Схема компараторов и ее граф

Теперь предположим, что схема не является сепаратором, т. е. при некотором входном наборе, скажем, среди  $k \leq n$  наибольших элементов  $p > \varepsilon k$  оказываются в младшей половине. Рассмотрим соответствующее этим элементам множество вершин в графе. В этом множестве  $m = \min\{p, \lfloor \varepsilon n \rfloor\}$  вершин соединены ребрами по меньшей мере с  $\alpha m$  вершинами другой доли. Это значит, что минимальный из  $p$  элементов уступает не менее чем  $p - 1 + \alpha m$  другим элементам. Но при  $m = p$  выполнено

$$p - 1 + \alpha m = p - 1 + m/\varepsilon - m/2 > p/\varepsilon - 1 > k - 1,$$

а при  $m = \lfloor \varepsilon n \rfloor < p$ :

$$p - 1 + m/\varepsilon - m/2 > (p - 1)/2 + m/\varepsilon > (\varepsilon n - 1)(1/\varepsilon + 1/2) > (1 + \varepsilon/2)n - 1/\varepsilon - 1 \geq n - 1.$$

Это противоречит тому, что выбранный элемент находится среди  $k$  наибольших.

*II.* Осталось доказать, что двудольный граф, составленный из подходящего числа  $r = r(\varepsilon)$  случайных паросочетаний<sup>7)</sup>, с ненулевой вероятностью является  $(\varepsilon, \alpha)$ -расширителем.

Заметим, что граф является  $(\varepsilon, \alpha)$ -расширителем, если он не содержит пустых (т.е. безреберных)  $(k, n - \alpha k)$ -подграфов при любом  $k \leq \varepsilon n$ . Вероятность того, что случайное паросочетание не пересекает (по ребрам) данный  $(k, n - \alpha k)$ -подграф, равна  $C_{\alpha k}^k / C_n^k$ . Тогда вероятность  $P_k$  того, что  $r$  случайных паросочетаний не пересекают хотя бы один из  $(k, n - \alpha k)$ -подграфов, при помощи простых соотношений  $\frac{1}{4\sqrt{k}} \left(\frac{\varepsilon n}{k}\right)^k \leq C_n^k \leq \left(\frac{\varepsilon n}{k}\right)^k$  оценивается как

$$\begin{aligned} P_k &\leq 2C_n^k C_n^{\alpha k} \left( \frac{C_{\alpha k}^k}{C_n^k} \right)^r \leq 2(4\sqrt{k})^r \left( \frac{e^{1+\alpha} n^{1+\alpha} (\alpha k)^r}{k^{1+\alpha} \alpha^\alpha n^r} \right)^k = \\ &= 2(4\sqrt{k})^r \left( \alpha e^{1+\alpha} \left( \frac{\alpha k}{n} \right)^{r-\alpha-1} \right)^k \leq \left( c_2 \left( \frac{c_1 \alpha k}{n} \right)^{r-\alpha-1} \right)^k, \end{aligned}$$

<sup>7)</sup>Распределение равномерное: все сочетания равновероятны.

где  $c_1 = (4\sqrt{k})^{1/k} \leq 4$  и  $c_2 = 2^{1/k}\alpha(c_1e)^{1+\alpha} \leq (8e)^{1+\alpha}$ .

При  $k \leq \varepsilon n/4$  выполнено  $c_1\alpha k \leq (1 - \varepsilon/2)n$ . Иначе  $k \geq \varepsilon n/4 \geq 16/\varepsilon^2$ , поэтому  $c_1 = e^{\ln(16k)/2k} < 1 + \ln(16k)/k \leq 1 + \frac{\varepsilon^2}{8} \ln \frac{16}{\varepsilon} \leq 1 + \varepsilon/2$ . Следовательно,  $c_1\alpha k < (1 - \varepsilon^2/4)n$ . В любом из двух случаев, если  $r$  выбрано несколько большим, чем  $\alpha + 4\ln(2c_2)/\varepsilon^2$ , получаем  $P_k < 2^{-k}$ . Тогда вероятность того, что рассматриваемый граф не является  $(\varepsilon, \alpha)$ -расширителем, не превосходит  $\sum_k P_k < 1$ .  $\square$

- Схема, существование которой устанавливает лемма, имеет глубину  $r \asymp 1/\varepsilon^3$  при  $\varepsilon \rightarrow 0$ . Более аккуратное рассуждение позволяет уточнить оценку до  $r \preccurlyeq \frac{1}{\varepsilon} \log \frac{1}{\varepsilon}$ , см., например, [277].

Легко проверить, что любой  $r$ -регулярный (все вершины имеют степень  $r$ ) двудольный граф является объединением  $r$  паросочетаний, поэтому схему можно построить из графа. Так же известны многие явные конструкции регулярных графов-расширителей, например, [51, 196].

**Лемма 5.6** ([277]). *При любом постоянном  $\varepsilon > 0$ , любых  $\lambda < 1/2$  и  $n$  существует  $(\lambda, \varepsilon)$ -сепаратор на  $2n$  входах глубины  $O(\log^4(1/\lambda))$  при  $\lambda \rightarrow 0$ .*

▷ Положим  $s = \lfloor 2\lambda n \rfloor$  и  $k = \lceil \log_2(1/\lambda) \rceil$ . Схема строится из  $k+1$  слоя  $(1/2, \varepsilon_0)$ -сепараторов (параметр  $\varepsilon_0$  будет выбран позднее): на первом слое — сепаратор на  $2n$  входах, на следующих двух — два сепаратора слева и справа для  $2^{k-1}s$  крайних элементов<sup>8</sup>, на следующем слое — сепараторы для  $2^{k-2}s$  элементов с каждого края и т.д. вплоть до последнего слоя из двух сепараторов на  $2s$  крайних элементах, см. рис. 5.2.

Из  $m \leq s$  наибольших (наименьших) элементов сепаратор первого слоя определяет в «неправильную» половину не более  $\varepsilon_0 m$  штук. В следующей паре сепараторов независимо от порядка их расположения: сепаратор правых (левых)  $2^{k-1}s$  элементов оставляет за пределами крайнего интервала еще максимум  $\varepsilon_0 m$  элементов, а сепаратор с другой стороны не ухудшает характеристики отсечения наибольших (наименьших) элементов<sup>9</sup>. На каждом из последующих слоев сепаратор с соответствующей стороны ошибочно выбрасывает из крайнего интервала еще не более  $\varepsilon_0 m$  элементов.

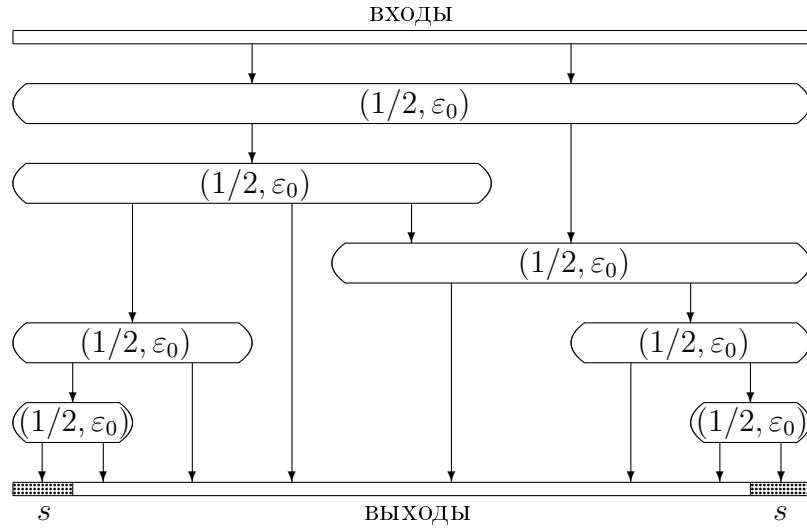
Таким образом, всего не более  $k\varepsilon_0 m$  наибольших (наименьших) элементов могут оказаться вне  $s$  крайних правых (левых) выходов схемы. Поэтому достаточно выбрать  $\varepsilon_0 = \varepsilon/k$ . Теперь оценка глубины схемы следует из леммы 5.5.  $\square$

**Теорема 5.6** ([119, 120]). *Сортировка  $n$  элементов выполняется схемой компараторов глубины  $O(\log n)$ .*

► Для простоты рассуждений будем полагать  $n = 2^k$ . Пусть  $\mu, \varepsilon > 0$  — параметры, которые будут выбраны позже. Схема сортировки строится из слоев параллельно расположенных  $(\lambda, \varepsilon)$ -сепараторов, где  $\lambda$  определяется индивидуально для каждого слоя, при этом  $\lambda \geq \mu$ . Удобно представить схему функционирующей

<sup>8</sup>Если  $2^{k-1}s = n$ , то эти два сепаратора можно расположить параллельно на одном слое.

<sup>9</sup>В любой схеме компараторов множество  $m$  наибольших (наименьших) элементов перемещается строго вправо (влево).

Рис. 5.2: Конструкция  $(\lambda, \varepsilon)$ -сепаратора

во времени: преобразования одного слоя выполняются за одну единицу времени. Действие слоя сепараторов мы рассматриваем как перегруппировку элементов в структуре, ассоциированной с полным двоичным деревом глубины  $k$ . При каждой вершине дерева имеется контейнер для хранения элементов.

В начальный момент времени  $t = 0$  все  $n$  элементов находятся в контейнере при корневой вершине. Далее, в любой момент времени элементы каждого непустого контейнера подвергаются приближенной сортировке при помощи сепаратора: элементы из крайних интервалов отправляются на уровень выше к корню, оставшиеся распределяются поровну между контейнерами дочерних вершин, в направлении листьев.

Определим емкость контейнера на глубине  $d$  в момент времени  $t$  как  $B_{d,t} = na^d\nu^t$ , где постоянные параметры  $a > 1$  и  $\nu < 1$  будут определены позже. Если в контейнере находится  $b$  элементов, то в случае  $\mu B_{d,t} < b/2$  содержимое контейнера, за исключением, быть может, нечетного элемента, упорядочивается при помощи композиции  $(1/2, \varepsilon)$ - и  $(\lambda, \varepsilon)$ -сепараторов,  $\lambda = \mu B_{d,t}/b$ , после чего по  $\lfloor \mu B_{d,t} \rfloor$  элементов с каждого края, а также нечетный элемент (при наличии), отправляются на уровень выше. Остальные элементы опускаются на уровень ниже: левая половина — в контейнер левой дочерней вершины, правая — в контейнер правой. Иначе, в случае  $\mu B_{d,t} \geq b/2$ , сепарация не производится — все элементы возвращаются в родительский контейнер. Исключение составляет корневой контейнер — его элементы просеиваются при помощи  $(1/2, \varepsilon)$ -сепаратора, делятся поровну на две части (число элементов в контейнере обязательно четно), которые отправляются в соответствующие контейнеры дочерних вершин. Процесс продолжается до тех пор, пока все элементы не окажутся на нижних  $O(1)$  уровнях дерева; точное условие завершения формулируется как  $B_{k,t} < 1/\mu$ . После этого применяются схемы сортировки в каждом поддереве глубины  $O(1)$ . Построенная схема содержит  $k \log_{1/\nu}(2a) + O(1) \asymp \log n$  слоев сепараторов и поэтому имеет за-

явленную глубину. Осталось проверить корректность ее функционирования.

Прежде всего заметим, что в любой момент времени контейнеры одного уровня заполнены одинаково. В частности, поэтому в корневом контейнере всегда четное число элементов. Кроме того, в контейнерах листьев не может оказаться более одного элемента<sup>10)</sup>. Следовательно, описанная процедура не выводит элементы за пределы дерева.

Также из построения ясно, что в любой момент все элементы сосредоточены либо на четных, либо на нечетных уровнях дерева.

*I.* Индукцией по  $t$  докажем, что при определенных условиях на параметры  $a, \varepsilon, \mu, \nu$  число элементов в любом контейнере никогда не превосходит его емкости. Утверждение очевидно при  $t = 0$ .

Рассмотрим произвольный контейнер на глубине  $d$ , пустой в момент времени  $t - 1$ . Число элементов в следующий момент времени в нем в случае  $B_{d,t} \geq a\nu$  не превосходит

$$2(2\mu B_{d+1,t-1} + 1) + B_{d-1,t-1}/2 = B_{d,t}(4\mu a + 1/(2a))/\nu + 2 < B_{d,t}(4\mu a + 3/a)/\nu,$$

что меньше  $B_{d,t}$  при условии

$$\underline{\nu \geq 4\mu a + 3/a}. \quad (5.15)$$

В оставшемся случае  $B_{d,t} < a\nu$  в момент времени  $t - 1$  все контейнеры на вышестоящих уровнях  $d' < d$  пусты, т.к.  $B_{d',t-1} < 1$ . Это значит, что все элементы находятся на уровнях  $d+1$  и ниже. При этом  $B_{d+1,t-1} < a^2$ , и при дополнительном предположении

$$\underline{a^2 = 1/\mu} \quad (5.16)$$

заключаем, что  $d + 1 \neq k$  (иначе процесс построения схемы был бы уже закончен). Значит, в каждом поддереве с корнем в вершине глубины  $d + 1$  в момент  $t - 1$  находится четное число элементов, следовательно, четное число элементов находится в корне поддерева (дочерней вершине по отношению к рассматриваемой). Таким образом, в контейнер глубины  $d$  в момент времени  $t$  отправляется не более  $4\mu B_{d+1,t-1} = 4\mu a B_{d,t}/\nu < B_{d,t}$  элементов согласно (5.15).

*II.* Доказательство корректности алгоритма основано на оценке числа посторонних элементов в каждом контейнере. Мы полагаем, что при размещении в листьях дерева элементы должны быть упорядочены в порядке возрастания слева направо. Для любого элемента *родными* вершинами мы считаем лист дерева, в котором элемент должен находиться после упорядочения, а также все вершины на пути от корня дерева к этому листу. В ходе выполнения алгоритма элемент некоторого контейнера будем называть *r-чужаком*, если он находится на расстоянии  $\geq r$  по ребрам дерева от ближайшей родной вершины.

Индукцией по  $t$  проверим, что в момент времени  $t$  число *r-чужаков* ( $r \geq 1$ ) в контейнере глубины  $d$  не превосходит  $\varepsilon^{r-1}\mu B_{d,t}$  при подходящем выборе параметров. База индукции тривиальна, т.к. в момент  $t = 0$  все элементы находятся в корневом, родном для них, контейнере. Докажем индуктивный переход.

---

<sup>10)</sup>Эти элементы должны отправиться вверх. Однако в действительности процесс построения схемы закончится еще раньше.

*III.* Сначала рассмотрим более простой случай  $r \geq 2$  (тогда можно полагать  $d \geq 2$ ). В разряд  $r$ -чужаков контейнера глубины  $d$  могут попасть  $(r+1)$ -чужаки из контейнеров дочерних вершин и  $(r-1)$ -чужаки из родительского контейнера в предыдущий момент времени. С учетом фильтрации их число оценивается сверху как

$$2\varepsilon^r \mu B_{d+1,t-1} + \varepsilon(\varepsilon^{r-2} \mu B_{d-1,t-1}) = \varepsilon^{r-1} \mu B_{d,t}(2\varepsilon a + 1/a)/\nu,$$

т.е. не превосходит  $\varepsilon^{r-1} \mu B_{d,t}$  при условии

$$\underline{\nu \geq 2\varepsilon a + 1/a}. \quad (5.17)$$

*IV.* Теперь рассмотрим случай  $r = 1$ . Чужаками в контейнере глубины  $d$  при вершине  $v$  могут стать 2-чужаки от дочерних вершин, а из родительской вершины — как чужаки, так и элементы, родные для другой ее дочерней вершины  $v'$ , в предыдущий момент времени. Число чужаков от дочерних вершин легко оценить как  $2\varepsilon \mu B_{d+1,t-1} = 2\varepsilon \mu a B_{d,t}/\nu$ . Более внимательного рассмотрения требует родительский контейнер.

Пусть родительский контейнер в момент  $t-1$  содержит  $q$  элементов, из них  $q_0$  родных для  $v$  элементов,  $q_1$  — родных для  $v'$ , а также  $q_2$  чужаков. Если  $q_0 \geq q/2$ , то число чужаков для вершины  $v$ , отправляемых в ее контейнер, оценивается как  $\varepsilon(q_1 + q_2) \leq \varepsilon q/2$ , т.е. как сумма ошибок  $(\lambda, \varepsilon)$ -сепаратора, не отправившего чужаков наверх, и  $(1/2, \varepsilon)$ -сепаратора, отправившего родные для  $v'$  элементы в неправильную половину. Иначе, если  $q_0 < q/2$ , то это число приходится оценивать уже как  $\varepsilon q/2 + (q/2 - q_0)$ , где второе слагаемое учитывает родные для  $v'$  элементы, которые оказываются в половине вершины  $v$  даже при правильной сортировке. Оценим величину  $q/2 - q_0$ .

Рассмотрим специальное гипотетическое распределение элементов по контейнерам в момент времени  $t-1$  с тем же числом элементов в каждом контейнере, что и в реальном распределении. Рассортируем и равномерно распределим все элементы между вершинами на уровне  $d$  (вершин  $v$  и  $v'$ ), а затем произвольно переместим элементы вверх и вниз по дереву для правильного заполнения всех контейнеров, но так, чтобы в родительскую по отношению к  $v$  и  $v'$  вершину попало соответственно  $\lceil q/2 \rceil$  и  $\lfloor q/2 \rfloor$  элементов от этих дочерних вершин.

В построенном распределении поддерево с корнем в вершине  $v$  содержит только родные для нее элементы, а в контейнере родительской к  $v$  вершине находится  $\geq q/2$  родных для  $v$  элементов. Оценим максимальное число родных для  $v$  элементов, которые можно переместить из этого контейнера в какие-либо другие. Так получим оценку для  $q/2 - q_0$ .

Применительно к контейнерам того же уровня  $d-1$ : для одного контейнера указанные элементы будут 1-чужаками, для двух — 2-чужаками, для четырех — 3-чужаками и т.д. Общее число доступных для размещения позиций на этом уровне оценивается как

$$\mu(1 + 2\varepsilon + (2\varepsilon)^2 + \dots) B_{d-1,t-1} < \mu B_{d-1,t-1}/(1 - 2\varepsilon). \quad (5.18)$$

На произвольном вышестоящем уровне  $d-h$  для одного контейнера рассматриваемые элементы будут родными, для другого — 1-чужаками, еще для двух —

2-чужаками, для четырех — 3-чужаками и т.д. Общее число доступных позиций на этих уровнях (при нечетных  $h \geq 3$ ) оценивается как

$$\begin{aligned} (1 + \mu(1 + 2\varepsilon + (2\varepsilon)^2 + \dots)) \sum_{i=1}^{d/2} B_{d-2i-1, t-1} < \\ \left(1 + \frac{\mu}{1 - 2\varepsilon}\right) \sum_{i \geq 1} a^{-2i} B_{d-1, t-1} = \left(1 + \frac{\mu}{1 - 2\varepsilon}\right) \frac{B_{d-1, t-1}}{a^2 - 1}. \end{aligned} \quad (5.19)$$

Поскольку в поддереве вершины  $v$  свободных для заполнения позиций уже нет, на произвольном нижестоящем уровне  $d+h$  доступно  $2^h$  контейнеров<sup>11</sup>, для которых указанные элементы будут  $(h+1)$ -чужаками,  $2^{h+1}$  контейнеров, для которых эти элементы будут  $(h+2)$ -чужаками, и т.д. Общее число возможных позиций не превосходит

$$\begin{aligned} \sum_{i=1}^{(k-d)/2} \mu((2\varepsilon)^{2i-1} + (2\varepsilon)^{2i} + \dots) B_{d+2i-1, t-1} < \\ \mu \sum_{i \geq 1} \frac{(2\varepsilon)^{2i-1}}{1 - 2\varepsilon} a^{2i-1} B_{d, t-1} = \frac{2\varepsilon a \mu B_{d, t-1}}{(1 - 2\varepsilon)(1 - (2a\varepsilon)^2)}. \end{aligned} \quad (5.20)$$

Суммируя (5.18), (5.19), (5.20), получаем

$$q/2 - q_0 < \left( \frac{1}{a^2 - 1} + \frac{\mu a^2}{(1 - 2\varepsilon)(a^2 - 1)} + \frac{2\varepsilon a^2 \mu}{(1 - 2\varepsilon)(1 - (2a\varepsilon)^2)} \right) B_{d-1, t-1}.$$

Как следствие, общее число чужаков в контейнере вершины  $v$  в момент  $t$  с учетом поступающих из дочерних вершин оценивается как

$$\left( 2\varepsilon a \mu + \frac{1}{a^3 - a} + \frac{\mu a}{(1 - 2\varepsilon)(a^2 - 1)} + \frac{2\varepsilon a \mu}{(1 - 2\varepsilon)(1 - (2a\varepsilon)^2)} \right) B_{d, t} / \nu,$$

что не превосходит  $\mu B_{d, t}$  при условии

$$\nu \geq 2\varepsilon a + \frac{1}{\mu(a^3 - a)} + \frac{a}{(1 - 2\varepsilon)(a^2 - 1)} + \frac{2\varepsilon a}{(1 - 2\varepsilon)(1 - (2a\varepsilon)^2)}. \quad (5.21)$$

V. Теперь легко видеть, что в момент  $t$  окончания процедуры построения схемы ни в одном контейнере нет чужаков (в предположении, что параметры выбраны верно). Действительно, в произвольном контейнере глубины  $d$  согласно доказанному выше — не более  $\mu B_{d, t} \leq \mu B_{k, t} < 1$  чужаков.

При этом в силу (5.16)  $B_{d, t} = a^{d-k} B_{k, t} < a^{d-k}/\mu \leq 1$  при  $d \leq k-2$ , значит, все элементы находятся в нижних двух слоях дерева.

Осталось указать выбор параметров, удовлетворяющий всем необходимым условиям (5.15), (5.16), (5.17), (5.21). Например, подходит  $\varepsilon = \mu = 1/100$ ,  $a = 10$ ,  $\nu = 0.7$ . ■

---

<sup>11</sup>В поддереве с корнем  $v'$ .

**Следствие 5.1** ([119, 120]).  $C_{\mathcal{B}_M}(\text{SORT}_n) \asymp \log n$ ,  $D_{\mathcal{B}_M}(\text{SORT}_n) \asymp n \log n$ .

- Мультипликативная константа в оценке глубины теоремы 5.6 безумно велика. В ряде работ предпринимались попытки уменьшить ее. Опубликованные с доказательствами оценки (для разнообразных модификаций алгоритма) имеют величину порядка нескольких тысяч, см., например, [169, 205, 277]. Некоторые прикидки допускают существование схем с глубиной в районе  $100 \log_2 n$  (упоминается в [307]). На практике лучшие результаты дают варианты схем Бэтчера [128] теоретической глубины порядка  $\log^2 n$ .

## Другие приложения

**Быстрое вычисление логарифма и экспоненты.** В отличие от простых арифметических действий, быстрое вычисление тригонометрических, логарифмических и показательных числовых функций с заданной точностью требует более изощренной техники.

Теоретически быстрые алгоритмы вычисления логарифма (с точностью до  $n$  знаков) имеют сложность порядка  $M(n) \log n$ ; первые из них предложены Ю. Саламином и Р. Брентом [157]. Они опираются на процедуру нахождения *арифметико-геометрического среднего* двух чисел АГС( $a, b$ ). По определению,

$$\text{АГС}(a, b) = \lim_{k \rightarrow \infty} a_k = \lim_{k \rightarrow \infty} b_k, \quad \text{где } a_0 = a, \quad b_0 = b, \quad a_{k+1} = (a_k + b_k)/2, \quad b_{k+1} = \sqrt{a_k b_k}.$$

Последовательности  $\{a_n\}, \{b_n\}$  сходятся с квадратичной скоростью при  $0 \leq b \leq a \leq 1$ : при  $a_k \gg b_k$  выполнено

$$\frac{b_{k+1}}{a_{k+1}} = \frac{2\sqrt{b_k/a_k}}{1 + b_k/a_k} \approx \frac{2}{\sqrt{b_k/a_k}},$$

а при  $a_k \asymp b_k$  справедливо

$$0 \leq a_{k+1} - b_{k+1} = \frac{a_{k+1}^2 - b_{k+1}^2}{2a_{k+2}} = \frac{(a_k - b_k)^2}{8a_{k+2}} \leq \frac{(a_k - b_k)^2}{8\text{АГС}(a, b)}.$$

Поэтому вычисление АГС( $a, b$ ) с точностью  $2^{-n}$  требует  $\log_2(a/b) + \log_2 n + O(1)$  итераций при небольших  $a, b$ .

Формула Гаусса связывает АГС с величиной эллиптического интеграла первого рода:

$$\frac{\pi}{2\text{АГС}(a, b)} = I(a, b) = \int_0^{\pi/2} \frac{d\tau}{\sqrt{a^2 \cos \tau + b^2 \sin \tau}} = \int_0^{+\infty} \frac{dx}{(x^2 + a^2)(x^2 + b^2)}. \quad (5.22)$$

Роль эллиптических интегралов сводится лишь к обоснованию формулы

$$|\ln(4/b) - I(1, b)| = O(b^2) \quad \text{при } b \rightarrow 0+. \quad (5.23)$$

Формула (5.23) служит руководством к построению алгоритма вычисления  $\ln X$ . Сдвигом позиции запятой можно сделать аргумент достаточно большим:  $2^k X \in [2^n, 2^{n+1}]$ . Тогда с высокой точностью  $\ln(2^k X) \approx I(1, b)$ , где  $b = 2^{2-k}/X$ . Окончательно,  $\ln X$  вычисляется как

$$\ln X \approx \frac{\pi}{2\text{АГС}(1, b)} - k \ln 2,$$

где при схемной реализации константы  $\pi$  и  $\ln 2$  (точнее, их приближенные значения) считаются известными. Сложность метода определяется  $2 \log_2 n + O(1)$  итерациями вычисления АГС( $1, b$ ), в которых используются арифметические действия (в том числе, извлечение корня) с  $O(n)$ -разрядными числами. Сложность одной итерации —  $O(M(n))$ .

Существуют многочисленные вариации описанного метода, некоторые из которых описываются на отличные от (5.22), (5.23) тождества. Оптимизированный алгоритм вычисления логарифма описан Д. Бернштейном в [137].

Имея быстрый способ вычисления логарифма, вычисление экспоненты  $e^Y$  можно реализовать методом последовательных приближений Ньютона, решая уравнение  $f(x) = Y - \ln x$  при помощи итераций  $x_{k+1} = x_k(Y + 1 - \ln x_k)$ . Метод также имеет сложность  $M(n) \log n$ .

На указанной базе строится вычисление тригонометрических и многих трансцендентных функций, подробнее см., например, в [160].

# Глава 6

## Алгебраический метод

$A$

Суть алгебраического метода состоит в переносе вычислений из исходной алгебраической структуры в некоторую другую при помощи преобразования, сохраняющего операции. Выигрыш получается, если во второй структуре интересующие операции выполняются проще, а сам переход не очень сложен.

### Умножение булевых матриц $A$

Рассмотрим умножение булевых матриц над булевым полукольцом  $(\mathbb{B}, \vee, \wedge)$ . Популярным приложением этой операции является построение транзитивного замыкания графов — определение связных компонент. Как известно, в базисе операций полукольца тривиальная верхняя оценка сложности  $C_{\mathcal{B}_M}(MM_n) \leq 2n^3 - n^2$  неулучшаема [276]. Ситуация меняется, если допустить расширение вычислительного базиса до полного. Сразу после появления метода Штрассена [316] быстрого умножения матриц, ряд исследователей (например, [101, 190]) заметили, что при выполнении булева умножения выгодно перейти к целочисленному.

**Лемма 6.1** ([190]).  $C(MM_n^{(\mathbb{B}, \vee, \wedge)}) \leq \log^2 n \cdot C_{\mathcal{A}^R}(MM_n^{\mathbb{Z}_{n+1}})$ .

▷ Погрузив булевые коэффициенты матриц в кольцо  $\mathbb{Z}_{n+1}$ , выполним умножение над этим кольцом. В конце выполним обратный переход, проверяя коэффициенты произведения на равенство нулю. Остается заметить, что сложность арифметических операций в кольце  $\mathbb{Z}_{n+1}$ , во всяком случае, не более чем квадратична по числу разрядов.  $\square$

**Следствие 6.1.**  $C(MM_n^{(\mathbb{B}, \vee, \wedge)}) \leq n^{\omega+o(1)}$ , где  $\omega < 2.38$  — экспонента сложности матричного умножения.

- Небольшое преимущество с точки зрения сложности дает использование вместо  $\mathbb{Z}_{n+1}$  прямого произведения  $\mathbb{Z}_{p_1} \times \dots \times \mathbb{Z}_{p_s}$  кольцо вычетов по нескольким взаимно простым модулям.

В общем случае задача быстрого умножения матриц над монотонными полукольцами не решена. Например, открытым остается вопрос о возможности умножения матриц над тропическими полукольцами  $(\mathbb{R}, \min, +)$ ,  $(\mathbb{R}, \max, +)$  с субкубической сложностью  $n^{3-\Omega(1)}$  в немонотонном базисе.

## Сложность формул для сложения по модулю 7 $\boxed{A}/_2$

Рассмотрим задачу построения коротких формул в базисе  $\mathcal{B}_2$  для оператора  $\text{MOD}_n^7(x)$  сложения по модулю 7. Применение формулы (4.1) приводит к оценке  $\Phi_{\mathcal{B}_2}(\text{MOD}_n^m) \preccurlyeq \Phi_{\mathcal{B}_0}(\text{MOD}_n^m) \preccurlyeq n^{1+\log_2 m}$ . Очевидно, что данный способ не раскрывает всех возможностей базиса  $\mathcal{B}_2$ .

Для полного бинарного базиса У. МакКолл в [265] предложил чуть более экономную формулу

$$\text{MOD}_{n_1+n_2}^{m,r}(X) = \bigwedge_{k=1}^{m-1} (\text{MOD}_{n_1}^{m,k}(X^1) \sim \text{MOD}_{n_2}^{m,r-k}(X^2)), \quad (6.1)$$

где « $\sim$ » означает булеву операцию эквивалентности и  $X = (X^1, X^2)$ ,  $|X^i| = n_i$ . Вычисление по этой формуле приводит к верхней оценке

$$\Phi_{\mathcal{B}_2}(\text{MOD}_n^m) \preccurlyeq n^{1+\log_2(m-1)}, \quad (6.2)$$

которая при  $m = 3$  пока остается рекордной<sup>1)</sup>. При  $m = 7$  формула (6.1) ведет лишь к оценке  $\Phi_{\mathcal{B}_2}(\text{MOD}_n^7) \prec n^{3.59}$ .

Д. ван Лейенхорст [255] предложил в случае  $m = 7$  перейти к вычислениям в мультипликативной группе поля  $\mathbb{F}_8$  с представлением ее элементов двоичными матрицами размера  $3 \times 3$ . Эта группа изоморфна группе  $(\mathbb{Z}_7, +)$ . Групповая операция в выбранном представлении группы  $\mathbb{F}_8^*$  является обычным умножением матриц над  $\mathbb{F}_2$ .

**Теорема 6.1** ([255]).  $\Phi_{\mathcal{B}_2}(\text{MOD}_n^7) \prec n^{2.59}$ .

► Пусть  $(7, 4)$ -оператор  $\pi : (\mathbb{Z}_7, +) \rightarrow \mathbb{F}_8^*$  выполняет переход между двумя представлениями по правилу  $r \rightarrow g^r$ , где  $g$  — порождающий элемент группы  $\mathbb{F}_8^*$ . Положим  $H_n(X) = g^{\sum_{i=1}^n x_i}$ . Оператор  $H_n(X)$  вычисляется рекурсивно по правилу умножения матриц

$$H_{n_1+n_2}[i, j](X) = \bigoplus_{k=0}^2 H_{n_1}[i, k](X^1) \cdot H_{n_2}[k, j](X^2), \quad (6.3)$$

откуда получаем  $\Phi_{\mathcal{B}_2}(H_n) \preccurlyeq n^{\log_2 6}$ . Но при этом виду

$$\text{MOD}_n^7(x) = \pi^{-1}(H_n(\pi(x_1), \dots, \pi(x_n)))$$

и  $\Phi_{\mathcal{B}_2}(\pi, \pi^{-1}) = O(1)$  также выполнено  $\Phi_{\mathcal{B}_2}(\text{MOD}_n^7) \preccurlyeq \Phi_{\mathcal{B}_2}(H_n)$ , следовательно,  $\Phi_{\mathcal{B}_2}(\text{MOD}_n^7) \prec n^{2.59}$ . ■

- Аналогичный прием по отношению к вычислению сумм по модулям 3 и 5 состоит в переходе к вычислениям в полях  $\mathbb{F}_4$  и  $\mathbb{F}_{16}$  соответственно и не приводит к улучшению оценки (6.2). Вычисление по правилам (6.3) ведет к оценке глубины  $D_{\mathcal{B}_2}(\text{MOD}_n^7) \lesssim 3 \log_2 n$ , которая незначительно улучшена автором в [89] до  $D_{\mathcal{B}_2}(\text{MOD}_n^7) \lesssim 2.93 \log_2 n$  при помощи использования разбиений переменных на три группы и специальной кодировки.

---

<sup>1</sup>При  $m \geqslant 5$  эта оценка уступает общей оценке сложности для класса симметрических функций  $\Phi_{\mathcal{B}_2}(S_n) \prec n^{2.84}$  [88].

### Умножение чисел при помощи ДПФ $A[\varepsilon]$

В 1963 г. А. Л. Тоом [98] не только обобщил метод Карацубы, но и предложил концепцию, которой с тех пор следуют все быстрые алгоритмы умножения. При помощи разбиения чисел на блоки числовое умножение превращается в умножение многочленов над подходящим образом выбранным кольцом  $R$ . Умножение в кольце  $R[x]$  при помощи интерполяции сводится к покомпонентному умножению в кольце  $R^N$ , где  $N$  — число точек интерполяции.

Согласно [50], первый быстрый алгоритм умножения чисел, основанный на ДПФ, был построен Н. С. Бахваловым — его метод имел сложность  $O(n \log^3 n)$ . Затем, в 1971 г. А. Шёнхаге и Ф. Штрассен [305] опубликовали сразу два быстрых метода умножения. Первый из них эксплуатирует естественную идею перехода к вычислениям над полем комплексных чисел  $\mathbb{C}$ , которое допускает ДПФ произвольного порядка.

**Теорема 6.2** ([305]).  $C(M_n) \leq n \log n \log \log n \dots (\log^{(d)} n)^2$  при любом  $d = O(1)$ .

► Пусть  $2n = 2^k q$ . Разобьем перемножаемые  $n$ -разрядные числа  $A$  и  $B$  на блоки длины  $q$  и при помощи замены  $2^q \rightarrow x$  перейдем к многочленам:

$$A \rightarrow A(x) = \sum_{i=0}^{2^{k-1}-1} a_i x^i, \quad B \rightarrow B(x) = \sum_{i=0}^{2^{k-1}-1} b_i x^i.$$

Искомое произведение  $AB$  восстанавливается из произведения многочленов  $C(x) = A(x)B(x)$  обратной подстановкой  $x = 2^q$  со сложностью порядка  $2^k(2q+k)$ , поскольку коэффициенты  $C(x)$  являются  $(2q+k)$ -разрядными числами.

Умножение многочленов интерпретируем как умножение в кольце  $\mathbb{C}[x]$ , где операции выполняются с такой точностью, чтобы коэффициенты многочленапроизведения, которые на самом деле являются целыми числами, были найдены с ошибкой  $< 1/2$ ; тогда их можно восстановить путем округления.

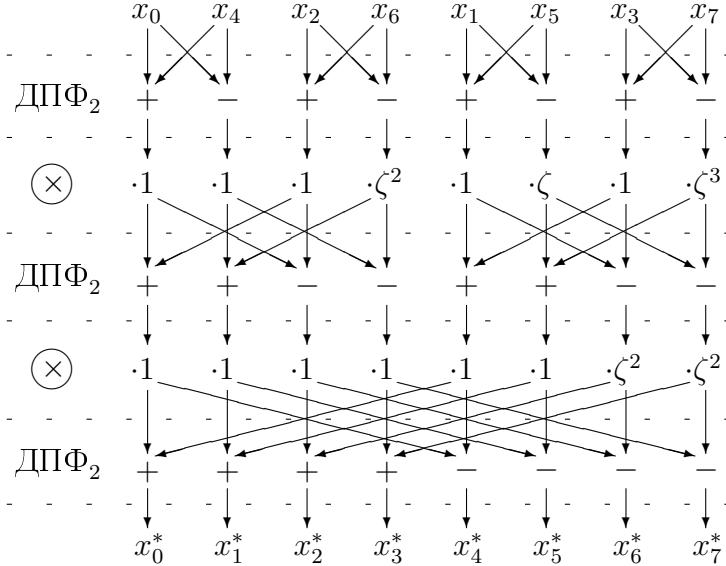
Произведение многочленов вычисляется при помощи ДПФ как<sup>2)</sup>

$$C(x) = \text{ДПФ}_{2^k, \zeta}^{-1} (\text{ДПФ}_{2^k, \zeta}(A(x)) \odot \text{ДПФ}_{2^k, \zeta}(B(x))), \quad \text{ДПФ}_{2^k, \zeta}^{-1} = 2^{-k} \cdot \text{ДПФ}_{2^k, \zeta^{-1}}, \quad (6.4)$$

а сами ДПФ выполняются методом теоремы 2.4. Напомним, что стандартная схема ДПФ порядка  $2^k$  состоит из  $k$  слоев, на которых параллельно выполняются ДПФ порядка 2 (это просто сложение и вычитание пары комплексных чисел), между которыми располагаются слои параллельных умножений на корни из единицы  $\zeta^j$ , см. лемму 2.1. На рис. 6.1 изображена схема ДПФ порядка 8.

Для выполнения как скалярных, так и нескалярных умножений используем рекурсивный вызов данного алгоритма умножения с последующим округлением до  $s$  разрядов после запятой. При этом комплексное умножение выполняется схемой из двух слоев: на первом — четыре вещественных умножения, на втором — вещественные сложение и вычитание.

<sup>2)</sup>Здесь для аргументов ДПФ используется полиномиальная нотация. Напомним также, что символ  $\odot$  обозначает покомпонентное произведение векторов.

Рис. 6.1: Схема ДПФ порядка 8 с примитивным корнем  $\zeta$ 

Заметим, что если (комплексные) коэффициенты вектора-аргумента ДПФ порядка  $2^k$  ограничены по модулю величиной  $m$ , то коэффициенты промежуточных векторов в процессе вычисления ДПФ заведомо ограничены величиной  $2^k m$ . Следовательно, при вычислении произведения  $C(x)$  по формуле (6.4) не встретится чисел, по модулю превосходящих  $M = 2^{3k+2q}$  (поскольку  $m < 2^q$ ). Таким образом, для записи вещественных частей комплексных чисел в процессе вычисления достаточно использовать  $3k + 2q + 1$  разрядов перед запятой<sup>3)</sup>. Число  $s$  разрядов после запятой определим, исходя из оценки величины ошибки.

Пусть  $\varepsilon_x$  означает ошибку вычисления (вещественной) величины  $x$ . Оценим грубо эволюцию абсолютной величины ошибки  $\varepsilon$  от слоя к слою. Ввиду  $(a + \varepsilon_a) \pm (b + \varepsilon_b) = (a \pm b) + (\varepsilon_a \pm \varepsilon_b)$ , на слоях аддитивных операций оценка величины ошибки удваивается,  $\varepsilon \rightarrow 2\varepsilon$ . В случае умножения имеем

$$\varepsilon_{ab} = (a + \varepsilon_a)(b + \varepsilon_b) - ab + \varepsilon_o = b\varepsilon_a + a\varepsilon_b + \varepsilon_a\varepsilon_b + \varepsilon_o,$$

где  $\varepsilon_o$  — ошибка, возникающая при округлении,  $|\varepsilon_o| \leq 2^{-s}$ . Поэтому в случае нескалярного умножения примем  $\varepsilon \rightarrow 3M\varepsilon + 2^{-s}$ , а в случае умножения на корни из единицы<sup>4)</sup> ( $|b| < 1$  и  $\varepsilon_b \leq 2^{-s}$ ) положим  $\varepsilon \rightarrow \varepsilon + 2M2^{-s}$ .

Поэтому при вычислении ДПФ порядка  $2^k$  оценка ошибки изменяется с  $\varepsilon$  на

$$4(2M2^{-s} + 4(2M2^{-s} + \dots + 4(2M2^{-s} + 2\varepsilon) \dots)) < M2^{2k-s} + 2^{2k}\varepsilon.$$

Окончательно, ошибку вычислений по формуле (6.4) оцениваем как

$$0 \xrightarrow{\text{ДПФ}} M2^{2k-s} \xrightarrow{\odot} 3M^22^{2k-s} + 2^{-s} < M^22^{2k+2-s} \xrightarrow{\text{ДПФ}} M^22^{4k+2-s} + M2^{2k-s} < M^22^{4k+3-s}.$$

<sup>3)</sup>С учетом того, что мы не допускаем погрешности более 1/2.

<sup>4)</sup>Это вычисленные заранее константы.

При выборе  $s = 10k + 4q + 4$  ошибка строго меньше  $1/2$ , что и требуется.

При  $q \asymp \log n$  и  $l = s + 3k + 2q + 1$  получаем рекуррентное соотношение

$$\mathsf{C}(M_n) \leq O(2^k k)(\mathsf{C}(M_l) + l),$$

которое, если через  $d$  шагов воспользоваться стандартным методом умножения, приводит к оценке в утверждении теоремы. ■

- Такая же оценка сложности (и, вероятно, тем же самым методом) была ранее получена А. А. Карапубой [19], но не опубликована.

Последующие более быстрые методы умножения отличаются прежде всего выбором кольца для выполнения ДПФ. Второй метод Шёнхаге—Штрассена [305] позволил достичь оценки  $\mathsf{C}(M_n) \leq n \log n \log \log n$ , которая оставалась рекордной на протяжении 35 лет. В нем используется сведение к умножению в кольце многочленов  $\mathbb{Z}_{\Phi_m}[x]/(x^{2^m+1}-1)$ , где  $\Phi_m = 2^{2^m} + 1$ . ДПФ выполняется на кольцом  $\mathbb{Z}_{\Phi_m}$ , которое отличается простотой умножений на корни из единицы — ими являются просто степени двойки.

Метод М. Фюрера [194] совмещает достоинства обоих упомянутых методов: логарифмическую скорость снижения размерности и простоту умножений на корни из единицы. Он переносит умножение в кольцо  $C_p[y]/(y^{2^{ps}} - 1)$ , где  $C_p = \mathbb{C}[x]/(x^{2^p} + 1)$ . Для умножения используется ДПФ порядка  $2^{s(p+1)}$ . Если (методом леммы 2.1) представить его в виде композиции ДПФ порядка  $2^{p+1}$ , то большая часть скалярных умножений будет выполняться на степени переменной  $x$ , которая является примитивным корнем<sup>5)</sup> степени  $2^{p+1}$  в  $C_p$ . Метод приводит к оценке  $\mathsf{C}(M_n) \leq n \log n \cdot 2^{O(\log^* n)}$ .

В методе Д. Харви и Ж. ван дер Хувена [211] целочисленное умножение сводится к умножению в кольце многочленов нескольких переменных  $C_p[x_1, \dots, x_d]/(x_1^{n_1} - 1, \dots, x_d^{n_d} - 1)$ . Для умножения в этом кольце используется многомерное ДПФ, которое, впрочем, легко сводится к обычным ДПФ. Переход в многомерное пространство радикально решает проблему простоты примитивных корней из единицы (достаточно ограничиться мономами переменных  $x_i$ ), но приносит ряд технических трудностей, преодоление которых позволило авторам [211] получить рекордный результат  $\mathsf{C}(M_n) \leq n \log n$ .

Аналогичная теория построена для умножения многочленов над произвольным кольцом  $R$ . Предложенный в [212] метод имеет оценку сложности  $O(n \log n)$  в случае справедливости недоказанной гипотезы. Безусловно доказанная оценка  $\mathsf{C}_{\mathcal{A}^R}(M_n^R) \leq n \log n \cdot 2^{O(\log^* n)}$  получена ранее теми же авторами вместе с Г. Лесером в [213].

Для более подробного ознакомления с теорией методов быстрого умножения рекомендуются обзорные работы Д. Бернштейна [136, 139] и современный обзор С. Б. Гашкова и автора в [13].

## Параллельные схемы деления чисел $[A][\varepsilon]$

Как несложно проверить, изложенный выше (теорема 5.1) метод Кука деления чисел [172] приводит к схемам глубины порядка  $\log^2 n$ . Важной вехой на пути развития быстрых параллельных алгоритмов стала работа П. Бима, С. Кука, Г. Гувера [132], в которой авторы построили схемы логарифмической глубины для деления, а также для близких задач. Несколько более простой и эффективный метод предложили Й. Хостад и Т. Лейтон [216]. Эти методы основаны на переходе к модулярной арифметике — популярнейшем алгебраическом приеме. Как объяснялось ранее, достаточно продемонстрировать возможность параллельного выполнения инвертирования.

<sup>5)</sup>Здесь мы допускаем обычную вольность речи, называя элементами фактор-кольца  $C_p$  не классы эквивалентных многочленов по модулю  $x^{2^p} + 1$ , а представителей классов.

**Теорема 6.3** ([132]).  $D(I_n) \asymp \log n$ .

► Без ограничения общности можем полагать, что входом схемы является  $n$ -разрядное число  $a \in [1/2, 1]$ . Пусть также  $\log_2 n \in \mathbb{N}$  и  $n \geq 16$ . Положим  $z = 1 - a$ . Тогда

$$a^{-1} = \sum_{k=0}^{\infty} z^k \approx \sum_{k=0}^{2n-1} z^k = \prod_{i=0}^{\log_2 n} (1 + z^{2^i}), \quad (6.5)$$

где погрешность приближения не превосходит  $2^{1-2n}$  ввиду  $z \leq 1/2$ . Если каждый из множителей в правой части (6.5) вычислен с абсолютной ошибкой  $\leq \varepsilon$ , то погрешность вычисления произведения (6.5) по абсолютной величине не превосходит

$$\log_2 n \cdot \varepsilon \cdot \sum_{k=0}^{2n-1} z^k \leq 2n \log_2 n \cdot \varepsilon \leq 2^{1-n} \varepsilon. \quad (6.6)$$

I. Параллельно вычислим все степени  $z^{2^i}$  с точностью  $2^{-2n}$  каждую,  $i \leq \log_2 n$ .

Для этого воспользуемся модулярной арифметикой (это центральная часть метода). Выберем взаимно простые числа  $p_1, \dots, p_s$  с условием  $P = p_1 \cdots p_s \geq 2^n$  так, что  $p_i \leq 2n^2$  и  $s \leq 2n^2$ . Это позволяет сделать закон распределения простых чисел (см., например, [294]).

По условию,  $2^n z$  —  $n$ -разрядное целое число. Выполним возведение его в степень  $2^i$  в кольце  $\mathbb{Z}_{p_1} \times \dots \times \mathbb{Z}_{p_s}$  и затем восстановим результат  $(2^n z)^{2^i} < 2^{n^2}$  при помощи Китайской теоремы об остатках.

Процедура выполняется в три этапа. Сначала вычисляются остатки  $a_j = 2^n z \bmod p_j$ , затем — степени  $a_j^{2^i} \bmod p_j$ , и наконец — искомое число  $(2^n z)^{2^i} \bmod P$ . Глубина выполнения первого и третьего этапов оценивается в следующих двух леммах. Напомним, что  $D(M_n) \asymp \log n$  и  $D(\Sigma_{m,n}) \asymp \log(mn)$  (см., например, следствия 4.1 и 4.2), а  $D(\mathcal{P}_n) \asymp n$  (см. далее (11.5) и следствие 12.1).

**Лемма 6.2.** *Остаток от деления  $n$ -разрядного числа  $A$  на  $m$ -разрядную константу  $r$  можно вычислить с глубиной  $O(m + \log n)$ .*

► Разобьем делимое на блоки длины  $m$ :  $A = \sum_{k=0}^{n/m} a_k 2^{km}$ . Вычислим  $A \bmod p$  по формуле

$$A \bmod p = \sum_{k=0}^{n/m} a_k (2^{km} \bmod p) \bmod p.$$

Поскольку множители  $2^{km} \bmod p$  можно считать предвычисленными константами, глубина оценивается как  $D(M_m) + D(\Sigma_{2m, n/m}) + D(\mathcal{P}_{2m + \log_2 n}) \leq m + \log n$ , поскольку сумма произведений имеет величину  $\leq (n/m)2^{2m}$  (операцию внешнего приведения по модулю мы здесь рассматриваем как булев оператор общего вида).  $\square$

**Лемма 6.3.** *Восстановление числа  $A \in [0, P)$  по остаткам от деления на  $m$ -разрядные взаимно простые константы  $p_1, \dots, p_s$ , где  $P = p_1 \cdots p_s$ , можно выполнить с глубиной  $O(\log(ms))$ .*

▷ Пусть  $a_i = A \bmod p_i$ . Тогда  $A = \sum_{i=1}^s u_i a_i \bmod P$ , где  $u_i < P/p_i$  — подходящие константы. Сумма произведений вычисляется с глубиной, не превосходящей  $D(M_{ms}) + D(\Sigma_{ms,s}) \leq \log(ms)$ . Заключительное приведение по модулю  $P$  можно выполнить по формуле  $X \bmod P = X - \lfloor X \cdot (1/P) \rfloor \cdot P$  с глубиной порядка  $D(M_{ms}) \asymp \log(ms)$  (константа  $1/P$  вычислена заранее с необходимой точностью).  $\square$

Как следствие, для глубины приближенного вычисления всех степеней  $z^{2^i}$  имеем оценку  $O(\log n)$ , т. к. операцию возведения в степень в  $\mathbb{Z}_{p_i}$  на втором этапе можно рассматривать как булев оператор общего вида.

*II.* Вычислим финальное произведение в (6.5), опять-таки при помощи модульной арифметики.

Для каждого из найденных приближенных значений  $\tilde{z}_i \approx z^{2^i}$  найдем остатки от деления целого числа  $b_i = \lfloor 2^{2n}(1 + \tilde{z}_i) \rfloor$  на  $p_j$  (лемма 6.2). Произведения  $\prod_{i=0}^{\log_2 n} b_i$  в  $\mathbb{Z}_{p_j}$  (т.е.  $O(\log n)$  штук  $O(\log n)$ -разрядных чисел) вычисляются в бинарном дереве из обычных числовых умножений, в конце результат приводится по модулю  $p_j$ . Окончательно искомое произведение  $\prod_i 2^{2n}(1 + \tilde{z}_i)$  восстанавливается при помощи леммы 6.3 (точно, поскольку  $2^{(2n+1)(\log_2 n+1)} \leq 2^{n^2} < P$  при  $n \geq 16$ ). Восстанавливая правильную позицию запятой, получаем приближение к  $a^{-1}$  с ошибкой  $\leq 2^{-n}$  в силу (6.6), поскольку множители в произведении заданы с точностью  $\varepsilon < 2^{1-2n}$  (складывается из ошибки в  $\tilde{z}_i$  и округления при переходе к  $b_i$ ). Глубина этапа имеет величину порядка  $\log n$ .  $\blacksquare$

- Отметим, что и оценки, и используемые в леммах 6.2, 6.3 конструкции далеко не оптимальны. Сложность своих схем деления логарифмической глубины Бим, Кук и Гувер [132] оценили в  $n^{4+o(1)}$ . Хостад и Лейтон [216] построили семейство схем сложности  $n^{1+\varepsilon}$  и глубины  $\varepsilon^{-2} \log n$ . В развитие этого результата Дж. Рейф и С. Тейт [292] показали, что деление возможно с глубиной порядка  $\log n \log \log n$  и оптимальной сложностью порядка  $M_{\log}(n) = O(n \log n)$  (с учетом результата [211]), где функционал  $M_{\log}(n)$  определяется так же, как  $M(n)$ , но с использованием алгоритма умножения логарифмической глубины.

### Модульная композиция многочленов $[A]/_2$

Оператор  $MC_n^R$  модульной композиции двух многочленов  $f, g \in R[x]$  степени  $< n$  по модулю третьего многочлена  $h$  степени  $n$  определяется как  $MC_n^R(f, g, h) = f(g) \bmod h$ . Операция композиции играет важную роль в арифметике конечных полей, в частности, в задаче факторизации многочленов над конечными полями (см., например, [234]).

Первый алгоритм субквадратичной сложности для модульной композиции построили Р. Брент и Ш. Кунг [159]<sup>6</sup>. В нем применяется сведение к умножению прямоугольных матриц. Пусть  $rs \geq n$ . Запишем  $f(x) = f_1(x) + x^r f_2(x) + \dots + x^{(s-1)r} f_s(x)$ , где  $\deg f_i < r$ . При помощи  $r+s-2$  последовательных умножений и делений с остатком вычисляются многочлены  $g_i = g^i \bmod h$ ,

<sup>6</sup>Более точно, в работе [159] рассматривался частный случай задачи с  $h(x) = x^n$ , но результат легко переносится на общий случай.

$i = 1, 2, \dots, r, 2r, \dots, (s-1)r$ . Далее композиции  $\varphi_j = f_j(g) \bmod h$ ,  $j = 1, \dots, s$ , вычисляются путем подстановки многочленов  $g_i$  вместо степеней  $x^i$ . Справедливо

$$\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \dots \\ \varphi_s \end{bmatrix}_{s \times n} = \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_s \end{bmatrix}_{s \times r} \cdot \begin{bmatrix} g_0 \\ g_1 \\ \dots \\ g_{r-1} \end{bmatrix}_{r \times n},$$

где в строках матриц выписаны коэффициенты соответствующих многочленов. Окончательно  $f(g) \equiv \varphi_1 + g_r \varphi_2 + \dots + g_{(s-1)r} \varphi_s \bmod h$ . При выборе  $r \sim s \sim \sqrt{n}$  метод имеет сложность  $C(MC_n) \prec C(MM_{r,s,n}) + (r+s) \cdot (C(M_n) + C(QR_{n,n})) \prec n^{1.63}$ , если подставить рекордную оценку сложности умножения прямоугольных матриц из работы [121].



Кристофер Уманс  
Калифорнийский  
технологический институт,  
с 2002

Для этого при  $n \leq d^m$ ,  $i = 0, \dots, m-1$ , при которой многочлен  $f(x)$  превращается в  $f^*(x_1, \dots, x_{m-1})$ . Многочлены  $g_i(x) = g^{d^i} \bmod h$  вычисляются стандартным образом (последовательными возвведениями в степень  $d$ ) с общей сложностью порядка  $m \log d \cdot M(n)$ .

Теперь степень многочлена  $\hat{f}(x) = f^*(g_0(x), \dots, g_{m-1}(x))$  не превосходит  $N = dmn$ , и он может быть вычислен при помощи интерполяции в  $N$  точках. Чтобы обеспечить запас точек интерполяции, перейдем от  $\mathbb{F}_p$  к  $\mathbb{F}_q$ , где  $q = p^t > N$ .

Сначала рассмотрим вспомогательные задачи вычисления значений многочлена одной переменной на наборе точек и обратную задачу интерполяции. Метод деления пополам приводит к следующему результату, вероятно впервые полученному Ш. Кунгом [249].

**Лемма 6.4 ([249]).** Схемами над  $\mathcal{A}^R$  сложности  $\prec M(n) \log n$  реализуются:

- (i) вычисление значений многочлена  $f(x) \in R[x]$  степени  $< n$  в  $n$  точках  $\alpha_0, \dots, \alpha_{n-1} \in R$ ;
- (ii) восстановление многочлена  $f(x) \in R[x]$  степени  $< n$  по заданным значениям  $f(\alpha_i) = c_i$  в точках  $\alpha_i$ ,  $i = 0, \dots, n-1$ .

▷ Рассмотрим сбалансированное бинарное дерево  $T$  с  $n$  листьями, ориентированное к корню. Листья пронумерованы числами от 0 до  $n-1$  — они соответствуют индексам точек интерполяции  $\alpha_i$ . Далее каждой вершине  $v$  сопоставляется

множество чисел  $I(v) = I(v') \cup I(v'')$ , где  $v', v''$  — вершины, предшествующие  $v$ . Корню дерева соответствует множество  $\llbracket n \rrbracket$ .

Положим  $p_I(x) = \prod_{i \in I} (x - \alpha_i)$ . Поскольку  $f(\alpha) = f \bmod (x - \alpha)$ , все значения  $f(\alpha_1), \dots, f(\alpha_n)$  можно найти, двигаясь по дереву  $T$  от корня и вычисляя в каждой вершине  $v$  многочлены  $f \bmod p_{I(v)}$ . При корне дерева имеем просто  $f \bmod p_{\llbracket n \rrbracket} = f$ , при листьях получим  $f(\alpha_i)$ .

Сложность вычисления вспомогательных многочленов  $p_{I(v)}$  оценивается как

$$\mathsf{C}(M_{n/2+1}) + 2\mathsf{C}(M_{n/4+1}) + \dots + (n/2)\mathsf{C}(M_2) \preccurlyeq \mathsf{M}(n) \log n.$$

Сложность делений с остатком, дающих все  $f \bmod p_{I(v)}$ , оценивается как

$$2\mathsf{C}(QR_{n,n/2+1}) + 4\mathsf{C}(QR_{n/2,n/4+1}) + \dots + n\mathsf{C}(QR_{3,2}) \preccurlyeq \mathsf{M}(n) \log n.$$

Это доказывает п. (i).

Перейдем к задаче интерполяции. Обозначим  $p_{I,k}^*(x) = \prod_{i \in I \setminus \{k\}} (x - \alpha_i)$ . Вычисления проведем по интерполяционной формуле Лагранжа

$$f(x) = \sum_{k=1}^n c_k^* \cdot p_{\llbracket n \rrbracket, k}^*(x), \quad c_k^* = c_k / p_{\llbracket n \rrbracket, k}^*(\alpha_k).$$

Сначала вычислим значения  $p_{\llbracket n \rrbracket, k}^*(\alpha_k)$ . Заметим, что  $p_{\llbracket n \rrbracket, k}^*(\alpha_k) = p'_{\llbracket n \rrbracket}(\alpha_k)$ , где  $p'_{\llbracket n \rrbracket}$  — производная многочлена  $p_{\llbracket n \rrbracket}$ . Многочлен  $p'_{\llbracket n \rrbracket}$  вычисляется с линейной сложностью, если известен  $p_{\llbracket n \rrbracket}$ . Тогда, согласно п. (i), значения многочлена  $p'_{\llbracket n \rrbracket}$  во всех точках  $\alpha_k$  вычисляются со сложностью  $\preccurlyeq \mathsf{M}(n) \log n$ . Еще  $n$  операций деления позволяют найти все  $c_k^*$ .

Обозначим  $f_I(x) = \sum_{k \in I} c_k^* \cdot p_{I,k}^*(x)$ . Продвигаясь по дереву в направлении от листьев к корню, последовательно вычислим все многочлены  $f_{I(v)}$  по формулам  $f_{I(v)} = f_{I(v')} p_{I(v'')} + f_{I(v'')} p_{I(v')}$ , где  $v'$  и  $v''$  предшествуют  $v$ . В листьях многочлены  $f_{I(v)}$  совпадают с константами  $c_j^*$ . В корне получаем искомый многочлен  $f(x)$ .

Все вспомогательные многочлены  $p_{I(v)}$  вычисляются со сложностью  $\preccurlyeq \mathsf{M}(n) \log n$ . После этого сложность вычисления всех  $f_{I(v)}$  оценивается как

$$2\mathsf{C}(M_{n/2+1}) + 4\mathsf{C}(M_{n/4+1}) + \dots + n\mathsf{C}(M_2) \preccurlyeq \mathsf{M}(n) \log n.$$

□

Согласно лемме 6.4, значения многочленов  $g_i(x)$  в точках интерполяции определяются со сложностью  $\preccurlyeq \underline{m\mathsf{M}(N) \log N}$  операций в  $\mathbb{F}_q$ . Остается решить задачу вычисления значений многочлена  $f^*$  от  $m$  переменных на наборе из  $N$  векторов  $\alpha_i = (\alpha_{i,0}, \dots, \alpha_{i,m-1}) \in \mathbb{F}_q^m$ .

Пусть  $\mathbb{F}_{q_0} \cong \mathbb{F}_p(\beta)$ , где  $q_0 = p^s \geq dm^2$ ,  $s \mid t$  и  $\beta$  — примитивный элемент поля  $\mathbb{F}_{q_0} \subset \mathbb{F}_q$ . Вычислим многочлены  $\varphi_1(x), \dots, \varphi_N(x) \in \mathbb{F}_q[x]$  степени  $< m$  по заданным значениям в точках  $\beta^j$ :

$$\varphi_i(\beta^j) = (\alpha_{i,j})^{q_0^{-j}} = (\alpha_{i,j})^{p^{t-sj}}, \quad j = 0, \dots, m-1.$$

Сложность вычисления степеней  $\alpha_{i,j}$  по порядку не превосходит  $\preccurlyeq Nm \log q$ , а интерполяция согласно лемме 6.4 выполняется со сложностью  $\preccurlyeq \underline{N\mathsf{M}(m) \log m}$  операций в  $\mathbb{F}_q$ .

Рассмотрим многочлен  $F(y) = f^*(y, y^{q_0}, \dots, y^{q_0^{m-1}})$  — его ненулевые коэффициенты совпадают с коэффициентами многочлена  $f(x)$ , только относятся к другим степеням, при этом  $\deg F < q_0^m$ . Полагая формально  $F \in S[y]$ , где  $S = \mathbb{F}_q[x]/(x^{q_0-1} - \beta)$ , найдем значения  $F(y)$  в  $N$  точках  $\varphi_i(x) \in S$  методом леммы 6.4 при помощи  $\mathbf{M}(Q) \log Q$  операций в  $S$ , где  $Q = \max\{N, q_0^m\}$ .

Определим степени Фробениуса многочленов  $\varphi_i(x) = \sum_{l=0}^{m-1} a_l x^l$  как  $\varphi_i^{[j]} = \sum_{l=0}^{m-1} a_l^{q_0^j} x^l$ . Теперь покажем, что  $f^*(\alpha_i) = F(\varphi_i(x))|_{x=1}$ . Действительно,

$$\begin{aligned} F(\varphi_i(x)) &= f^*\left(\varphi_i(x), \varphi_i^{q_0}(x), \dots, \varphi_i^{q_0^{m-1}}(x)\right) = \\ &f^*\left(\varphi_i(x), \varphi_i^{[1]}(x^{q_0}), \dots, \varphi_i^{[m-1]}(x^{q_0^{m-1}})\right) \equiv \\ &f^*\left(\varphi_i(x), \varphi_i^{[1]}(\beta x), \dots, \varphi_i^{[m-1]}(\beta^{m-1} x)\right) \mod x^{q_0-1} - \beta, \end{aligned}$$

причем степень последнего многочлена не превосходит  $(d-1)m^2 < q-1$ , т.е. это в точности результат приведения по модулю  $x^{q_0-1} - \beta$ . Остается заметить, что (после подстановки  $x = 1$ )  $\varphi_i^{[j]}(\beta^j) = (\varphi_i(\beta^j))^{q_0^j} = \alpha_{i,j}$  ввиду  $\beta \in \mathbb{F}_{q_0}$ . Сложность подстановок не превосходит  $q_0 N$  операций в  $\mathbb{F}_q$ .

Теперь многочлен  $\hat{f}(x)$  восстанавливается методом леммы 6.4 при помощи  $\mathbf{M}(N) \log N$  операций в  $\mathbb{F}_q$ . Заключительное приведение  $\hat{f}$  по модулю  $h$  выполняется со сложностью  $\mathbf{C}(QR_{N,n}) \preccurlyeq \mathbf{M}(N) \log N \cdot \mathbf{M}(n)$  над  $\mathbb{F}_p$ .

Сложность метода оценивается суммой оценок сложности отдельных шагов, которые подчеркнуты по ходу доказательства. Окончательно получаем

$$\begin{aligned} \mathbf{C}_{\mathcal{A}^{\mathbb{F}_p}}(MC_n) &\preccurlyeq m \log d \cdot \mathbf{M}(n) + dm \log n \cdot \mathbf{M}(n) + \mathbf{M}(Q) \log Q \cdot M(q_0)M(t) + \\ &[m\mathbf{M}(N) \log N + Nm \log q + N\mathbf{M}(m) \log m + q_0 N + \mathbf{M}(N) \log N] M(t) \preccurlyeq \\ &(Nt)^{1+o(1)}(q_0 + \log q) + (q_0^{m+1}t)^{1+o(1)}. \end{aligned}$$

С учетом  $N = dm n$  и возможности выбрать  $d^m \leq dn$ ,  $q_0 \leq pdm^2$  и  $q \leq q_0 dm n$  (при этом  $t \preccurlyeq \log(dm n)$ ), получаем

$$\mathbf{C}_{\mathcal{A}^{\mathbb{F}_p}}(MC_n) \preccurlyeq p(dm)^{O(1)} n^{1+o(1)} + (d^2(pdm^2)^{m+1} n)^{1+o(1)},$$

откуда следует заявленная оценка. ■

При  $p = n^{o(1)}$  можно выбрать  $1 \prec m \prec \frac{\log n}{\log \log n}$  так, что  $p^m = n^{o(1)}$  и  $d = n^{o(1)}$ , поэтому  $\mathbf{C}_{\mathcal{A}^{\mathbb{F}_p}}(MC_n) = n^{1+o(1)}$ .

- Чуть более тонкое рассуждение позволяет получить примерно такую же оценку сложности, как в теореме 6.4, только с  $\text{char } \mathbb{F}_p$  вместо  $p$  [324].

Уточненные оценки сложности алгоритмов леммы 6.4, полученные А. Бостаном и Э. Шостром в [152], имеют вид  $\lesssim 1.5\mathbf{M}(n) \log_2 n$  для задачи (i) вычисления значений в  $n$  точках и  $\lesssim 2.5\mathbf{M}(n) \log_2 n$  для задачи (ii) интерполяции по значениям в точках при  $n = 2^k$ .

Альтернативный алгоритм для выполнения центрального этапа модульярной композиции — вычисления значений многочлена нескольких переменных — предложен в [142].

Ж. ван дер Хувен и Г. Лесер [223] путем модификации метода К. Уманса и К. Кедлай [238]<sup>7</sup>) распространяли описанный алгоритм на поля произвольного порядка и кольца вычетов, получив для *битовой* сложности оценку  $C_{B_2}(MC_n^R) = (n \log q)^{1+o(1)}$  при  $R = \mathbb{F}_q$  или  $R = \mathbb{Z}_q$ .

## Другие приложения

**Умножение в полях Мерсенна.** Рассмотрим задачу умножения по модулю простого числа Мерсенна  $2^p - 1$ , т.е. фактически умножения в простом поле  $\mathbb{Z}_{2^p - 1}$ . Его можно представить как циклическую свертку порядка  $p$  векторов двоичной записи перемножаемых чисел. Поэтому умножение выполняется при помощи ДПФ порядка  $p$  с примитивным корнем  $2 \in \mathbb{Z}_{2^p - 1}$ . Но  $p$  — тоже простое число, и ДПФ такого порядка, как правило, реализуется не очень эффективно. Однако, в указанной ситуации можно воспользоваться приемом, предложенным Р. Крэндаллом и Б. Фейгином [179], и позволяющим в некоторых случаях свести ДПФ «неудобного» порядка к ДПФ «удобного» порядка.

Прием состоит в переходе к приближенному вычислению вещественного ДПФ произвольного порядка  $N$ . Разобьем перемножаемое число  $X = [x_{p-1}, \dots, x_0]$  на  $N$  блоков примерно одинаковой длины:  $X = [X_{N-1}, \dots, X_0]$ . Пусть  $B_i$  — позиция начала  $i$ -го блока. Запишем  $X$  в виде:

$$X = \sum_{i=0}^{N-1} X_i \cdot 2^{B_i} = \sum_{i=0}^{N-1} \left( X_i \cdot 2^{B_i - ip/N} \right) 2^{ip/N} = \sum_{i=0}^{N-1} X'_i \cdot 2^{ip/N}.$$

Теперь умножение  $X$  на  $Y = \sum_{i=0}^{N-1} Y'_i \cdot 2^{ip/N}$  можно выполнить при помощи двух ДПФ порядка  $N$  с примитивным корнем  $2^{p/N} \in (\mathbb{R} \bmod 2^p - 1)$  и одного обратного ДПФ. Из вектора результата  $[Z'_{N-1}, \dots, Z'_0]$  искомое произведение находится как

$$XY = \sum_{i=0}^{N-1} \left( Z'_i \cdot 2^{ip/N - B_i} \right) 2^{B_i} \bmod 2^p - 1.$$

При выборе размера блоков близким к длине машинного слова описанный метод эффективно реализуется на стандартных компьютерах.

**Арифметика в нормальных базисах конечных полей.** Конечное поле порядка  $q^n$  изоморфно фактор-кольцу многочленов  $\mathbb{F}_q[x]/(m_n(x))$  по модулю неприводимого над  $\mathbb{F}_q$  многочлена  $m_n(x)$  степени  $n$ . Таким образом, операции в конечном поле по существу есть операции с многочленами. Иногда, однако, могут быть полезны альтернативные представления.

В общем случае, элементы поля  $\mathbb{F}_{q^n}$  представляются линейными комбинациями базисных элементов  $\alpha_0, \dots, \alpha_{n-1}$  над  $\mathbb{F}_q$ . В стандартном (полиномиальном) представлении  $\alpha_i = \alpha^i$  (здесь  $\alpha$  — генератор базиса, корень неприводимого многочлена). Из прочих представлений наиболее популярным является нормальное. Его базисные элементы имеют вид  $\alpha_i = \beta^{q^i}$ , где  $\beta$  — порождающий элемент базиса (нормальный элемент поля).

Идея о выполнении переходов между стандартными и нормальными базисами для ускорения вычислений, по-видимому, впервые высказана Э. Калтофеном и В. Шаупом в [234]. Они же построили схему субквадратичной сложности  $O(n^{1.82})$  (в операциях поля  $\mathbb{F}_q$ ) для перехода от нормального к стандартному представлению. Аналогичная схема для перехода в обратную сторону построена автором в [81]<sup>8</sup>). Алгоритмы перехода подобны алгоритму модулярной композиции Брента—Кунга [159].

Рассмотрим задачу построения схем над полным арифметическим базисом  $\mathcal{A}_{\mathbb{F}_q}$  для перехода между двумя представлениями произвольного элемента  $y \in \mathbb{F}_{q^n}$ :

$$y = a_0 + a_1\beta + \dots + a_{n-1}\beta^{n-1} = b_0\beta + b_1\beta^q + \dots + b_{n-1}\beta^{q^{n-1}}.$$

<sup>7</sup>Метод [238] адаптирован для модели RAM-программ.

<sup>8</sup>В работе [234] переход к нормальному представлению выполняется вероятностным алгоритмом.

Пусть  $n \leq ms$ . Если известно нормальное представление (коэффициенты  $b_i$ ), запишем

$$y = \gamma_0 + \gamma_1^{q^m} + \dots + \gamma_{s-1}^{q^{m(s-1)}}, \quad \text{где } \gamma_k = b_{mk}\beta + b_{mk+1}\beta^q + \dots + b_{mk+m-1}\beta^{q^{m-1}}.$$

Тогда

$$\begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \dots \\ \gamma_{s-1} \end{bmatrix}_{s \times n}^A = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \dots \\ \gamma_{s-1} \end{bmatrix}_{s \times m}^B \cdot \begin{bmatrix} \beta \\ \beta^q \\ \dots \\ \beta^{q^{m-1}} \end{bmatrix}_{m \times n}^A,$$

где в строках матриц с индексом  $A$  записаны коэффициенты стандартных представлений элементов, а в матрице с индексом  $B$  — коэффициенты нормального представления при  $\beta, \beta^q, \dots, \beta^{q^{m-1}}$ . Далее, степени элементов  $\gamma_i$  в стандартном базисе вычисляются при помощи модулярной композиции:

$$f(x)^{q^k} \bmod m_n(x) = f(x^{q^k}) \bmod m_n(x) = f(\xi(x)) \bmod m_n(x), \quad \xi(x) = x^{q^k} \bmod m_n(x).$$

Остается найти сумму  $\sum \gamma_k$ . Сложность указанного метода (изложена модификация [234] из [81]) при малых  $q$  и выборе  $s \approx m \approx \sqrt{n}$  оценивается как  $C(MM_{s,m,n}) + sC(MC_n) + sn \prec n^{1.63}$  при подстановке известных оценок сложности модулярной композиции [324] (см. теорему 6.4) и умножения прямоугольных матриц [121], когда характеристика поля не слишком велика.

Метод [81] перехода в обратную сторону опирается на следующее наблюдение: нормальные представления элементов  $y$  и  $y^{q^k}$  отличаются циклическим сдвигом на  $k$  позиций. Сначала вычисляем степени  $y^{q^m}, y^{q^{2m}}, \dots, y^{q^{(s-1)m}}$ . Затем из известных частичных нормальных представлений для  $1, x, \dots, x^{n-1}$  находим частичные представления элементов  $y^{q^m}, y^{q^{2m}}, \dots, y^{q^{(s-1)m}}$ :

$$\begin{bmatrix} y \\ y^{q^m} \\ \dots \\ y^{q^{(s-1)m}} \end{bmatrix}_{s \times m}^B = \begin{bmatrix} y \\ y^{q^m} \\ \dots \\ y^{q^{(s-1)m}} \end{bmatrix}_{s \times n}^A \cdot \begin{bmatrix} 1 \\ x \\ \dots \\ x^{n-1} \end{bmatrix}_{n \times m}^B.$$

Их совокупность в силу приведенного выше замечания дает все нормальные координаты  $y$ . Сложность алгоритма не превосходит  $sC(MC_n) + C(MM_{s,n,m}) \prec n^{1.63}$ .

Для полноты картины отметим, что смена одного стандартного представления на другое выполняется при помощи операции модулярной композиции: если  $y = f(\beta)$  в стандартном базисе с генератором  $\beta$ , то  $y = f(\xi(x)) \bmod m_n(x)$  в базисе с генератором  $\alpha$  — корнем многочлена  $m_n(x)$ , где  $\beta = \xi(\alpha)$ .

Еще проще выполняется переход между двумя нормальными представлениями: легко проверить, что матрица перехода является циркулянтной<sup>9</sup>), поэтому сложность операции не превосходит  $C(M_n^{\mathbb{F}_q})$  [81].

Ясно, что при выполнении многих арифметических операций, скажем, умножения или деления, в нормальном базисе выгоден переход к стандартному базису. Обратный пример, когда целесообразно перейти из стандартного базиса в нормальный, по всей видимости, доставляет оператор вычисления всех автоморфизмов Фробениуса:  $y \rightarrow (y^q, y^{q^2}, \dots, y^{q^n})$ . В нормальном базисе эта операция «бесплатна», а для стандартного базиса в [81] предложен использующий идею перехода к нормальному базису алгоритм сложности порядка  $nC(M_n^{\mathbb{F}_q})$ , что несколько меньше, чем у алгоритма [201], не использующего идеи перехода.

На практике часто используются специальные нормальные базисы с простой таблицей умножения (оптимальные, гауссовые), которые можно найти во многих полях. Но и для них стандартный алгоритм умножения Месси—Омуры [264] имеет квадратичную сложность. В то же время, как показали А. А. Болотов и С. Б. Гашков [6], эти базисы обладают низкой транзитивной сложностью<sup>10</sup>), обычно порядка  $n \log n$ , поэтому сложность умножения в этих базисах

<sup>9</sup>Напомним, что все строки циркулянтной матрицы порождаются циклическими сдвигами одного и того же вектора.

<sup>10</sup>Сложность перехода к стандартному представлению и обратно.

по порядку близка к  $C(M_n^{\mathbb{F}_q})$ . Аналогичный результат, использующий идею перехода неявно, получен чуть раньше в [199].

В работе [203] рассматривается задача о построении (арифметических) схем для перехода между полиномиальными и нормальными базисами в расширениях полей характеристики 0. Предложенные алгоритмы имеют сложность  $O(n^{1.99})$  относительно степени расширения  $n$ , но они вероятностные.

# Глава 7

## Специальная кодировка e

Переход к альтернативной кодировке (представлению) входов, в которой требуемая функция вычисляется проще, особенно популярен в булевых вычислениях. Этот прием по духу близок к идее алгебраического метода, но свободен от структурных алгебраических ограничений.

### Схемная сложность суммирования битов e

Комбинируя стандартные блоки  $FA_3$  сложения трех бит, легко построить схему суммирования  $n$  бит сложности  $5n + O(\log n)$ , см. рис. 7.1.

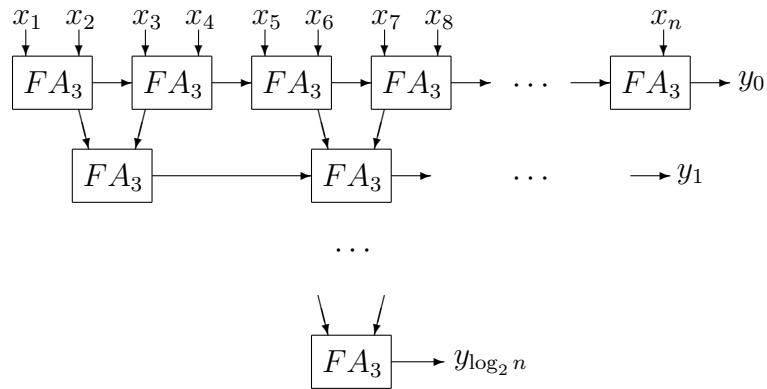


Рис. 7.1: Стандартная схема суммирования  $n$  бит

В свете того, что  $n$ -разрядный сумматор, составленный из блоков  $FA_3$ , оптимален [69], естественно было предположить, что то же верно и для оператора сложения битов. Несколько неожиданно в 2010 г. группа математиков из Санкт-Петербурга [180] обнаружила, что это не так.

**Теорема 7.1** ([180]).  $C_{B_2}(\Sigma_{1,n}) \leq 4.5n + O(\log n)$ .

► Конструкция использует идею перехода от стандартной записи битов к кодированию двух бит  $x, y$  парой  $(x, x \oplus y)$ . Специальный компрессор, обозначаемый  $MDFA$ , выполняет сложение пяти бит по правилу  $x_1 + x_2 + x_3 + x_4 + x_5 = 2(y_1 + y_2) + y_0$  со сложностью 8, если две пары входов даны в измененной кодировке, и в этой же кодировке вычисляется пара выходов  $(y_1, y_2)$ , см. рис 7.2. Теперь замена базового компрессора в исходной схеме рис. 7.1, см. рис. 7.3, сразу приводит к требуемой оценке. ■



Александр Сергеевич  
Куликов  
Санкт-Петербургский  
университет, с 2005

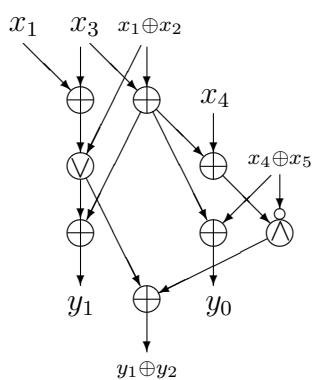


Рис. 7.2: Схема компрессора  $MDFA$

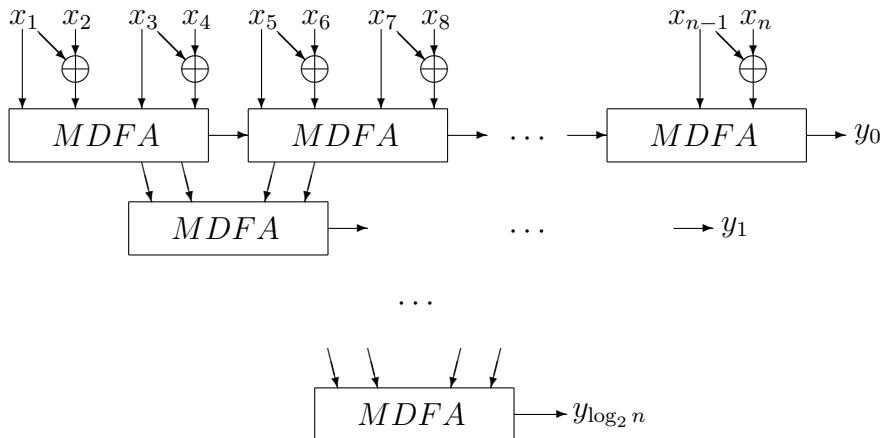


Рис. 7.3: Схема суммирования  $n$  бит из компрессоров  $MDFA$

**Следствие 7.1** ([180]). *Сложность класса  $\mathcal{S}_n$  симметрических булевых функций  $n$  переменных удовлетворяет оценке  $C_{\mathcal{B}_2}(\mathcal{S}_n) \leq 4.5n + o(n)$ .*

- Идея кодировки  $(x, x \oplus y)$  восходит к работе Л. Стокмейера [315], который использовал ее для доказательства асимптотически точной оценки сложности оператора сложения битов по модулю 4:  $C_{\mathcal{B}_2}(\text{MOD}_n^4) = 2.5n \pm O(1)$ .

Компрессор MDFA также позволяет улучшить стандартную оценку сложности сложения трех  $n$ -разрядных чисел. Если учесть, что  $5n - 3$  операций необходимо для сложения двух чисел [69], то было бы естественно предположить, что сложение трех чисел потребует примерно  $10n$  операций. Однако это тоже не так. Организуя компрессоры *MDFA* в цепочку (в том же стиле, что и компрессоры *FA*<sub>3</sub> в стандартной схеме сумматора, см. теорему 1.3), получаем схему сумматора трех чисел с оценкой сложности  $C_{B_2}(\Sigma_{n,3}) < 9n$ .

### Вещественная сложность комплексного ДПФ $\boxed{e}$

Теорема 2.4 оценивает сложность ДПФ порядка  $N = 2^k$  над полем  $\mathbb{C}$  асимптотически как  $1.5N \log_2 N$  арифметических операций, и эта оценка пока не улучшена. С прикладной точки зрения больший интерес представляет оценка сложности, выраженная в вещественных операциях. Напомним, что комплексное сложение или вычитание выполняется за 2 вещественных сложения или вычитания, а умножение на комплексную константу выполняется за 6 вещественных операций (4 скалярных умножения и 2 сложения, либо 3 умножения и 3 сложения-вычитания). Тогда из теоремы 2.4 следует  $C_{A_L^R}(\text{ДПФ}_N[\mathbb{C}]) < 5N \log_2 N$ .

Лучшую оценку дает известный с 1980-х гг. алгоритм ДПФ «с расщепленным основанием» (split-radix FFT), который учитывает, что умножения на корни 4-й степени  $\pm i$  выполняются бесплатно.

**Теорема 7.2** ([334]). *Пусть  $N = 2^k$ . Тогда  $C_{A_L^R}(\text{ДПФ}_N[\mathbb{C}]) \leq 4N \log_2 N$ .*

► Напомним, что компоненты ДПФ порядка  $PQ$  могут быть вычислены по формулам (2.7):

$$x_{pQ+q}^* = \sum_{j=0}^{P-1} (\zeta^Q)^{jp} \cdot \zeta^{jq} \cdot \sum_{i=0}^{Q-1} (\zeta^P)^{iq} x_{iP+j} \quad (7.1)$$

при  $p = 0, \dots, P-1$  и  $q = 0, \dots, Q-1$ .

Полагая  $P = 2^{k-1}$  и  $Q = 2$ , вычислим по формуле (7.1) компоненты с четными индексами, т.е. при  $q = 0$ . При любом  $p = 0, \dots, P-1$  получаем

$$x_{2p}^* = \sum_{j=0}^{P-1} (\zeta^2)^{jp} (x_j + x_{P+j}).$$

Вычисления сводятся к  $P = 2^{k-1}$  сложениям комплексных коэффициентов и выполнению ДПФ порядка  $2^{k-1}$ .

Для вычисления компонент с нечетными индексами положим  $P = 2^{k-2}$ ,  $Q = 4$  и вновь воспользуемся формулой (7.1). При любом  $p = 0, \dots, P-1$  имеем

$$\begin{aligned} x_{4p+1}^* &= \sum_{j=0}^{P-1} (\zeta^4)^{jp} \cdot \zeta^j \cdot (x_j - x_{2P+j} + i(x_{P+j} - x_{3P+j})), \\ x_{4p+3}^* &= \sum_{j=0}^{P-1} (\zeta^4)^{jp} \cdot \zeta^{3j} \cdot (x_j - x_{2P+j} - i(x_{P+j} - x_{3P+j})). \end{aligned}$$

Эти формулы используют  $4P$  комплексных сложений-вычитаний,  $2P$  умножений на степени примитивного корня  $\zeta$  и два ДПФ порядка  $2^{k-2}$ .

Окончательно получаем соотношение

$$C_{\mathcal{A}_L^R}(\text{ДПФ}_N) \leq C_{\mathcal{A}_L^R}(\text{ДПФ}_{N/2}) + 2C_{\mathcal{A}_L^R}(\text{ДПФ}_{N/4}) + 6N,$$

которое разрешается так, как заявлено, с учетом начальных данных  $C_{\mathcal{A}_L^R}(\text{ДПФ}_2) = 4$  и  $C_{\mathcal{A}_L^R}(\text{ДПФ}_1) = 0$ . ■

- Точная оценка сложности метода имеет вид  $4N \log_2 N - 6N + 8$ . Она опубликована Р. Явне в [334], но получена более сложным способом. Метод ДПФ с расщеплением основания был опубликован практически одновременно в [183, 263, 327].

Долгое время оценка теоремы 7.2 казалась неприступной, пока Дж. ван Бускирк не обнаружил, что она все-таки может быть улучшена (опубликовано в [260]). Ключевую роль в методе играет перенормировка входного вектора  $X \rightarrow \sigma \odot X$ , благодаря которой часть скалярных умножений в схеме становятся более простыми. Заметим, что умножение на константы вида  $\pm 1 + a\mathbf{i}$  или  $a \pm \mathbf{i}$  можно выполнить, используя по два вещественных сложения-вычитания и умножения.

Пусть  $\|a\| = \max\{|\Re a|, |\Im a|\}$  означает  $l_\infty$ -норму комплексного числа  $a = \Re a + \Im a \cdot \mathbf{i}$ .

**Теорема 7.3** ([260]). Пусть  $N = 2^k$ . Тогда  $C_{\mathcal{A}_L^R}(\text{ДПФ}_N[\mathbb{C}]) \leq 3\frac{7}{9}N \log_2 N + 2N$ .

- Пусть  $\zeta_N = e^{2\pi\mathbf{i}/N}$  — примитивный корень порядка  $N$  в  $\mathbb{C}$ . При всех  $j \in \mathbb{Z}$  определим вещественные коэффициенты

$$\sigma_{N,j} = \prod_{0 \leq l < k/2} \left\| \zeta_N^{j \cdot 4^l} \right\|.$$

В силу  $\left\| \zeta_N^j \right\| = \left\| \zeta_N^{\pm j \pm N/4} \right\|$  эти коэффициенты обладают свойствами симметрии  $\sigma_{N,j} = \sigma_{N,-j}$  и периодичности  $\sigma_{N,j} = \sigma_{N,j+N/4}$ . Кроме того, по построению число  $(\sigma_{N/4,j}/\sigma_{N,j})\zeta_N^j = \zeta_N^j/\left\| \zeta_N^j \right\|$  имеет вид  $\pm 1 + a\mathbf{i}$  или  $a \pm \mathbf{i}$ .

Через  $\dot{x}_j$  обозначим нормированные коэффициенты входного вектора ДПФ:  $\dot{x}_j = \sigma_{N,j}^{-1}x_j$ . Будем строить схемы для нормированных преобразований

$$\text{ДПФ}'_N(x_0, x_1, \dots, x_{N-1}) = \text{ДПФ}_{N,\zeta_N}[\mathbb{C}](\dot{x}_0, \dot{x}_1, \dots, \dot{x}_{N-1}).$$

Согласно формуле (7.1) с выбором параметров  $P = 2^{k-2}$  и  $Q = 4$  и свойству периодичности коэффициентов  $\sigma_{k,j}$ , для компонент  $\dot{x}_i^*$  преобразования  $\text{ДПФ}'_N$  выполнено (далее  $\zeta = \zeta_N$ ):

$$\dot{x}_{4p+q}^* = \sum_{j=0}^{P-1} (\zeta^4)^{jp} \cdot \zeta^{jq} \cdot \sigma_{N,j}^{-1} \cdot \gamma_{j,q}, \quad \gamma_{j,q} = \sum_{r=0}^3 \mathbf{i}^{rq} x_{rP+j}$$

Внутренние суммы  $\gamma_{j,q}$  образуют компоненты  $P$  ДПФ порядка 4 и вычисляются при помощи 16 вещественных сложений-вычитаний каждое. Дальнейшие вычисления при  $q = 0, 1, 3$  выполняются по формулам

$$\begin{aligned}\dot{x}_{4p}^* &= \sum_{j=0}^{P-1} (\zeta^4)^{jp} \sigma_{N/4,j}^{-1} \cdot (\sigma_{N/4,j}/\sigma_{N,j}) \cdot \gamma_{j,0}, \\ \dot{x}_{4p+1}^* &= \sum_{j=0}^{P-1} (\zeta^4)^{jp} \sigma_{N/4,j}^{-1} \cdot (\sigma_{N/4,j}/\sigma_{N,j}) \zeta^j \cdot \gamma_{j,1}, \\ \dot{x}_{4p+3}^* &= \sum_{j=0}^{P-1} (\zeta^4)^{j(p+1)} \sigma_{N/4,j}^{-1} \cdot (\sigma_{N/4,j}/\sigma_{N,j}) \zeta^{-j} \cdot \gamma_{j,3}.\end{aligned}\quad (7.2)$$

(Обратим внимание на циклический сдвиг вектора коэффициентов внешнего ДПФ в последней сумме — он позволяет от умножений на  $\zeta^{3j}$  перейти к более простым умножениям на нормированные числа  $\zeta^{-j}$ .)

Вычисления по формулам (7.2) состоят в выполнении  $N/4$  умножений на действительные константы,  $N/2$  умножений на константы вида  $\pm 1 + a\mathbf{i}$  или  $a \pm \mathbf{i}$  и трех преобразований типа  $\text{ДПФ}'_{N/4}$ .

Для вычисления оставшихся компонент  $\dot{x}_{4p+2}^*$  используем формулу (7.1) с параметрами  $P = 2^{k-3}$  и  $Q = 8$ :

$$\begin{aligned}\dot{x}_{8p+2}^* &= \sum_{j=0}^{P-1} (\zeta^8)^{jp} \sigma_{N/8,j}^{-1} \cdot (\sigma_{N/8,j}/\sigma_{N/2,j}) (\zeta^2)^j \cdot \alpha_j, \\ \dot{x}_{8p+6}^* &= \sum_{j=0}^{P-1} (\zeta^8)^{j(p+1)} \sigma_{N/8,j}^{-1} \cdot (\sigma_{N/8,j}/\sigma_{N/2,j}) (\zeta^2)^{-j} \cdot \beta_j,\end{aligned}$$

где

$$\begin{aligned}\alpha_j &= (\sigma_{N/2,j}/\sigma_{N,j}) \gamma_{j,2} + \mathbf{i}(\sigma_{N/2,j+N/8}/\sigma_{N,j+N/8}) \gamma_{j+N/8,2}, \\ \beta_j &= (\sigma_{N/2,j}/\sigma_{N,j}) \gamma_{j,2} - \mathbf{i}(\sigma_{N/2,j+N/8}/\sigma_{N,j+N/8}) \gamma_{j+N/8,2}.\end{aligned}$$

Напомним, что  $\sigma_{N/2,j} = \sigma_{N/2,j+N/8}$ . Указанные вычисления выполняются при помощи  $N/4$  умножений на действительные или мнимые константы,  $N/4$  умножений на константы вида  $\pm 1 + a\mathbf{i}$  или  $a \pm \mathbf{i}$  и двух преобразований  $\text{ДПФ}'_{N/8}$ .

В итоге получаем соотношение

$$C_{\mathcal{A}_L^R}(\text{ДПФ}'_N) \leq 3C_{\mathcal{A}_L^R}(\text{ДПФ}'_{N/4}) + 2C_{\mathcal{A}_L^R}(\text{ДПФ}'_{N/8}) + 8.5N,$$

которое в согласии с начальными условиями  $C(\text{ДПФ}'_4) \leq 16$ ,  $C(\text{ДПФ}'_2) = 4$  и  $C(\text{ДПФ}'_1) = 0$ , разрешается как  $C_{\mathcal{A}_L^R}(\text{ДПФ}'_N) \leq 3\frac{7}{9}N \log_2 N$ . Остается учесть  $2N$  операций для перехода от координат  $X$  к  $\dot{X}$ . ■

- Метод теоремы 7.3 также объясняется в [229, 138]. Аккуратная оценка сложности метода имеет вид

$$\frac{34}{9}N \log_2 N - \frac{124}{27}N - 2 \left(1 + \frac{(-1)^k}{9}\right) \log_2 N + 8 + \frac{16}{27} \cdot (-1)^k.$$

Дополнительно используя прием сведения ДПФ к преобразованию Уолша—Адамара, Дж. Альман и К. Рао [123] улучшили оценку вещественной сложности комплексного ДПФ до  $C_{\mathcal{A}_L^R}(\text{ДПФ}_N[\mathbb{C}]) \leq 3.75N \log_2 N + O(N)$ .

### Умножение матриц. Ускорение метода Штрассена $\boxed{e}/\boxed{2}$

Из теоретически быстрых методов умножения матриц наибольший прикладной интерес представляет метод Штрассена [316]. Усилия многих исследователей были сосредоточены на оптимизации времени работы алгоритма. Напомним, что в базовой схеме метода Штрассена в модификации Винограда умножение матриц размера  $2 \times 2$  выполняется за 7 умножений и 15 аддитивных операций в кольце. Если мультипликативная сложность базовой схемы определяет экспоненту сложности метода, то аддитивная сложность приблизительно пропорциональна мультипликативной постоянной в оценке сложности. Так, по формуле (5.10) при  $n = 2^k$  получаем  $C_{\mathcal{A}^R}(MM_n) \leq 6n^{\log_2 7}$ .

М. Бодрато [148] заметил, что быстрое умножение  $2 \times 2$  матриц можно выполнить за 12 аддитивных операций, если использовать альтернативное представление  $X \rightarrow \widehat{X}$  (см. также [237]). Например [237], пусть

$$\widehat{X} = \begin{bmatrix} \widehat{x}_{11} & \widehat{x}_{12} \\ \widehat{x}_{21} & \widehat{x}_{22} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} - x_{21} + x_{22} \\ x_{22} - x_{21} & x_{12} + x_{22} \end{bmatrix}. \quad (7.3)$$

Теперь произведение  $\widehat{Z} = \widehat{X}\widehat{Y}$  может быть вычислено по формулам

$$\begin{aligned} u_1 &= \widehat{x}_{11}\widehat{y}_{11}, & u_2 &= \widehat{x}_{12}\widehat{y}_{12}, & u_3 &= \widehat{x}_{21}\widehat{y}_{21}, & u_4 &= \widehat{x}_{22}\widehat{y}_{22}, \\ u_5 &= (\widehat{x}_{12} - \widehat{x}_{21})(\widehat{y}_{22} - \widehat{y}_{12}), & u_6 &= (\widehat{x}_{12} - \widehat{x}_{11})(\widehat{y}_{12} - \widehat{y}_{21}), & u_7 &= (\widehat{x}_{22} - \widehat{x}_{12})(\widehat{y}_{12} - \widehat{y}_{11}), \\ \widehat{z}_{11} &= u_1 + u_5, & \widehat{z}_{12} &= u_2 + u_5 - u_6 + u_7, & \widehat{z}_{21} &= u_3 + u_7, & \widehat{z}_{22} &= u_4 - u_6, \end{aligned} \quad (7.4)$$

которые используют 7 умножений и 12 сложений-вычитаний.

**Теорема 7.4** ([148, 237]). *Пусть  $n = 2^k$ . Тогда для кольца  $R$  выполнено*

$$C_{\mathcal{A}^R}(MM_n) \leq 5n^{\log_2 7} + O(n^2 \log n).$$

► Рекурсивно распространим определение альтернативного представления на матрицы размера  $2^k \times 2^k$ . При  $k = 1$  представление  $\widehat{X}$  задано формулами (7.3). Если определено представление для матриц размера  $n/2 \times n/2$ , то представление  $n \times n$  матрицы  $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$  зададим как<sup>1</sup>)

$$\widehat{X} = \begin{bmatrix} \widehat{X}_{11} & \widehat{X}_{12} - \widehat{X}_{21} + \widehat{X}_{22} \\ \widehat{X}_{22} - \widehat{X}_{21} & \widehat{X}_{12} + \widehat{X}_{22} \end{bmatrix}. \quad (7.5)$$

<sup>1</sup> Следует заметить, что матричная форма записи для альтернативного представления условия: произведение матриц в измененном представлении определяется не теми правилами, что в стандартном.

**Лемма 7.1.** Для сложности смены представления  $n \times n$  матрицы  $X$  справедливо

$$C_A(X \rightarrow \hat{X}), C_A(\hat{X} \rightarrow X) \leq \frac{3}{4} \cdot n^2 \log_2 n.$$

▷ Однократное применение формул (7.3) или (7.5) стоит 3 аддитивные операции. Так же и в другую сторону: при  $n = 2$

$$X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} = \begin{bmatrix} \hat{x}_{11} & \hat{x}_{12} - \hat{x}_{21} \\ \hat{x}_{22} - \hat{x}_{12} & \hat{x}_{21} - \hat{x}_{12} + \hat{x}_{22} \end{bmatrix}.$$

Таким образом, для сложности  $T(n)$  перехода между представлениями  $n \times n$  матриц имеем рекуррентное соотношение  $T(n) \leq 4T(n/2) + 3n^2/4$ , которое разрешается в точности как заявлено.  $\square$

Обозначим через  $\widehat{MM}_n$  оператор умножения матриц в альтернативном представлении. При его реализации методом теоремы 2.3 или 5.3 выполняется рекурсивный вызов алгоритмов умножения все меньшей и меньшей размерности, при этом всякий раз перемножаемые матрицы имеют готовый для применения формул (7.4) вид. Для сложности  $\widehat{MM}_n$  можно воспользоваться оценкой (5.10)<sup>2)</sup>, в которой  $m = 2$ ,  $r = 7$  и  $s = 12$ . Тогда получаем

$$C(MM_n) \leq C(\widehat{MM}_n) + 3T(n) \leq 5n^{\log_2 7} - 4n^2 + \frac{9}{4} \cdot n^2 \log_2 n.$$

■

Оценка теоремы 7.4 несколько условна, поскольку на практике при малых  $n$  вызывается не метод Штрассена, а, скажем, стандартный алгоритм умножения — в этом случае мультиплекативная константа в оценке сложности оказывается еще меньше (см., например, [163]).

- Бодрато [148] также показал, что при любом представлении  $2 \times 2$  матриц кодами минимальной длины менее чем 12 аддитивными операциями алгоритм умножения штрассеновского типа обойтись не сможет.

Э. Карстадт и О. Шварц [237] также нашли удобные формы представления матриц для других потенциально интересных алгоритмов, в частности, для алгоритма Смирнова [96], основанного на быстром умножении  $3 \times 3$  и  $3 \times 6$  матриц (см. также [135]). Кроме того, более экономные представления возможны для колец характеристики 2, см. [236].

Аналогичный способ ускорения алгоритма Штрассена, опирающийся на другую, избыточную, кодировку, предложили М. Ценк и М. Хасан [163]. Их алгоритм приводит к такой же оценке, как в теореме 7.4,  $\lesssim 5n^{\log_2 7}$ , а при условии применения стандартного алгоритма умножения при малых размерах матриц — к оценке  $C(MM_n) \lesssim 3.55n^{\log_2 7}$  (для  $n = 2^k$ ).

### Сложность формул для сложения по модулю 5 $e[U]/_2[\cdot]$

Рассмотрим задачу вычисления суммы  $n$  булевых переменных по модулю  $m$  в базисе  $\mathcal{B}_0$ . Применение простых формул (4.1) приводит к оценке  $\Phi_{\mathcal{B}_0}(\text{MOD}_n^m) \leq n^{\log_2 m + 1}$  [47]. При  $m = 3$  эта оценка пока не улучшена, а при больших  $m$  более

<sup>2)</sup>Здесь существенно, что новая кодировка сохраняет размер матриц, поэтому сложность аддитивных операций с матрицами не меняется.

короткие формулы позволяет строить метод, предложенный автором в [89]. В его основе лежит специальным образом расширенная кодировка входов.

Пусть  $S \subset \mathbb{Z}_m$ . Определим функции

$$\text{MOD}_n^{m,S}(X) = \left( \sum_{i=1}^n x_i \bmod m \in S \right).$$

Очевидно, набор всех функций  $\text{MOD}_n^{m,S}$ ,  $0 < |S| < m$ , задает значение суммы переменных по модулю  $m$ .

Каждому множеству  $S$  сопоставим булеву  $m \times m$  матрицу  $I_m^S$ , строки и столбцы которой занумерованы числами из  $\mathbb{Z}_m$ , а элементы определяются как  $I_m^S[i, j] = (i + j \in S)$ . Рассмотрим некоторое покрытие матрицы  $I_m^S$  прямоугольниками (сплошь единичными подматрицами): пусть  $k$ -й прямоугольник образован пересечением строк  $A_k$  и столбцов  $B_k$ . Тогда при  $X = (X^1, X^2)$ ,  $|X| = n$ ,  $|X^i| = n_i$ ,

$$\text{MOD}_n^{m,S}(X) = \bigvee_k \text{MOD}_{n_1}^{m,A_k}(X^1) \cdot \text{MOD}_{n_2}^{m,B_k}(X^2). \quad (7.6)$$

Из (4.1) вытекает тождество

$$\text{MOD}_n^{m,S}(X) = \bigvee_{k=0}^{m-1} \text{MOD}_{n_1}^{m,k}(X^1) \cdot \text{MOD}_{n_2}^{m,S-k}(X^2), \quad (7.7)$$

отвечающее покрытию матрицы отдельными строками (здесь  $S - k$  обозначает множество  $\{r - k \mid r \in S\}$ ). Ранг этого покрытия (число покрывающих матриц) равен  $m$ .

Нетривиальные оценки сложности можно получить, используя покрытия ранга  $< m$ . Если в случае  $m = 3$  при любом  $S \neq \emptyset$ ,  $\mathbb{Z}_m$  матрицы  $I_m^S$  имеют полный ранг, то уже при  $m = 5$  матрицы  $I_m^{\mathbb{Z}_m \setminus \{r\}}$  (совпадающие с  $\bar{I}_m$  с точностью до перестановки строк и столбцов) имеют ранг 4, причем покрытие образуют прямоугольники со сторонами 2 и 3, см. рис. 7.4.

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

Рис. 7.4: Покрытие матрицы  $\bar{I}_5$

**Теорема 7.5 ([89]).**  $\Phi_{\mathcal{B}_0}(\text{MOD}_n^5) \prec n^{3.22}$ .

► Итак, функция  $\text{MOD}_n^{5,S}$  при  $|S| = 4$  реализуется формулой (7.6) с 4-мя слагаемыми, а при  $|S| = 1$  — двойственной формулой (конъюнкция дизъюнкций) такого же размера. Для остальных функций  $\text{MOD}_n^{5,S}$  выбираем реализацию (7.7), поскольку матрицы  $I_5^S$  при  $2 \leq |S| \leq 3$  имеют полный ранг.

Используя метод потенциалов, оценим сложность формул, к которым приводит указанная стратегия. Обозначим через  $p_n$  сложность формул для  $\text{MOD}_n^{5,S}$ ,  $|S| \in \{1, 4\}$ , а через  $q_n$  — сложность формул для  $\text{MOD}_n^{5,S}$ ,  $|S| \in \{2, 3\}$ . Пусть  $r_n = \max\{p_n, q_n/a\}$ , где параметр  $a$  будет выбран позднее. Из (7.6) и (7.7) следует

$$r_{2n} \leq \max\{8q_n, 5(p_n + q_n)/a\}.$$

С целью минимизировать отношение  $r_{2n}/r_n$  выбираем  $a = \frac{5+\sqrt{185}}{16}$ , в результате получаем  $r_{2n} \leq 8ar_n$ . Следовательно,  $\Phi_{B_0}(\text{MOD}_n^5) \preccurlyeq n^{3+\log_2 a} \prec n^{3.22}$ . ■

- Незначительное уточнение оценки теоремы 7.5 возможно при дополнительном подборе оптимального отношения  $n_1/n_2$ . Указанным методом в работе [89] также получены оценки

$$\Phi_{B_0}(\text{MOD}_n^7) \prec n^{3.63}, \quad D_{B_0}(\text{MOD}_n^5) \lesssim 3.35 \log_2 n, \quad D_{B_0}(\text{MOD}_n^7) \lesssim 3.87 \log_2 n.$$

При больших  $m$  приоритет имеют оценки глубины и сложности оператора  $\Sigma_{1,n}$ , см. (7.9).

### Быстрое возведение многочленов в степень $e$

Рассмотрим задачу быстрого вычисления степени многочлена  $f(x)^m$  в фактор-кольце  $R[x]/(p(x))$ , где  $\deg f < \deg p = n$ . При реализации арифметики конечных полей с подобными операциями приходится иметь дело регулярно.

Прямолинейный подход состоит в выполнении последовательных умножений с приведением по модулю  $p(x)$ . Если вычислять степени многочлена  $f(x)$ , следуя оптимальной аддитивной цепочке для  $m$ , то всего потребуется  $L(m) \sim \log_2 m$  шагов сложности  $C_{\mathcal{A}^R}(M_n^R) + C_{\mathcal{A}^R}(QR_{2n,n}^R)$  каждый. Даже в благоприятном случае (теорема 5.2) деление с остатком «стоит» примерно двух умножений, а обычно — несколько больше. П. Монтгомери [268] заметил, что умножение в фактор-кольце можно упростить, если перейти к специальному представлению элементов. Далее излагается адаптация числового метода Монтгомери для многочленов.

Положим  $b(x) = x^{n-1}$  и  $q(x) = b^{-1}(x) \bmod p(x)$  (в предположении, что  $p(x)$  имеет ненулевой свободный член, многочлены  $b$  и  $p$  взаимно просты<sup>3</sup>). Рассмотрим преобразование  $f \rightarrow f^* = fb \bmod p$ . Оно устанавливает соответствие, которое на самом деле является гомоморфизмом, если операцию сложения образов ввести обычным образом, а умножение определить как  $f \star g = fgq \bmod p$  (умножение Монтгомери). Тогда  $(f \pm g)^* = f^* \pm g^*$  и  $(fg)^* = f^* \star g^*$ .

Пусть далее  $h(x) = -p^{-1}(x) \bmod b(x)$ . Следующая лемма предлагает простой способ реализации умножения Монтгомери.

---

<sup>3</sup>Если  $p(x) = p_0(x)x^k$ , где  $x \nmid p_0(x)$ , то задача становится проще, т.к. вычисления можно выполнить отдельно по модулям  $x^k$  и  $p_0(x)$ . В приложениях (например, в арифметике конечных полей) многочлен  $p$  обычно неприводим, в частности,  $x \nmid p(x)$ .

**Лемма 7.2.** Пусть  $\deg f \leq 2n - 2$ . Тогда

$$\frac{f + ((f \bmod b)h \bmod b)p}{b} = fq \bmod p. \quad (7.8)$$

▷ Числитель дроби делится на  $b$ , поскольку  $b \mid f(1 + hp)$ . Поэтому дробь является многочленом степени  $< n$ , который, как видно при домножении на  $bq$ , принадлежит тому же классу эквивалентности по модулю  $p$ , что и многочлен  $fq$ . Таким образом, два многочлена просто совпадают.  $\square$

Формула (7.8), если в нее вместо  $f$  подставить произведение  $fg$  многочленов степени  $< n$ , означает, что сложность умножения Монтгомери  $f \star g$  не превосходит  $3C_{\mathcal{A}^R}(M_n^R) + O(n)$ . Переход от  $f$  к  $f^* = f \star (b^2 \bmod p)$  стоит трех умножений, а в обратную сторону — двух.

**Теорема 7.6.** Пусть  $p(x) \in R[x]$ ,  $\deg p = n$  и  $x \nmid p(x)$ . Возведение в фиксированную степень  $m$  в кольце  $R[x]/(p(x))$  можно выполнить схемой сложности  $(3L(m) + 3)(C_{\mathcal{A}^R}(M_n^R) + O(n))$  над базисом  $\mathcal{A}^R$ .

► Руководствуясь кратчайшей аддитивной цепочкой  $a_0 = 1, a_1, \dots, a_k = m$  для числа  $m$ , последовательно вычислим многочлены  $\varphi_i = f^{a_i} q^{a_i-1} \bmod p$  по правилу (7.8), стартуя с  $f = f^{a_0} q^{a_0-1} \bmod p$ . Окончательно  $f^m \bmod p = \varphi_k \star a$ , где  $\varphi_k = f^m q^{m-1} \bmod p$ , а многочлен  $a = b^m \bmod p$  вычислен заранее.  $\blacksquare$

Более изящная схема вычислений  $f \rightarrow f^* \rightarrow (f^m)^* \rightarrow f^m$  (по модулю  $p$ ) полноценно использует вспомогательную кодировку, но требует на два умножения больше. Однако этот способ эффективен в случае нефиксированного многочлена  $p$  или когда число  $m$  тоже является входом алгоритма и задано двоичным кодом (что часто встречается на практике; тогда используется бинарная аддитивная цепочка для  $m$ ). Разумеется, кодировка Монтгомери подходит для широкого круга арифметических задач в фактор-кольце, не только для возведения в степень.

- Почти без изменения методика модульных вычислений переносится на числовые кольца  $\mathbb{Z}_p$  — для них она и была изначально разработана [268]. Обычно  $p$  — простое  $n$ -разрядное число, а  $b = 2^n$ . Лемма 7.2 уточняется следующим образом: при  $f < p^2$  левая часть (7.8) либо совпадает с правой, либо превосходит ее на  $p$ .

Альтернативой методу Монтгомери служит метод П. Барретта [127]. Метод основан на подмене делителя: частное  $\lfloor f/p \rfloor$  приблизительно равно  $\lfloor \lfloor f/b \rfloor \cdot \lfloor b^2/p \rfloor / b \rfloor$ . Если предварительно вычислить  $\lfloor b^2/p \rfloor$ , далее умножение по модулю  $p$  выполняется при помощи трех обычных умножений и нескольких операций линейной сложности.

## Другие приложения

**Формулы для симметрических булевых функций.** Произвольная симметрическая булева функция  $n$  переменных имеет вид  $h(\Sigma_{1,n}(X))$ , и стандартный метод ее вычисления заключается в следующем: сначала вычисляется арифметическая сумма входов, а затем реализуется функция  $h$  от  $\log_2 n$  переменных. При этом, поскольку разные разряды арифметической суммы, вообще говоря, имеют различную сложность и глубину, на втором шаге, как показано

в [105, 278], выгодно использовать метод каскадов<sup>4)</sup>. Так, в случае использования стандартных  $(3, 2)$ -компрессоров верхние оценки глубины оператора  $\Sigma_{1,n}$  и класса симметрических функций  $\mathcal{S}_n$  соотносятся как [278]

$$\mathsf{D}_{\mathcal{B}_2}(\Sigma_{1,n}) \lesssim 3.71 \log_2 n, \quad \mathsf{D}_{\mathcal{B}_2}(\mathcal{S}_n) \lesssim 3.81 \log_2 n$$

(первая оценка доказана в следствии 4.1).

В методе автора [88] сумма  $\Sigma = \Sigma_{1,n}(X)$  кодируется тройкой  $(\Sigma_2, \Sigma_3, \Sigma')$ , где  $\Sigma_2 = \Sigma \bmod 2^k$ ,  $\Sigma_3 = \Sigma \bmod 3^l$  и  $|\Sigma' - \Sigma| < E$  при  $2^k \cdot 3^l \cdot (2E - 1) \geq n$ . Величина  $\Sigma_2$  вычисляется предложенной в [278] модификацией метода компрессоров, величина  $\Sigma_3$  — аналогично, только в троичной системе счисления при помощи специальных троичных компрессоров, приближенное значение суммы  $\Sigma'$  вычисляется методом Л. Вэльянта [325], см. далее на стр. 106. Истинное значение суммы  $\Sigma = \Sigma_{1,n}(X)$  определяется из своего кода при помощи несложной арифметической процедуры в духе Китайской теоремы об остатках. Симметрическую функцию общего вида можно сразу вычислять как  $h(\Sigma_2, \Sigma_3, \Sigma')$ , не восстанавливая сумму  $\Sigma$ . Эффективность метода обусловлена тем, что младшие разряды сумм в методе компрессоров вычисляются проще, чем старшие. На этом пути получены рекордные на сегодняшний день оценки:

$$\begin{aligned} \Phi_{\mathcal{B}_0}(\Sigma_{1,n}) &\leq n^{3.91}, & \Phi_{\mathcal{B}_0}(\mathcal{S}_n) &\leq n^{4.01}, & \Phi_{\mathcal{B}_2}(\Sigma_{1,n}) &\leq n^{2.84}, & \Phi_{\mathcal{B}_2}(\mathcal{S}_n) &\leq n^{2.95}, \\ \mathsf{D}_{\mathcal{B}_0}(\Sigma_{1,n}) &\lesssim 4.14 \log_2 n, & \mathsf{D}_{\mathcal{B}_0}(\mathcal{S}_n) &\lesssim 4.24 \log_2 n, & \mathsf{D}_{\mathcal{B}_2}(\Sigma_{1,n}) &\lesssim 3.02 \log_2 n, & \mathsf{D}_{\mathcal{B}_2}(\mathcal{S}_n) &\lesssim 3.1 \log_2 n. \end{aligned} \quad (7.9)$$

Эти оценки, однако, неконструктивны в силу способа вычисления  $\Sigma'$ . Конструктивная версия метода, использующая сокращенную кодировку  $(\Sigma_2, \Sigma_3)$ , предложена автором ранее в [85, 86].

**Почти монотонная сложность булевых функций.** В монотонных или преимущественно монотонных вычислениях предварительная сортировка векторов переменных нередко позволяет строить более простые схемы. Рассмотрим известный пример. В 1957 г. А. А. Марков [52] установил фундаментальный факт: любую систему булевых функций от  $n$  переменных можно реализовать схемой в базисе  $\mathcal{B}_0$ , используя всего  $b(n) = \lceil \log_2(n+1) \rceil$  элементов отрицания<sup>5)</sup>. Любая схема над  $\mathcal{B}_0$  достаточно эффективно преобразуется в схему с  $b(n)$  отрицаниями. Сначала отрицания опускаются на уровень входов по правилам де Моргана — при этом сложность схемы не более чем удваивается. После этого остается реализовать оператор  $V_n = (\bar{x}_1, \dots, \bar{x}_n)$ , экономно используя элементы отрицания.

Рекордная верхняя оценка сложности вычисления  $V_n$  схемой с  $b(n)$  элементами отрицания получена Р. Билзом [130] (см. также [131]) и имеет величину  $O(n \log n)$ . В основе метода лежит наблюдение М. Фишера [189] о том, что на упорядоченном наборе входов оператор  $V_n$  реализуется просто, с линейной сложностью.

Действительно, при  $n = 3$  и  $x_1 \geq x_2 \geq x_3$  схема с  $b(3) = 2$  отрицаниями строится так. Вычисляется  $\bar{x}_2$ , затем  $x_1 \bar{x}_2 \vee x_3$ , затем  $(x_1 \bar{x}_2 \vee x_3) = (\bar{x}_1 \vee x_2) \bar{x}_3$ . Остается заметить, что

$$\bar{x}_2((\bar{x}_1 \vee x_2) \bar{x}_3) = \bar{x}_1 \bar{x}_2 \bar{x}_3 = \bar{x}_1 \quad \text{и} \quad \bar{x}_2 \vee ((\bar{x}_1 \vee x_2) \bar{x}_3) = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 = \bar{x}_3.$$

Метод обобщается на случай произвольного  $n = 2k - 1$ : описанные вычисления выполняются с тройками  $x_i \geq x_k \geq x_{k+i}$ , в середине выполняется вызов алгоритма, вычисляющего  $V_{k-1}$ .

Легко проверить, что общая сложность схемы не превосходит  $4n$ . Выгоду от упорядочения входного набора демонстрирует и нижняя оценка, доказанная К. Танакой и Т. Нисино [319]: в общем случае сложность реализации оператора  $V_n$  схемами с  $b(n)$  отрицаниями не может быть меньше, чем  $5n$  (при  $n \geq 3$ ).

Оборотной стороной монотонной кодировки является нелинейная сложность перехода к ней и в обратную сторону<sup>6)</sup>. Переход  $x_1, \dots, x_n \xrightarrow{\text{SORT}_n} x_{\pi(1)}, \dots, x_{\pi(n)}$  выполняется схемой ком-

<sup>4)</sup>Метод заключается в последовательном применении разложения функции по выбранной переменной:  $f(x_1, x_2, \dots, x_n) = x_1 f(1, x_2, \dots, x_n) \vee \bar{x}_1 f(0, x_2, \dots, x_n)$ , см., например, [48].

<sup>5)</sup>Более того, он указал правило, позволяющее определить инверсионную сложность (минимальное число элементов отрицания) функции по таблице ее значений. Для произвольной функции  $n$  переменных инверсионная сложность не превосходит  $\lfloor \log_2(n+1) \rfloor$ .

<sup>6)</sup>Речь идет, конечно, о монотонной части кода, используемого для входного набора.

параторов<sup>7)</sup> со сложностью  $\preccurlyeq n \log n$ , если использовать методы быстрой сортировки из семейства [119]. Ключевое наблюдение Билза [130] состоит в том, что для возврата к исходному порядку переменных  $\bar{x}_{\pi(1)}, \dots, \bar{x}_{\pi(n)} \rightarrow \bar{x}_1, \dots, \bar{x}_n$  можно обратить исходную схему сортировки. Действительно, если компаратор в схеме выполняет преобразование  $(f, g) \rightarrow (f \vee g, f \cdot g)$ , то по известным  $\bar{f} \vee \bar{g} = \bar{f} \cdot \bar{g}$  и  $\bar{f} \cdot g = \bar{f} \vee \bar{g}$  функции  $\bar{f}$  и  $\bar{g}$  вычисляются посредством формул

$$\bar{f} = \bar{f} \cdot \bar{g} \vee (\bar{f} \vee \bar{g})g, \quad \bar{g} = \bar{f} \cdot \bar{g} \vee (\bar{f} \vee \bar{g})f.$$

Как следствие, переход к истинному порядку переменных также выполняется со сложностью  $\preccurlyeq n \log n$ . Таким образом, информация о структуре сортирующей схемы дополняет монотонную кодировку входного набора.

Вместо сортировки в методе достаточно использовать выбор среднего элемента, схемы получаются проще, но порядок сложности не меняется<sup>8)</sup>. Вопрос о существовании схемы линейной сложности для  $V_n$  остается открытым. Х. Моризуми и Г. Сузуки [271] построили схему линейной сложности, используя  $\log^{1+o(1)} n$  элементов отрицания.

**Интерполяция и вычисление значений многочлена в точках арифметической прогрессии.** Известно, что вычисление значений многочлена степени  $< n$  на наборе из  $n$  точек, а также обратная задача восстановления коэффициентов многочлена по значениям в точках выполняются со сложностью  $\preccurlyeq M(n) \log n$ : уточненные в [152] оценки имеют вид соответственно  $\lesssim 1.5M(n) \log_2 n$  и  $\lesssim 2.5M(n) \log_2 n$  при  $n = 2^k$ . Указанные операции можно выполнить еще быстрее, если множество точек имеет специальную структуру. В частности, для случая арифметической прогрессии Ю. Герхард [202] указал чуть более эффективный алгоритм. Он основан на переходе от стандартной записи многочлена  $f(x) = \sum f_i x^i$  к записи в форме Ньютона.

Обозначим через  $\alpha_0, \dots, \alpha_{n-1} \in R$  набор точек интерполяции. Ньютоново представление многочлена  $f(x)$  степени  $< n$  характеризуется вектором коэффициентов  $(g_0, \dots, g_{n-1})$ , где

$$f(x) = g_0 + g_1(x - \alpha_0) + g_2(x - \alpha_0)(x - \alpha_1) + \dots + g_{n-1}(x - \alpha_0) \cdots (x - \alpha_{n-2}).$$

Пусть  $\alpha_i = \alpha_0 + ih$ , где  $h$  — разность арифметической прогрессии. В силу тождества

$$V(x) = G(x)S(x) \bmod x^n, \quad \text{где } V(x) = \sum_{i=0}^{n-1} \frac{f(\alpha_i)}{i!h^i} x^i, \quad G(x) = \sum_{i=0}^{n-1} g_i x^i, \quad S(x) = \sum_{i=0}^{n-1} \frac{1}{i!h^i} x^i,$$

все значения  $f(\alpha_i)$  вычисляются по заданным  $g_i$  со сложностью  $C_A(M_n^R) + O(n)$ . С такой же сложностью выполняется обратная задача, так как

$$G(x) = V(x)S^{-1}(x) \bmod x^n, \quad \text{где } S^{-1}(x) = \sum_{i=0}^{n-1} \frac{(-1)^i}{i!h^i} x^i.$$

Переход между стандартным и ньютоновым представлением многочлена выполняется фактически методом леммы 6.4 с использованием вспомогательного дерева. Аккуратная оценка сложности перехода в любую сторону, выполненная А. Бостаном и Э. Шостом в [152], имеет величину  $\lesssim M(n) \log_2 n$  при  $n = 2^k$ . Следовательно, так же оценивается сложность и вычисления значений в точках арифметической прогрессии, и обратной задачи интерполяции.

Отметим, что в случае когда точки  $\alpha_0, \dots, \alpha_{n-1}$  образуют геометрическую прогрессию, сложность вычисления значений и интерполяции имеет порядок  $M(n)$  как в стандартном, так и в ньютоновом представлении многочленов [152].

<sup>7</sup>Напомним, что компаратор — это схема, выполняющая преобразование  $(x, y) \rightarrow (x \vee y, xy)$ . Схема из компараторов — это фактически формула, составленная из элементов-компараторов, в том смысле, что ветвления входов переменных и выходов компараторов запрещены. Множество таких схем реализует различные частичные порядки наборов переменных.

<sup>8</sup>Нижняя оценка сложности схем компараторов для выбора среднего элемента имеет величину  $\Omega(n \log n)$  [2].

# Глава 8

## Принципы двойственности

$\boxed{d}$

Концепция двойственности — эффективный инструмент в теории синтеза. В общей форме, она позволяет доказывать близость сложности решения двух задач, двойственных в определенном смысле. Так, построив быстрый алгоритм для двойственной задачи, можно получить столь же эффективное решение (или установить его существование) для исходной.

**Сложность универсальной матрицы. Принцип транспонирования**  $\boxed{d}$

Рассмотрим задачу вычисления линейного преобразования с булевой  $k \times 2^k$  матрицей  $Y_k$ , составленной из всевозможных различных столбцов высоты  $k$ .

$$Y_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Универсальные матрицы играют важную роль в теории синтеза, а также являются проверочными для кодов Хемминга.

**Теорема 8.1.**  $L(Y_k) = 2^{k+1} - 2k - 2$ .

► Результат теоремы следует из более общего факта, известного как *принцип транспонирования*. Вероятно, он был впервые сформулирован Б. С. Митягиным и Б. Н. Садовским [53] в 1965 г.

**Лемма 8.1** ([53]). *В любой коммутативной полугруппе  $(G, +)$  для любой  $m \times n$  матрицы  $A$  без нулевых строк и столбцов выполнено*

$$L_+(A) + m = L_+(A^T) + n. \quad (8.1)$$

► Схему, выполняющую преобразование  $X \rightarrow AX$ , удобно представлять в виде графа, ориентированного от входов к выходам. В каждой вершине выполняется сложение промежуточных сумм, поступающих по входящим ребрам. Если в

вершину входит  $q$  ребер, то вычисление суммы выполняется за  $q - 1$  операций. Заметим, что эквивалентная сложность схемы в пересчете на двухвходовые операции сложения равна разности между числом ребер и числом вершин, не считая входов.

Далее заметим, что элемент матрицы  $A[i, j]$  равен числу ориентированных путей между  $j$ -м входом и  $i$ -м выходом (пути подсчитываются в согласии с групповой операцией).

Теперь, если в произвольной схеме  $S$  изменить ориентацию ребер на противоположную, полученная схема будет вычислять транспонированную матрицу  $A^T$  (по свойству сохранения числа путей), а ее эквивалентная сложность будет равна  $L(S) + m - n$  (общее число вершин в графе схемы сохранилось, только выходы стали входами).  $\square$

Теперь теорема доказывается совсем просто. Транспонированная матрица  $Y_k^T$  содержит  $2^k - k - 1$  различных строк веса  $\geq 2$ . Согласно лемме 3.1, эта оценка достижима. Значит,  $L(Y_k^T) = 2^k - k - 1$ . Осталось применить лемму 8.1, не забыв учесть наличие нулевого столбца в  $Y_k$ .  $\blacksquare$

Верхнюю оценку теоремы 8.1 можно доказать и методом деления пополам, однако применение леммы 8.1 дает одновременно доказательство оптимальности схемы. С другой стороны, хотя принцип транспонирования конструктивен, схемы, построенные с его помощью, могут иметь замысловатую структуру.

- Теорема 8.1 в частности означает, что  $L_{\oplus}(Y_k) = 2^{k+1} - 2k - 2$ . Более того, А. В. Чашкин [109] (см. также [110]) показал, что добавление в базис нелинейных операций не дает выигрыша:  $C_{B_2}(L_{Y_k}) = 2^{k+1} - 2k - 2$ . Скорее всего, это характерно для линейных операторов вообще. С другой стороны, без линейных операций приходится сложнее: как показали А. С. Куликов, О. Меланич и И. Михайлин [247],  $C_{U_2}(L_{Y_k}) \geq 5(2^k - k - 1)$ .

В более широком линейном базисе справедлив аналог леммы 8.1, см. лемму 8.6 ниже.

С помощью принципа транспонирования без труда устанавливается, что оценка (3.4) аддитивной сложности набора чисел  $n_1, \dots, n_s$  справедлива и для сложности вычисления вектора  $(n_1, \dots, n_s)$ , иначе говоря, для сложности линейного преобразования  $x_1, \dots, x_s \rightarrow n_1x_1 + \dots + n_sx_s$ .

Вообще, лемма 8.1 оказывается полезной если не в получении принципиально новых результатов, то в упрощении доказательств. Скажем, лемма 3.3 доказывается чуть проще с применением принципа транспонирования, при этом несколько точнее получается оценка.

### Параллельные схемы сумматоров. Метод Гринчука $\boxed{d}/2$

Результат теоремы 2.6 о сложности вычисления системы переносов не может быть улучшен без дополнительных предположений относительно полукольца  $R$ , в котором выполняются вычисления. Но в наиболее интересном случае  $R = (\mathbb{B}, \vee, \wedge)$  М. И. Гринчук [15] получил более сильную оценку, существенно используя двойственность операций дизъюнкции и конъюнкции.

Напомним, что двойственная к булевой функции  $f$  функция обозначается  $f^*$  и определяется как  $f^*(X) = \overline{f(\overline{X})}$ . Схема для функции  $f$  превращается в схему для  $f^*$  при замене всех элементов на двойственные (это *принцип двойственности*). В частности,  $C_{B_M}(f) = C_{B_M}(f^*)$ . Подробнее см. в [116].

В методе Гринчука вычисление функции одного типа сводится к вычислению функции двойственного типа. Принцип двойственности позволяет игнорировать разницу между типами и провести рекурсивное рассуждение. Тривиальной иллюстрацией такого подхода служит и доказательство теоремы 1.1.

Итак, задача построения параллельного сумматора состоит в минимизации глубины функций

$$F_n(X, Y) = y_{n-1} \vee x_{n-1}(y_{n-2} \vee \dots \vee x_2(y_1 \vee x_1 y_0) \dots). \quad (8.2)$$

**Теорема 8.2** ([15]).  $D_{\mathcal{B}_M}(F_n) \leq \log_2 n + \log_2 \log n + O(1)$ .

► Как и в доказательстве теоремы 2.6, рассмотрим более широкое семейство функций:

$$\begin{aligned} F_{r,2k-1}(X, Y) &= x_{r+k-1} \cdot \dots \cdot x_k(y_{k-1} \vee x_{k-1}(\dots(y_1 \vee x_1 y_0) \dots)), \\ F_{r,2k}(X, Y) &= x_{r+k-1} \cdot \dots \cdot x_k(y_{k-1} \vee x_{k-1}(\dots(y_1 \vee x_1(y_0 \vee x_0)) \dots)). \end{aligned}$$

(Второй индекс в обозначении функции указывает число переменных в части с чередованием операций.) По построению,  $F_i = F_{0,2i-1}$ . Обозначим  $d(r, m) = D_{\mathcal{B}_M}(F_{r,m})$ .

**Лемма 8.2.**

$$d(r+s, m) \leq \max\{\lceil \log_2 r \rceil, d(s, m)\} + 1, \quad (8.3)$$

$$d(r, 2s+m) \leq \max\{d(r, 2s), d(s+1, m-1)\} + 1. \quad (8.4)$$

► Первое соотношение тривиально. Второе вытекает из разложения, обобщающего тождество  $y_1 \vee x_1 y_0 = (y_1 \vee x_1)(y_1 \vee y_0)$ .

Пусть  $X^k$  и  $Y^k$  обозначают (под)наборы переменных  $X$  и  $Y$  с индексами, отсчитываемыми вверх от  $k$ . Положим  $t = \lceil m/2 \rceil$ , и также  $P = x_{r+s+t-1} \cdot \dots \cdot x_{s+t}$  и  $Q = y_{s+t-1} \vee \dots \vee y_t$ . Тогда можно записать

$$\begin{aligned} F_{r,2s+m}(X, Y) &= P \cdot F_{0,2s+m}(X, Y) = P \cdot F_{0,2s}(X^t, Y^t) \cdot (Q \vee F_{0,m}(X, Y)) = \\ &= F_{r,2s}(X^t, Y^t) \cdot F_{s+1,m-1}^*(Y^{(m \bmod 2)-1}, X^{m \bmod 2}). \end{aligned} \quad (8.5)$$

В силу принципа двойственности, двойственные функции  $f$  и  $f^*$  имеют одинаковую глубину в базисе  $\mathcal{B}_M$ , откуда сразу следует (8.4).  $\square$

При  $h \geq 2$  и  $0 \leq r \leq 2^h - 1$  определим вспомогательные величины  $\nu(h, r)$  правилами:

$$\nu(2, 0) = \nu(2, 1) = \nu(2, 2) = 2, \quad \nu(2, 3) = 0,$$

а при  $h > 2$  рекуррентно как

$$\nu(h+1, r) = \begin{cases} \nu(h, r-2^h), & 2^h \leq r < 2^{h+1}, \\ \nu(h, r) + \nu(h, 1+\nu(h, r)/2), & 0 \leq r < 2^h. \end{cases} \quad (8.6)$$

Деление на 2 всегда возможно, поскольку функция  $\nu(h, r)$  принимает только четные значения.

Величина  $\nu(h, r)$  имеет смысл нижней оценки наибольшего числа  $m$ , такого что функция  $F_{r,m}$  реализуется с глубиной  $h$ . Это формально доказывает следующая лемма.

**Лемма 8.3.** *При любых  $h \geq 2$  и  $r < 2^h$  справедливо  $d(r, \nu(h, r)) \leq h$ .*

▷ При  $h = 2$  утверждение проверяется непосредственно. Докажем индуктивный переход от  $h$  к  $h + 1$ .

Если  $2^h \leq r < 2^{h+1}$ , то согласно (8.3), (8.6) и индуктивному предположению

$$d(r, \nu(h+1, r)) = d(r, \nu(h, r - 2^h)) \leq \max\{h, d(r - 2^h, \nu(h, r - 2^h))\} + 1 = h + 1.$$

Иначе, если  $r < 2^h$ , то, применяя (8.6), затем (8.4) и предположение индукции, получаем

$$\begin{aligned} d(r, \nu(h+1, r)) &= d(r, \nu(h, r) + \nu(h, 1 + \nu(h, r)/2)) \leq \\ &\leq \max\{d(r, \nu(h, r)), d(1 + \nu(h, r)/2, \nu(h, 1 + \nu(h, r)/2))\} + 1 \leq h + 1. \end{aligned}$$

□

**Лемма 8.4.**  $\nu(h, r) \geq \frac{2^h - r - 1}{h}$ .

▷ При  $h \leq 3$  неравенство устанавливается непосредственно. Докажем индуктивный переход от  $h$  к  $h + 1$ .

Если  $r \geq 2^h$ , то

$$\nu(h+1, r) = \nu(h, r - 2^h) \geq \frac{2^h - (r - 2^h) - 1}{h} > \frac{2^{h+1} - r - 1}{h+1}.$$

В случае  $r < 2^h$  имеем

$$\begin{aligned} \nu(h+1, r) &= \nu(h, r) + \nu(h, 1 + \nu(h, r)/2) \geq \\ &\geq \frac{2^h - 2}{h} + \left(1 - \frac{1}{2h}\right) \nu(h, r) \geq \frac{2^h - 2}{h} + \left(1 - \frac{1}{2h}\right) \frac{2^h - r - 1}{h}. \end{aligned}$$

Полученная оценка  $l_1(r)$  зависит от  $r$  линейно с коэффициентом  $a = -\frac{2h-1}{2h^2}$ . Поэтому для того, чтобы убедиться, что  $l_1(r) \geq l_2(r) = \frac{2^{h+1}-r-1}{h+1}$ , линейной функции от  $r$  с коэффициентом  $-\frac{1}{h+1} > a$ , на отрезке  $[0, 2^h - 1]$ , достаточно сравнить значения функций в правой точке отрезка,  $r = 2^h - 1$ . При  $h \geq 3$  выполнено

$$l_1(2^h - 1) = \frac{2^h - 2}{h} \geq \frac{2^h}{h+1} = l_2(2^h - 1),$$

что завершает доказательство индуктивного перехода.

□

При  $r = 0$  лемма 8.4 дает оценку  $\nu(h, 0) \geq (2^h - 1)/h$ . Тогда с помощью леммы 8.3 выводим  $d(0, n) \leq \log_2 n + \log_2 \log n + O(1)$ . ■

Подчеркнем, что ключевым пунктом доказательства является формула (8.5), справедливая в силу двойственности функций базиса.

**Следствие 8.1** ([15]).  $D_{B_0}(\Sigma_n) \leq \log_2 n + \log_2 \log n + O(1)$ .

- Метод теоремы 8.2 оптимален с точностью до аддитивной постоянной в оценке глубины, т.к. ранее Б. Комменц-Вальтер доказала [170] оценку  $D_{B_M}(F_n) \geq \log_2 n + \log_2 \log n - O(1)$ . Не должно удивлять, что принцип двойственности играет роль и в доказательстве нижней оценки. Вместе с Ю. Саттлером [171] они получили и оценку для полного базиса

$$D_{B_0}(F_n) \geq \log_2 n + (1 - o(1)) \log_2 \log \log n,$$

откуда, как заметил В. М. Храпченко [108], в силу  $D_{B_0}(F_n) \leq D_{B_0}(\Sigma_n) + O(1)$  следует аналогичная нижняя оценка для глубины сложения  $D_{B_0}(\Sigma_n)$ .

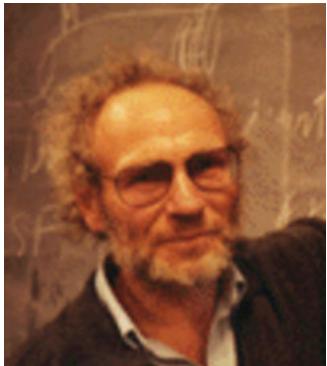
При помощи конструкции Гринчука А. Херманн [219] построила  $n$ -разрядный сумматор линейной сложности и глубины  $\log_2 n + \log_2 \log n + \log_2 \log \log n + O(1)$ .

### Умножение прямоугольных и квадратных матриц $\boxed{d}$

Важнейшим инструментом в теории быстрых алгоритмов умножения матриц является трилинейное тождество, найденное В. Я. Паном [64], позволяющее, в частности, сводить умножение прямоугольных матриц к умножению квадратных.

**Лемма 8.5** ([64]). *В коммутативном кольце  $R$  для любой перестановки  $\pi$  индексов  $m, p, q$ :*

$$\text{rk}^R MM_{m,p,q} = \text{rk}^R MM_{\pi(m),\pi(p),\pi(q)}, \quad \underline{\text{rk}}^R MM_{m,p,q} = \underline{\text{rk}}^R MM_{\pi(m),\pi(p),\pi(q)}.$$



Виктор Яковлевич  
Пан

Городской университет  
Нью-Йорка, с 1988

▷ Докажем только второе равенство. Пусть оператор  $MM_{m,p,q}$  допускает  $(d, r)$ -представление типа (5.11):

$$u^d XY = \sum_{l=1}^r C_l(u) X_l(u) Y_l(u) \bmod u^{d+1}. \quad (8.7)$$

Транспонируя его<sup>1)</sup>, получаем

$$u^d Y^T X^T = \sum_{l=1}^r C_l^T(u) Y_l(u) X_l(u) \bmod u^{d+1}.$$

Таким образом,

$$\underline{\text{rk}} MM_{q,p,m} \leq \underline{\text{rk}} MM_{m,p,q}. \quad (8.8)$$

Скалярно умножая (8.7) на  $m \times q$  матрицу  $Z^T$  (или, что то же самое, выполняя тензорную свертку<sup>2)</sup> с  $Z^T$ ), получаем *трилинейное тождество Пана* для  $(d, r)$ -представлений:

$$u^d \sum_{1 \leq i \leq m, 1 \leq j \leq p, 1 \leq k \leq q} x_{ij} y_{jk} z_{ki} = \sum_{l=1}^r X_l(u) Y_l(u) Z_l(u) \bmod u^{d+1}, \quad (8.9)$$

<sup>1</sup>Здесь используется коммутативность кольца.

<sup>2</sup>При подходящей расстановке индексов, например,  $x_i^j y_j^k z_k^i$ .

где  $Z_l(u)$  — линейные комбинации переменных  $z_{ik}$ .

Подставляя в (8.9)  $x_{ij} = 1$  и  $x_{i'j'} = 0$  при всех  $(i', j') \neq (i, j)$ , получаем  $(d, r)$ -представление для произвольной суммы  $\sum_{k=1}^q y_{jk} z_{ik}$ . Все такие представления можно объединить формулой

$$u^d Y Z^T = \sum_{l=1}^r C'_l(u) Y_l(u) Z_l(u) \bmod u^{d+1},$$

где  $C'_l(u) \in R[u]^{p \times m}$ . Таким образом,

$$\underline{\text{rk}} MM_{m,p,q} \leq \underline{\text{rk}} MM_{p,q,m}. \quad (8.10)$$

Вместе (8.8) и (8.10) влечут утверждение леммы.  $\square$

Лемма 8.5 устанавливает двойственность между билинейными алгоритмами умножения матриц, одинаковых с точностью до перестановок линейных размежеров. При помощи леммы 5.4 для ранга произведения квадратных матриц сразу получаем

**Следствие 8.2.** В коммутативном кольце  $R$

$$\text{rk}^R MM_{mpq} \leq (\text{rk}^R MM_{m,p,q})^3, \quad \underline{\text{rk}}^R MM_{mpq} \leq (\underline{\text{rk}}^R MM_{m,p,q})^3.$$

- При помощи этой техники итальянские математики [144] вывели из  $\underline{\text{rk}} MM_{2,3,2} \leq 10$  неравенство  $\underline{\text{rk}} MM_{12} \leq 1000$ , что с учетом наблюдения Д. Бини [143] (теорема 5.4) влечет  $C_A(MM_n) \leq n^{\log_{12} 1000 + o(1)} \prec n^{2.78}$  — на тот момент это была рекордная оценка сложности умножения матриц.

Практический интерес представляет оценка А. В. Смирнова  $\text{rk} MM_{3,3,6} \leq 40$  [96]. На практике алгоритмы теоремы 5.3, основанные на оценках для рангов, имеют приоритет перед алгоритмами теоремы 5.4, основанными на оценках граничных рангов. Таким образом, метод умножения матриц со сложностью порядка  $n^{3 \log_{54} 40} \prec n^{2.775}$  имеет прикладной потенциал почти на уровне алгоритма Штрассена. В плане практического применения также заслуживают внимания результаты В. Я. Пана, полученные непосредственным построением экономных трилинейных декомпозиций. В частности, в [274] он показал, что  $\text{rk} MM_{44} \leq 36133$ , откуда  $C_A(MM_n) \leq n^{\log_{44} 36133} \prec n^{2.774}$  (на текущий момент, это рекордный результат для сложности умножения матриц, не использующий оценки граничных рангов). Подробнее см. в [65]. Практически значимое ускорение алгоритмов Пана (примерно в духе теоремы 7.4) получили Т. Хадас и О. Шварц [210].

### Сложность рациональной функции и ее градиента $\boxed{d}$

Важную роль в теории сложности арифметических схем играет обнаруженный В. Бауром и Ф. Штрассеном факт об эквивалентности арифметической сложности рациональной функции (например, многочлена) и ее градиента [129]. Напомним, что градиент функции  $f(x_1, \dots, x_n)$  определяется как  $\nabla f = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)$ . Изящный способ доказательства результата Баура—Штрассена, основанный на применении принципа транспонирования линейных схем, предложили братья С. Б. и И. Б. Гашковы в [10]. Предварительно нам понадобится следующее обобщение леммы 8.1.

**Лемма 8.6** ([82]). Для любой  $m \times n$  матрицы  $A$  над кольцом  $R$  с единицей выполнено

$$C_{\mathcal{A}_L^R}(L_{AT}) \leq C_{\mathcal{A}_L^R}(L_A) + m - 1, \quad (8.11)$$

причем это неравенство достигается на схемах (для  $L_A$  и  $L_{AT}$ ) с одним и тем же набором элементов скалярного умножения, не считая умножений на  $-1$ .

▷ Мы докажем чуть более слабую оценку  $C_{\mathcal{A}_L^R}(L_{AT}) \leq C_{\mathcal{A}_L^R}(L_A) + m$ . Доказательство очень похоже на доказательство леммы 8.1. Рассмотрим (ориентированный) граф схемы, вычисляющей  $L_A$ , и припишем ребрам графа веса: вес 1 для ребер, ведущих в элемент сложения, веса 1 и  $-1$  для ребер, ведущих в элемент вычитания, и вес  $a$  — для ребра, ведущего в элемент умножения на  $a$ . Граф с весами содержит полную информацию об исходной схеме

В качестве иллюстрации на рис. 8.1а приведена схема, вычисляющая преобразование  $y_1 = x_1 - x_2 - ax_3$ ,  $y_2 = x_2 - ax_3$ . (Операция умножения на константу обозначена  $\bullet$ , отрицательные входы элементов вычитания отмечены кружками.) На рис. 8.1б изображен граф схемы.

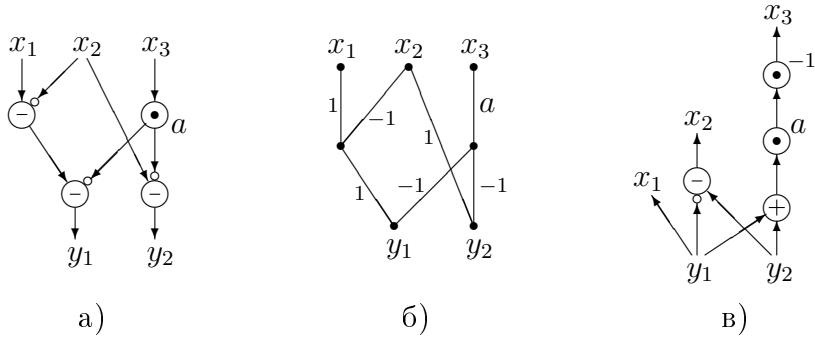


Рис. 8.1: Преобразование схемы (а) в граф (б) и в транспонированную схему (в).

Вес ориентированного пути в графе определим как произведение весов составляющих путь ребер. По построению, элемент матрицы  $A[i, j]$  равен сумме весов всех путей, соединяющих  $i$ -й вход и  $j$ -й выход. В таком смысле граф вычисляет матрицу  $A$ .

При обращении ориентации графа получим граф, вычисляющий матрицу  $A^T$  в указанном смысле. Осталось превратить его в схему над базисом  $\mathcal{A}_L$ .

Для этого сначала ко всем ребрам с весами, отличными от  $\pm 1$ , присоединим элементы скалярного умножения<sup>3)</sup> (тем самым, мы восстанавливаем набор скалярных умножений исходной схемы). После этого в схеме остаются только ребра с весами  $\pm 1$ . Оставшиеся вершины графа должны быть заменены деревьями из аддитивных элементов.

Выполним процедуру «подъема» отрицательных меток от входов к выходам. Если не все ребра, входящие в вершину, имеют метки  $-1$ , то суммирование в

<sup>3)</sup>Более точно, мы ребро веса  $a$  заменяем элементом умножения на  $a$  с входящим и исходящим ребрами веса 1.

данной вершине реализуется деревом из двухходовых сложений и вычитаний<sup>4)</sup>. Иначе, инвертируем метки входящих в вершину ребер и исходящих из нее, за исключением ребер, ведущих к элементам скалярного умножения — в этом случае инвертирование применяется к ребрам, исходящим из указанных элементов умножения. Процесс обрывается на выходах схемы.

Если все ребра, присоединенные к выходу схемы, имеют метки  $-1$ , то суммирование в этом выходе реализуется деревом из элементов сложения и элементом умножения на  $-1$ . На рис. 8.1в изображена полученная в результате схема для рассматриваемого примера.

Согласно аргументации из леммы 8.1, число двухходовых элементов в полученной схеме и в исходной отличается на  $m-n$  (оно равно разности между числом ребер и числом вершин в графе, исключая входы; при транспонировании изменяется множество входов). Множество элементов умножения на нетривиальные константы в обеих схемах одно и то же. Дополнительно используется не более  $n$  элементов умножения на  $-1$  (по числу выходов).

На самом деле, как можно проверить, в результате процедуры подъема отрицательных меток хотя бы один выход получает входящее ребро с меткой  $1$ . Поэтому требуется не более  $n-1$  дополнительных умножений на  $-1$ .  $\square$

- Оценка леммы 8.6 неулучшаема, как показывает пример преобразования

$$(x, y_1, \dots, y_m) \rightarrow (x - y_1, \dots, x - y_m).$$

**Теорема 8.3** ([129]). Для произвольной рациональной функции  $f \in R(X)$ , где  $R$  — кольцо с единицей, выполнено  $C_{\mathcal{A}_D^R}(f, \nabla f) \leq 4C_{\mathcal{A}_D^R}(f)$ .

► Рассмотрим минимальную схему  $S$  для функции  $f(x_1, \dots, x_n)$ . Введем новые формальные переменные  $dx_1, \dots, dx_n$  — дифференциалы переменных — и построим схему  $S'$  для дифференциала функции  $f$ ,

$$df = \frac{\partial f}{\partial x_1} dx_1 + \dots + \frac{\partial f}{\partial x_n} dx_n.$$

Схема  $S'$  строится параллельно схеме  $S$  индуктивно от входов к выходам. Переменной  $x_i$  соответствует дифференциал  $dx_i$  (база индукции). Теперь, если очередной элемент схемы  $S$  выполняет операцию  $u \circ v$ , и дифференциалы  $du, dv$  уже вычислены, то  $d(u \circ v)$  вычисляется по правилам:

$$d(u \pm v) = du \pm dv, \quad d(uv) = u \cdot dv + v \cdot du, \quad d(u/v) = (du - (u/v) \cdot dv)/v. \quad (8.12)$$



Фолкер Штрассен  
Цюрихский университет,  
с 1968 по 1988

<sup>4)</sup> В вырожденном случае, когда в вершину входит единственное ребро веса 1, вставляется пустое дерево, т. е. вершина вместе с входящим в нее ребром просто удаляются из схемы.

В случае скалярного умножения на  $a$  имеем просто  $d(au) = a \cdot du$ .

Если рассматривать  $S'$  как схему, построенную на входах  $dx_1, \dots, dx_n$  в базисе  $\mathcal{A}_D^{R(X)}$  (т.е. в качестве констант допускаются всевозможные функции из  $R(X)$ ), то по построению ее сложность не превосходит  $3C(S)$  в силу (8.12).

Поскольку дифференциал и градиент как линейные операторы над  $R(X)$  переходят друг в друга при транспонировании<sup>5)</sup>, с помощью леммы 8.6 получаем

$$C_{\mathcal{A}_L^{R(X)}}(\nabla f) \leq C_{\mathcal{A}_L^{R(X)}}(df) \leq 3C(S),$$

причем схема для градиента использует ровно те «константы», что и схема  $S'$ . Все необходимые константы вычисляются схемой  $S$ , см. (8.12), поэтому после присоединения схемы  $S$  к построенной схеме получаем требуемый результат:  $C_{\mathcal{A}_D^R}(f, \nabla f) \leq 4C(S)$ . ■

Метод [129] позволяет установить связь между сложностью обращения матрицы и вычисления ее определителя.

**Следствие 8.3** ([129]). *Пусть  $A = (a_{ij})$  — невырожденная матрица размера  $n \times n$ . Тогда сложности вычисления обратной матрицы  $A^{-1}$  и определителя  $\det A$  как функций коэффициентов  $a_{ij}$  связаны соотношением*

$$C_{\mathcal{A}_D^R}(A^{-1}) \leq 4C_{\mathcal{A}_D^R}(\det A) + n^2.$$

▷ Обозначим  $A^{-1} = (b_{ij})$ . Тогда, согласно правилу Крамера,

$$b_{ij} = \frac{1}{\det A} \frac{\partial \det A}{\partial a_{ji}}.$$

Таким образом, по теореме 8.3

$$C(A^{-1}) \leq C(\det A, \nabla \det A) + n^2 \leq 4C(\det A) + n^2.$$

□

- Фактически метод быстрого вычисления градиента был предложен ранее С. Линнайнмаа [257]. Независимо результат теоремы 8.3 с менее точной оценкой сложности получен в работе [22].

То, что обе задачи — обращение матрицы и вычисление определителя — по порядку не сложнее умножения матриц, доказано Ф. Штрассеном в [316].

Автором в [82] получена (вообще говоря) более сильная, чем в теореме 8.3, оценка

$$C_{\mathcal{A}_D^R}(f, \nabla f) \leq 3C_{\mathcal{A}_D^R}(f) + n, \tag{8.13}$$

где  $n$  — число аргументов функции  $f$ . Доказательство опирается на более симметричное вычисление дифференциалов мультилипликативных операций:

$$d(uv) = (du/u + dv/v) \cdot (uv), \quad d(u/v) = (du/u - dv/v) \cdot (u/v).$$

При этом, в отличие от метода [129], в схемах, построенных методом [82], множество функций, на которые выполняется деление, может отличаться от аналогичного множества в исходной

<sup>5)</sup>На единственный вход транспонированной схемы подается константа  $1 \in R$ .

схеме, вычисляющей функцию  $f$ . Неравенство (8.13) справедливо и в случае различного веса аддитивных и мультипликативных операций [82].

Теорема 8.3 сохраняет силу и в базисе без деления  $\mathcal{A}^R$ . Естественно, при этом речь идет о сложности многочленов. Оценка (8.13) в такой ситуации неприменима.

Э. Калтофен и М. Зингер [235] заметили, что оценка сложности теоремы 8.3 достигается вместе с сохранением порядка глубины исходной схемы. То же верно и для оценки (8.13), см. [82].

# Глава 9

## Вероятностный метод

$P$

В этой главе речь пойдет о методах синтеза, опирающихся на вероятностные рассуждения. При этом обычно доказывается, что среди множества схем определенного вида найдется та, которая реализует нужную функцию.

### Монотонные формулы для симметрических пороговых функций

$P$

Напомним, что через  $T_n^k$  обозначается симметрическая пороговая функция  $n$  переменных с порогом  $k$ :  $T_n^k(x_1, \dots, x_n) = (x_1 + \dots + x_n \geq k)$ . В случае  $k = 1$  тривиально выполняется  $\Phi_{\mathcal{B}_M}(T_n^1) = n$ . Функция с порогом 2 легко вычисляется методом деления пополам. Пусть  $X = (X_1, X_2)$ ,  $|X_i| = n_i$ . Справедлива формула [26]

$$T_{n_1+n_2}^2(X) = T_{n_1}^1(X_1) \cdot T_{n_2}^1(X_2) \vee T_{n_1}^2(X_1) \vee T_{n_2}^2(X_2). \quad (9.1)$$

**Теорема 9.1** ([32]).  $\Phi_{\mathcal{B}_M}(T_n^2) \leq n \lfloor \log_2 n \rfloor + 2(n - 2^{\lfloor \log_2 n \rfloor}) \sim n \log_2 n$ .

► Применяем (9.1) рекурсивно, разбивая множество переменных пополам. ■

- Р. Е. Кричевский [32] вместе с верхней оценкой теоремы 9.1 получил и нижнюю оценку  $\Phi_{\mathcal{B}_M}(T_n^2) \geq n \log n$ . На самом деле, как позже установили Дж. Радхакришнан [291] и С. А. Ложкин [36] (по модулю структурного результата Кричевского  $\Phi_{\mathcal{B}_0}(T_n^2) = \Phi_{\mathcal{B}_M}(T_n^2)$  [32]), оценка теоремы 9.1 является точной даже в классе формул над базисом  $\mathcal{B}_0$ .

При постоянных  $k \geq 3$  конструктивные методы синтеза не столь точны, однако можно доказать существование сравнительно простых схем. Следующий результат принадлежит Л. С. Хасину [102].

**Теорема 9.2** ([102]). *При постоянном  $k \geq 2$  справедливо  $\Phi_{\mathcal{B}_M}(T_n^k) \leq n \log n$ .*

► Пусть  $k \mid n$  и  $(X_i^1, \dots, X_i^k)$  — случайные разбиения множества переменных  $X$  на равномощные подмножества<sup>1)</sup>  $X_i^j$ ,  $i \in \mathbb{N}$ . Рассмотрим случайную функцию

$$f(X) = \bigvee_{i=1}^t H_i, \quad H_i = T_{n/k}^1(X_i^1) \cdot T_{n/k}^1(X_i^2) \cdot \dots \cdot T_{n/k}^1(X_i^k). \quad (9.2)$$

Функция не имеет импликант длины  $< k$ , поэтому  $f(X) \geq T_n^k(X)$ . Вероятность того, что  $H_i$  содержит некоторую импликанту  $I = x_{j_1} \cdot \dots \cdot x_{j_k}$ , оценим как

$$\mathbf{P}(H_i \geq I) = k! \mathbf{P}(x_{j_1} \in X_i^1, \dots, x_{j_k} \in X_i^k) = k! \frac{n/k}{n} \cdot \frac{n/k}{n-1} \cdot \dots \cdot \frac{n/k}{n-k+1} \geq \frac{k!}{k^k} > \frac{1}{e^k}.$$

Теперь вероятность того, что  $f$  не содержит импликанту  $I$ , можно оценить как

$$\mathbf{P}(f \not\geq I) = \prod_{i=1}^t \mathbf{P}(H_i \not\geq I) \leq (1 - e^{-k})^t.$$

При  $t \approx e^k k \ln n$  в силу неравенства  $(1 - 1/x)^x < 1/e$  при  $x > 1$ , имеем

$$\mathbf{P}(f \neq T_n^k) = \mathbf{P}(\exists_I f \not\geq I) \leq C_n^k \mathbf{P}(f \not\geq I) < C_n^k / n^k \leq 1.$$

Как следствие,  $\mathbf{P}(f = T_n^k) > 0$ . Поэтому некоторая формула вида (9.2) реализует функцию  $T_n^k$  со сложностью  $tn \asymp n \log n$ . ■

- Оценка теоремы 9.2 по порядку точна,  $\Phi_{\mathcal{B}_M}(T_n^k) \asymp n \log n$ , например, ввиду простой нижней оценки  $\Phi_{\mathcal{B}_M}(T_n^k) \geq \Phi_{\mathcal{B}_M}(T_{n-k+2}^2)$  [102].

Конструктивно для симметрических функций с малыми порогами доказывается верхняя оценка  $\Phi_{\mathcal{B}_M}(T_n^k) \leq n \log n (k^2/2)^{\log^* n}$  в работе М. Клеймана и Н. Пиппенджера [240].

### Монотонные формулы для функции голосования $[P] \boxed{\varepsilon}$

Из доказательства теоремы 9.2 видно, что с ростом  $k$  сложность метода растет экспоненциально, и для реализации пороговых функций с большими порогами, в частности, для функции голосования  $\text{maj}_n = T_n^{n/2}$ , метод не годится. Изящное решение проблемы нашел Л. Вэльянт [325]. Пусть  $\alpha = \frac{3-\sqrt{5}}{2} \approx 0.38$ .

**Теорема 9.3** ([325]).  $D_{\mathcal{B}_M}(\text{maj}_n) \lesssim 2(1 + \log_{4\alpha} 2) \log_2 n < 5.28 \log_2 n$ .

► На самом деле, мы будем доказывать чуть более слабую оценку  $D_{\mathcal{B}_M}(\text{maj}_n) \lesssim 2(1 + \log_\beta 2) \log_2 n$  при произвольном постоянном  $\beta \in (1, 4\alpha)$ .

---

<sup>1</sup>Более формально, мы рассматриваем равномерное распределение на множестве всех перестановок  $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$  и полагаем

$$X^j = (x_{\pi((j-1)n/k+1)}, x_{\pi((j-1)n/k+2)}, \dots, x_{\pi(jn/k)}), \quad 1 \leq j \leq k.$$



Лесли Гэбриэл

Вэльянт

Гарвардский университет,  
с 1982

формулы:

$$p_k = \mathbf{P}(G(X_0) = 1 \mid G \in \Delta_k), \quad q_k = \mathbf{P}(G(X_1) = 0 \mid G \in \Delta_k).$$

По построению,

$$p_{k+1} = (1 - (1 - p_k)^2)^2, \quad q_{k+1} = 1 - (1 - q_k^2)^2. \quad (9.3)$$

Легко проверить, что последовательности  $\{p_k\}$  и  $\{q_k\}$  монотонно убывают на интервалах  $(0, \alpha)$  и  $(0, 1-\alpha)$  соответственно (края интервалов являются корнями уравнений  $x = (1 - (1 - x)^2)^2$  и  $x = 1 - (1 - x^2)^2$ ).

**Лемма 9.1.** *При некотором постоянном  $\gamma > 0$  и некотором  $t \leq \log_\beta n$  выполнено*

$$p_t < \alpha - \gamma, \quad q_t < 1 - \alpha - \gamma.$$

▷ При  $t = 0$  вероятности ошибки равны<sup>2)</sup>

$$\begin{aligned} p_0 &= \mathbf{P}(G \equiv x_i \mid x_{0,i} = 1, G \in \Delta_0) = \alpha - \frac{\alpha}{n}, \\ q_0 &= \mathbf{P}(G \equiv 0 \mid G \in \Delta_0) + \mathbf{P}(G \equiv x_i \mid x_{1,i} = 0, G \in \Delta_0) = 1 - \alpha - \frac{\alpha}{n}. \end{aligned}$$

Полагая  $\varepsilon_k = p_k - \alpha$ , с учетом  $(1 - \alpha)^2 = \alpha$  из (9.3) получаем

$$\begin{aligned} p_{k+1} &= (1 - (1 - \alpha + \varepsilon_k)^2)^2 = \\ &\quad \alpha - 4\alpha\varepsilon_k + ((2(1 - \alpha) + \varepsilon_k)^2 - 2(1 - \alpha)) \varepsilon_k^2 \leq \alpha - \beta\varepsilon_k \quad (9.4) \end{aligned}$$

при условии  $\varepsilon_k \leq \gamma_1$ . Как следствие, при некотором  $t_1 = \log_\beta n - O(1)$  имеем  $\varepsilon_{t_1} \geq \beta^{t_1} \varepsilon_0 > \gamma_1$ .

<sup>2)</sup>Через  $x_{0,i}$  и  $x_{1,i}$  обозначены компоненты векторов  $X_0$  и  $X_1$ .

Аналогично, обозначая  $\delta_k = q_k - (1 - \alpha)$ , получаем

$$\begin{aligned} q_{k+1} &= 1 - (1 - (1 - \alpha - \delta_k))^2 = \\ &= 1 - \alpha - 4\alpha\delta_k - ((2(1 - \alpha) - \delta_k)^2 - 2(1 - \alpha))\delta_k^2 \leq 1 - \alpha - \beta\delta_k, \end{aligned} \quad (9.5)$$

при условии  $\delta_k \leq \gamma_2$ . Как следствие, при некотором  $t_2 = \log_\beta n - O(1)$  выполнено  $\delta_{t_2} \geq \beta^{t_2}\varepsilon_0 > \gamma_2$ .

Остается выбрать  $t = \max\{t_1, t_2\}$  и  $\gamma = \min\{\gamma_1, \gamma_2\}$ .  $\square$

**Лемма 9.2.** *Пусть  $p_t < \alpha - \gamma$  и  $q_t < 1 - \alpha - \gamma$  при постоянном  $\gamma > 0$ . Тогда при некотором  $u = \log_2 n + O(1)$  выполнено  $p_{t+u}, q_{t+u} < 1/2^{n+1}$ .*

$\triangleright$  I. Сначала покажем, что  $p_{t+u'}, q_{t+u'} \leq 1/8$  при подходящем  $u' = O(1)$ . При этом можем полагать, что  $\gamma$  достаточно мало, скажем,  $\gamma \leq 0.3$ . Из (9.3) следует

$$p_{k+1} = p_k^2(2 - p_k)^2, \quad q_{k+1} = q_k^2(2 - q_k^2). \quad (9.6)$$

Легко проверить, что функции  $p(x) = x - x^2(2 - x)^2$  и  $q(x) = x - x^2(2 - x^2)$  на отрезках  $I_p = [1/8, \alpha - \gamma]$  и соответственно  $I_q = [1/8, 1 - \alpha - \gamma]$  выпуклы вверх, следовательно, принимают минимальные значения на концах (эти значения положительны). Следовательно, существует  $\tau > 0$ , при котором  $\tau \leq \min_{x \in I_p} p(x), \min_{x \in I_q} q(x)$ . Тогда  $p_{k+1} \leq p_k - \tau$ , как только  $p_k \in I_p$  и  $q_{k+1} \leq q_k - \tau$  при  $q_k \in I_q$ . Поэтому можно выбрать  $u' = (1 - \alpha - \gamma - 1/8)/\tau$ .

II. Из (9.6) видно, что  $p_{k+1} \leq 4p_k^2$  и  $q_{k+1} \leq 2q_k^2$ . Поэтому, отталкиваясь от  $p_{t+u'}, q_{t+u'} \leq 1/8$ , выводим

$$p_{t+u'+i} \leq 2^{-(2^{i+2})}, \quad q_{t+u'+i} \leq 2^{-(2^{i+1}+1)}.$$

Таким образом, при  $i = \lceil \log_2 n \rceil$  получаем требуемые оценки.  $\square$

Объединяя результаты лемм 9.1 и 9.2, находим, что с вероятностью  $< 1/2^{n+1}$  формула из  $\Delta_{t+u}$  неверно вычисляет функцию  $\text{maj}_n$  на наборе  $X_0$ , и то же для набора  $X_1$ . Как следствие, с вероятностью  $> 1/2$  формула из  $\Delta_{t+u}$  (имеющая глубину  $2(t + u)$ ) вычисляет функцию  $\text{maj}_n$ .  $\blacksquare$

**Следствие 9.1.**  $\Phi_{B_M}(\text{maj}_n) \prec n^{5.28}$ .

- Для доказательства теоремы 9.3 в исходной формулировке требуется усиление леммы 9.1. Например, можно показать, что при  $t \leq \log_{4\alpha}(n/8)$  выполнено

$$p_t < \alpha - \frac{(4\alpha)^t}{4n}, \quad q_t < 1 - \alpha - \frac{(4\alpha)^t}{3n}. \quad (9.7)$$

$\triangleright$  Действительно, из (9.4) в силу  $\varepsilon_k \leq \alpha$  вытекает

$$4\alpha\varepsilon_k \geq \varepsilon_{k+1} \geq 4\alpha\varepsilon_k - ((2 - \alpha)^2 - 2(1 - \alpha))\varepsilon_k^2 > 4\alpha(1 - \varepsilon_k)\varepsilon_k.$$

Как следствие, имеем

$$\varepsilon_t \geq 4\alpha(1 - \varepsilon_{t-1})\varepsilon_{t-1} \geq (4\alpha)^t\varepsilon_0 \prod_{i=0}^{t-1} (1 - \varepsilon_i) \geq (4\alpha)^t\varepsilon_0 \prod_{i=0}^{t-1} (1 - (4\alpha)^i\varepsilon_0). \quad (9.8)$$

Используя справедливое при  $0 \leq x \leq 1/2$  неравенство  $\ln(1-x) \geq -2x$ , с учетом  $\varepsilon_0 = \alpha/n$  и  $(4\alpha)^t \leq n/8$ , произведение в (9.8) можно оценить как

$$\prod_{i=0}^{t-1} (1 - (4\alpha)^i \varepsilon_0) \geq e^{-2\varepsilon_0(1+(4\alpha)+\dots+(4\alpha)^{t-1})} \geq e^{-2\varepsilon_0(4\alpha)^t/(4\alpha-1)} \geq e^{-\alpha/(16\alpha-4)}.$$

Так получаем первое неравенство в (9.7):  $\varepsilon_t \geq (4\alpha)^t \varepsilon_0 e^{-\alpha/(16\alpha-4)} > (4\alpha)^t / (4n)$ .

Аналогичным образом, из (9.5) следует

$$\delta_{k+1} \leq 4\alpha\delta_k + ((2(1-\alpha))^2 - 2(1-\alpha))\delta_k^2 < 4\alpha(1+\delta_k/4)\delta_k, \quad (9.9)$$

а при дополнительном предположении  $\delta_k \leq \alpha/4$  — также

$$\delta_{k+1} \geq 4\alpha\delta_k + ((2(1-\alpha) - \alpha/4)^2 - 2(1-\alpha))\delta_k^2 > 4\alpha\delta_k.$$

Тем самым, верно второе неравенство в (9.7):  $\delta_t \geq (4\alpha)^t \delta_0 > (4\alpha)^t / (3n)$ , если только  $\delta_{t-1} \leq \alpha/4$ .

Докажем по индукции, что  $\delta_k < 2(4\alpha)^k \delta_0$  при  $k \leq t$ . Как следствие, получим требуемое неравенство  $\delta_t \leq \alpha/4$ . При  $i = 0$  доказывать нечего. Проверим индуктивный переход от  $k-1$  к  $k$ . Согласно (9.9), предположению индукции, неравенствам  $1+x \leq e^x$  и  $(4\alpha)^k \leq n/8$ ,

$$\begin{aligned} \delta_k &\leq 4\alpha(1+\delta_{k-1}/4)\delta_{k-1} \leq (4\alpha)^k \delta_0 \prod_{i=0}^{k-1} (1+\delta_i/4) \leq (4\alpha)^k \delta_0 \prod_{i=0}^{k-1} (1+(4\alpha)^i \delta_0/2) \leq \\ &(4\alpha)^k \delta_0 e^{(1+4\alpha+\dots+(4\alpha)^{k-1})\delta_0/2} \leq (4\alpha)^k \delta_0 e^{\delta_0(4\alpha)^k/(8\alpha-2)} \leq (4\alpha)^k \delta_0 e^{\alpha/(64\alpha-16)} < 2(4\alpha)^k \delta_0. \end{aligned}$$

□

Р. Боппана [150] заметил, что метод Вэльянта для произвольной пороговой функции влечет оценку  $\Phi_{B_M}(T_n^k) \leq k^{4.28} n \log n$  (это лучше, чем в методе теоремы 9.2). Он же установил, что используемая в доказательстве порождающая формула  $(G_1 \vee G_2)(G_3 \vee G_4)$  оптимальна среди бесповторных; в более строгой форме это утверждение доказано М. Дубинером и У. Цви ком [182]. В 3-местном монотонном базисе  $\{\text{maj}_3\}$  А. Гупта и С. Махаджан [208] получили аналогичным методом оценку  $\Phi_{\{\text{maj}_3\}}(\text{maj}_n) \leq n^{4.3}$ .

Метод Вэльянта доказуемо эффективен в построении приближений пороговых функций. Обозначим через  $\Psi_n^k$  класс монотонных функций, принимающих нулевые значения на наборах веса  $\leq k$ , и единичные значения — на наборах веса  $\geq n-k$ . Методом теоремы 9.3 для любого постоянного  $\alpha \in (0, 1/2)$  строится формула сложности  $O(n^2)$ , реализующая некоторую функцию из  $\Psi_n^{\alpha n}$ . Сложность формулы оптимальна по порядку ввиду доказанной Э. Муром и К. Шенноном [270] нижней оценки  $\Phi_{B_M}(f) \geq (k+1)^2$  для произвольной функции  $f \in \Psi_n^k$ .

Отметим, что известные нижние оценки сложности для функции голосования имеют вид  $\Phi_{B_M}(\text{maj}_n) \geq n^2$  [270] (что также следует из  $\Phi_{B_0}(\text{maj}_n) \geq n^2$  [104]) и  $\Phi_{\{\text{maj}_3\}}(\text{maj}_n) \geq n^{\log_2 3}$  [71].

В нескольких работах предпринимались попытки осуществить дерандомизацию метода. Например, С. А. Ложкин и А. А. Семенов [40] указали на возможность построения формулы сложности  $L \asymp k^{6.28} n \log n$  для функции  $T_n^k$  за время порядка  $L \cdot \log n$ .

Напомним, что в расширенном до полного базисе лучшие известные оценки [88] вытекают из (7.9) и оказываются ненамного сильнее:  $\Phi_{B_0}(\text{maj}_n) \prec n^{3.91}$ ,  $D_{B_0}(\text{maj}_n) \lesssim 4.14 \log_2 n$ .

## Параллельные схемы для логического перманента $[P][A]$

*Логический перманент* порядка  $n$  — это функция  $n^2$  булевых переменных:

$$PERM_n(X) = \bigvee_{\pi} x_{1,\pi(1)} \cdot x_{2,\pi(2)} \cdot \dots \cdot x_{n,\pi(n)}, \quad (9.10)$$

где дизъюнкция берется по всем перестановкам  $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ . Перманент выражает наличие совершенного паросочетания в двудольном графе с долями из  $n$  вершин.

По существу, перманент является монотонной версией определителя (матрицы переменных  $X$ ). Неудивительно, что в излагаемом далее способе вычисления перманента используется сведение к вычислению определителя. Возьмем за основу простую схему для определителя, предложенную А. Л. Чистовым [167] (см. также [145]), в модификации Н. Пиппенджера [286]. Пусть  $+_n$  означает операцию сложения  $n$  аргументов. Рассмотрим базис  $\mathcal{A}_n = \{*, -, +_{n+1}\}$ , в котором сложность операции  $+_{n+1}$  полагается равной  $n$ .

**Теорема 9.4** ([167, 145, 286]). *Определитель порядка  $n$  над полем  $F$  может быть вычислен схемой в базисе  $\mathcal{A}_n^F$  сложности  $O(n^4 \log n)$  и глубины  $O(\log n)$ .*

Как следствие, для схемы над стандартным арифметическим базисом  $\mathcal{A}^F$  получаем оценки сложности  $O(n^4 \log n)$  и глубины  $O(\log^2 n)$ .

► Для эффективного вычисления определителя достаточно подобрать подходящее алгебраическое выражение. Обозначим через  $A_k$  главную подматрицу<sup>3)</sup> размера  $k \times k$  матрицы  $A$ . Согласно правилу Крамера, если матрица  $A_k$  — обратима, то

$$\det A_{k-1} / \det A_k = (A_k^{-1})[k, k],$$

откуда для матрицы  $A \in F^{n \times n}$  получаем

$$1 / \det A = \prod_{k=1}^n (A_k^{-1})[k, k]. \quad (9.11)$$

Заметим, что определитель  $\det A$  является с точностью до множителя  $(-1)^n$  коэффициентом при  $x^n$  определителя матричного многочлена<sup>4)</sup>  $P(x) = I_n - xA$ . Используя (9.11) и тождество для формальных степенных рядов  $(1 - x)^{-1} = \sum_{i=0}^{\infty} x^i$  вместе с его матричным вариантом  $(I_k - xA_k)^{-1} = \sum_{i=0}^{\infty} (xA_k)^i$ , получаем формулу [286]

$$\begin{aligned} \det P(x) &= \left( \prod_{k=1}^n (P(x)_k^{-1})[k, k] \right)^{-1} = \prod_{k=1}^n \left( 1 + \sum_{i=1}^{\infty} (A_k)^i[k, k]x^i \right)^{-1} = \\ &= 1 + \sum_{j=1}^{\infty} \left( 1 - \prod_{k=1}^n \left( 1 + \sum_{i=1}^{\infty} (A_k)^i[k, k]x^i \right) \right)^j. \end{aligned} \quad (9.12)$$

Поскольку нас интересует коэффициент при  $x^n$ , суммы в правой части (9.12) можно ограничить первыми  $n$  слагаемыми.

Вычисления выполняются по формуле (9.12). Предварительно заметим, что в базисе  $\mathcal{A}_n$  стандартным способом умножение  $n \times n$  матриц выполняется со

<sup>3</sup>Главная — левая верхняя угловая подматрица.

<sup>4</sup> $\det P(x)$  — это перевернутый характеристический многочлен матрицы  $A$ .

сложностью  $O(n^3)$  и глубиной 2, умножение матрицы на вектор — со сложностью  $O(n^2)$  и глубиной 2, а умножение многочленов по модулю  $x^{n+1}$  — со сложностью  $O(n^2)$  и глубиной 2.

Обозначим  $v_k = (0, \dots, 0, 1)$  — вектор длины  $k$ . Результатом умножения  $v_k$  на  $k \times k$  матрицу  $B$  является нижняя строка матрицы.

1) При любом  $k$  найдем  $v_k(A_k)^i$  для всех  $i \leq n$ .

Для этого вычисляем последовательно  $(A_k)^2, (A_k)^4, \dots, (A_k)^{2^{\lceil \log_2 n \rceil}}$ . Параллельно, по мере готовности множителей, для  $i = 2^h + j$  вычисляем строки  $v_k(A_k)^i = (v_k(A_k)^j)(A_k)^{2^h}$ . В результате, в частности, получены все коэффициенты  $(A_k)^i[k, k]$ .

Схема имеет (суммарно по всем  $k$ ) сложность  $O(n^4 \log n)$  и глубину  $O(\log n)$ .

2) Внутреннее произведение  $n$  многочленов в (9.12) по модулю  $x^{n+1}$  вычисляется со сложностью  $O(n^3)$  и глубиной  $O(\log n)$ .

3) Окончательно сумма вида  $\sum_{j=0}^n p^j(x)$  определяется как  $\prod_{j=0}^{\log_2 n} (1 + p^{2^j}(x))$  по модулю  $x^{n+1}$ . Эти вычисления сводятся к  $2 \log_2 n$  умножениям многочленов и поэтому реализуются схемой сложности  $O(n^2 \log n)$  и глубины  $O(\log n)$ . ■

Связь между перманентом  $PERM_n(X)$  и определителем устанавливает простое наблюдение [185]. Зададим коэффициенты  $n \times n$  матрицы  $T(Y)$  как  $T[i, j] = x_{i,j}y_{i,j}$ , где  $y_{i,j}$  — формальные переменные. (Эта матрица называется *матрицей Туттма*.) Тогда

$$PERM_n(X) = 1 \iff \det T(Y) \neq 0. \quad (9.13)$$

Л. Ловас [258] заметил, что указанный критерий позволяет установить существование простых параллельных схем для перманента, поскольку равенство нулю определителя проверяется по значениям в наборе случайных точек. Удобным инструментом для обоснования корректности метода служит лемма Дж. Шварца [306], которую мы докажем в ослабленной форме.

**Лемма 9.3** ([306]). *Любой мультилинейный многочлен  $f \in \mathbb{Z}[x_1, \dots, x_n]$ , не равный тождественно нулю, имеет на множестве  $\llbracket m \rrbracket^n$  не более  $nm^{n-1}$  корней.*

▷ Доказательство по индукции. Случай  $n = 1$  тривиален. Докажем переход от  $n - 1$  к  $n$ . Без ограничения общности, пусть  $f$  существенно зависит от переменной  $x_1$ . Запишем  $f(X) = x_1 q(x_2, \dots, x_n) + r(x_2, \dots, x_n)$ , где  $q \not\equiv 0$ . По индуктивному предположению, многочлен  $q$  принимает нулевые значения не более чем на  $(n-1)m^{n-2}$  наборах из  $\llbracket m \rrbracket^{n-1}$ . Для любого другого набора найдется не более одного значения  $x_1$ , обнуляющего многочлен  $f$ . Общее число корней многочлена  $f$  в  $\llbracket m \rrbracket^n$ , следовательно, не превосходит  $(n-1)m^{n-1} + m^{n-1}$ . □

Вероятно, впервые на существование параллельных схем полиномиальной сложности для перманента указали А. Бородин, И. фон Гатен и Дж. Хопкрофт [151]. Мы приводим более элементарное доказательство.

**Теорема 9.5.** *Функция  $PERM_n$  реализуется булевой схемой над конечным полным базисом со сложностью  $O(n^7 \log^2 n)$  и глубиной  $O(\log^2 n)$ .*

- При различных значениях набора переменных  $X$  всего получается менее  $2^{n^2}$  различных невырожденных матриц Тутта  $T(Y)$ . Положим  $m = n^2 2^{n^2}$ . Как следует из леммы 9.3, найдется целочисленная матрица  $Q \in \llbracket m \rrbracket^{n \times n}$ , для которой выполнено  $\det T(Q) \neq 0$ , если  $\det T(Y) \not\equiv 0$ .

Таким образом,  $\text{PERM}_n(X) = (\det T(Q) \neq 0)$ . Для понижения сложности расчетов воспользуемся модулярной арифметикой. Заметим, что  $|\det T(Q)| < n!m^n < n^{3n}2^{n^3}$  для любой матрицы Тутта  $T$ . Согласно асимптотическому закону распределения простых чисел (см., например, [294]), произведение первых  $s$  простых чисел  $p_1 \cdot \dots \cdot p_s$  превосходит  $2n^{3n}2^{n^3}$  при некотором  $s \asymp n^3/\log n$ . Поэтому  $\det T(Q) = 0$  равносильно выполнению  $\det T(Q) \equiv 0 \pmod{p_i}$  при всех  $i = 1, \dots, s$ .

Стандартным образом умножение в поле  $\mathbb{F}_p$  реализуется со сложностью  $O(\log^2 p)$  и глубиной  $O(\log \log p)$  как  $(ab \pmod p) = ab - p\lfloor abq \rfloor$ , где  $q$  — подходящее приближение числа  $1/p$ . Сумма  $n$  слагаемых в поле  $\mathbb{F}_p$  вычисляется со сложностью  $O(n \log p + \log^2 p)$  и глубиной  $O(\log(n \log p))$ , скажем, методом компрессоров с последующим приведением по модулю  $p$ .

Следовательно, согласно теореме 9.4 значение  $r_i = \det T(Q) \pmod{p_i}$  определяется схемой сложности  $O(n^4 \log^3 n)$  и глубины  $O(\log^2 n)$  при любом  $i \leq s$ . Заключительная проверка равенства нулю всех  $r_i$  реализуется тривиально со сложностью  $O(s \log n)$  и глубиной  $O(\log n)$ . ■

- Аккуратный анализ метода теоремы 9.4 позволяет получить схему глубины  $O(\log^2 n)$  и сложности порядка  $n^{\omega+1+o(1)}$  в базисе  $\mathcal{A}^F$ , где  $\omega < 2.38$  — экспонента сложности матричного умножения, см. [145]. Достоинством метода является отсутствие операций деления и слабые требования к кольцу коэффициентов (вместо поля можно взять коммутативное кольцо с единицей). В базисе с делением сложность вычисления определителя матриц над числовыми полями (при сохранении порядка глубины) может быть понижена до  $O(n^{2.78})$  методом З. Галила и В. Я. Пана [198], если подставить современные оценки сложности умножения прямоугольных матриц [121]. Напомним, что при отсутствии ограничений на глубину определитель можно вычислить со сложностью  $n^{\omega+o(1)}$  [316].

Вопрос о возможности реализации определителя и перманента с глубиной  $o(\log^2 n)$  пока открыт, неизвестны и явные конструкции схем для перманента сложности  $n^{O(1)}$  и глубины  $\log^{O(1)} n$ . Далее (теорема 11.4) мы рассмотрим альтернативный подход к вычислению перманента схемами без ограничения на глубину.

### Сложность линейных операторов с плотными матрицами $\boxed{P}/_2$

Для любой булевой матрицы  $A$  тривиально выполняется  $L(A) \leq |A|$ . В частности,  $n \times n$  матрица с малым числом единиц,  $|A| \asymp n$ , имеет линейную сложность  $L(A) \asymp n$ . Менее тривиальным оказывается вопрос о сложности матриц с малым числом нулей. Верно ли, что  $L(A) \leq |\bar{A}|$ ? Метод работы российских математиков [248] фактически дает утвердительный ответ на этот вопрос.

**Теорема 9.6.** Для любой булевой  $n \times n$  матрицы  $A$  веса  $n^2 - qn$ , где  $1 \leq q \leq n \log^{-4} n$ , выполнено  $L(A) \leq qn$ .

- Нули разбивают каждую строку матрицы на *интервалы* — под интервалом мы понимаем подстроку из единиц в последовательно идущих столбцах. Интервалной суммой назовем сумму переменных с последовательными индексами

$x_i + x_{i+1} + \dots + x_j$ . Нам понадобится вспомогательная лемма о сложности вычисления интервальных сумм, которая доказана Н. Алоном и Б. Шибером [124].

**Лемма 9.4** ([124]). *Любую интервальную сумму на множестве  $n$  переменных можно записать в виде  $u + v$ , где все необходимые суммы  $u, v$  вычисляются аддитивной схемой сложности  $n \log n$ .*

▷ Разделим множество переменных пополам, с младшими и со старшими индексами. Со сложностью  $O(n)$  вычислим префиксные суммы переменных со старшими индексами и суффиксные суммы переменных с младшими индексами. Любая интервальная сумма со слагаемыми из обеих половин представима в виде суммы некоторых суффикса и префикса из уже вычисленного. Для реализации сумм, целиком лежащих внутри каждой из половин, воспользуемся рекурсией. Сложность схемы  $T(n)$  удовлетворяет соотношению  $T(n) \leq 2T(n/2) + O(n)$ , откуда  $T(n) \leq n \log n$ .  $\square$

При подходящей перестановке строк матрицы  $A$  можно представить в виде  $A = \begin{bmatrix} A_0 \\ A_1 \end{bmatrix}$ , где каждая строка подматрицы  $A_0$  содержит  $\leq q \log n$  нулей, а каждая строка подматрицы  $A_1$  — более  $q \log n$  нулей. По условию, матрица  $A_1$  имеет размер  $O(n/\log n) \times n$ .

Строки матрицы  $A_1^T$  распадаются на  $O(qn)$  интервалов. Поэтому согласно лемме 9.4,  $L(A_1^T) \leq qn$ , следовательно, в силу принципа транспонирования (лемма 8.1),  $L(A_1) \leq qn$ .

Рассмотрим матрицу  $A_0$ . Строки матрицы  $A_0$  также распадаются на  $O(qn)$  интервалов. Разобьем матрицу на вертикальные полосы ширины  $l = \log n$ . Интервалы, которые целиком лежат внутри одной полосы, назовем *простыми*, остальные — *составными*.

*I.* Покажем, что любую составную интервальную сумму можно записать в виде  $t+u+v+w$ , где все промежуточные суммы  $t, u, v, w$  вычисляются аддитивной схемой сложности  $(n/l) \log(n/l) + O(n)$ .

Для этого в каждой полосе вычислим префиксные и суффиксные суммы с общей сложностью  $O(n)$ , включая суммы всех переменных полосы (групповые суммы). Групповые суммы (их  $n/l$ ) подадим на входы схемы из леммы 9.4: получим компоненты  $u, v$  для интервальных групповых сумм. Сложность схемы  $(n/l) \log(n/l)$ . Осталось заметить, что произвольная составная интервальная сумма, отличная от групповой суммы, неизбежно пересекает несколько полос и поэтому представляется как сумма суффикса  $t$ , префикса  $w$  и интервальной групповой суммы  $u + v$ .

Как следствие, все составные интервальные суммы для матрицы  $A_0$  вычисляются со сложностью  $(n/l) \log(n/l) + O(qn)$ . Если обозначить через  $s$  суммарное число полос с простыми интервалами по всем строкам матрицы  $A_0$ , то имеем  $L(A_0) \leq (n/l) \log(n/l) + O(qn) + sl$ .

*II.* Центральным местом доказательства является следующее наблюдение: всегда найдется перестановка столбцов матрицы  $A_0$ , при которой число  $s$  достаточно мало. Рассуждение проводится вероятностным методом.

Пусть строка содержит  $r$  нулей. Число перестановок элементов строки, при которых в конкретной полосе есть простой интервал, грубо оценим сверху как  $C_{l+2}^2 C_n^{r-2}$  (первый множитель оценивает число способов задания границ интервала, второй — число способов разместить остальные  $r - 2$  нулей).

Тогда математическое ожидание числа полос с простыми интервалами в одной строке по порядку не превосходит

$$(n/l) \frac{C_{l+2}^2 C_n^{r-2}}{C_n^r} \leq \frac{n l r (r-1)}{(n-r)(n-r+1)} \leq \frac{q^2 \log^3 n}{n}$$

с учетом  $l = \log n$  и  $r \leq q \log n$ . Следовательно,  $\mathbf{E}[s] \leq q^2 \log^3 n$ . Таким образом, с ненулевой вероятностью  $s \leq q^2 \log^3 n$ . Окончательно получаем

$$\mathsf{L}(A) \leq \mathsf{L}(A_0) + \mathsf{L}(A_1) \leq qn + q^2 \log^4 n \leq qn.$$

■

В частности, в наиболее интересном случае  $q = O(1)$  имеет место (это основной результат работы [248])

**Следствие 9.2** ([248]). Для любой булевой  $n \times n$  матрицы  $A$  веса  $n^2 - O(n)$  выполнено  $\mathsf{L}(A) \leq n$ .

- Мощностная нижняя оценка показывает, что результат теоремы 9.6 точен по порядку при  $\log(n/q) \asymp \log n$ . Авторы [248] также предложили конструктивный, но более длинный путь доказательства п. II.

Общую задачу оценки аддитивной сложности класса булевых  $m \times n$  матриц заданного веса  $\alpha m n$ ,  $0 < \alpha < 1$ , поставил Э. И. Нечипорук в [55, 57]. Он установил порядок сложности (более точно с асимптотической точки зрения, чем дает теорема 9.6) при некоторых соотношениях между  $\alpha$ ,  $m$  и  $n$ .

# Глава 10

## Метод массового производства m

Идея массового производства работает, когда стоимость (сложность) единицы продукции может быть понижена при изготовлении нескольких подобных единиц. В настоящей главе мы рассматриваем с одной стороны ситуации, в которых это возможно в принципе, а с другой — ситуации, в которых массовое производство приносит полезный результат.

### Групповые линейные преобразования m

Рассмотрим задачу применения линейного преобразования  $AX$  к нескольким независимым (переменным) векторам  $X_1, \dots, X_m$ . В модели монотонных аддитивных схем очевидно выполнено  $L(AX_1, \dots, AX_m) = m \cdot L(AX)$ . Однако при выполнении вычислений в поле часто можно получить лучшую оценку, используя быстрое умножение матриц. Следующий пример взят из работы Б. Пауля [282]<sup>1</sup>.

**Теорема 10.1.** Пусть  $\mathbb{F}$  — поле,  $A \in \mathbb{F}^{n \times n}$  и  $S$  — билинейная схема умножения матриц размера  $n \times n$  и  $n \times m$  над  $\mathbb{F}$ . Тогда  $C_{\mathcal{A}_L^{\mathbb{F}}}(AX_1, \dots, AX_m) \leq C(S)$ .

► Доказательство тривиально. Вектора  $X_1, \dots, X_m$  организуются в матрицу переменных  $\mathbf{X}$ . Для вычисления произведения  $AX$  используем схему  $S$ , в которой, поскольку один из входов (матрица  $A$ ) постоянный, нескалярные умножения превращаются в скалярные (важно, что схема  $S$  билинейна). ■

Используя известную оценку сложности умножения матриц [178, 121], получаем

**Следствие 10.1.** Пусть  $A \in \mathbb{B}^{n \times n}$ . Тогда  $L_{\oplus}(I_n \otimes A) \leq n^{2.38}$ .

- Следствие указывает на существование булевой матрицы вида  $A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$ , которая вычисляются проще, чем составляющие ее матрицы  $A_1$  и  $A_2$ :  $L_{\oplus}(A) < L_{\oplus}(A_1) + L_{\oplus}(A_2)$ . Достаточно рассмотреть матрицу  $I_n \otimes A$  с условием  $L_{\oplus}(A) \sim n^2 / \log_2 n$ . Используя результаты о быстром

<sup>1</sup> В работе [282] он формулируется применительно к реализации линейных преобразований над  $\{\mathbb{B}, \vee\}$  схемами в полном базисе.

умножении на узкие прямоугольные матрицы, например, [175], можно убедиться, что экономия при совместной реализации матриц может достигать двух раз:  $\sup_{L(A)>0} \frac{L(A_1)+L(A_2)}{L(A)} = 2$ .

### Вычисление булевой функции на нескольких входных наборах $m \times c$

В свете того, что путь к оптимальной реализации случайной булевой функции лежит через вычисление линейного оператора (теорема 3.2), естественно ожидать, что также можно ускорить вычисление значений функции на нескольких входных наборах. Результат Д. Улига [100] подтверждает это предположение (мы доказываем его в ослабленной форме).

**Теорема 10.2** ([100]). *Пусть  $X, Y$  — непересекающиеся группы  $n$  булевых переменных. Тогда для произвольной булевой функции  $f$*

$$C_{B_2}(f(X), f(Y)) \lesssim 2^n/n.$$

► Разложим функцию  $f(X)$  по первым  $r$  переменным  $X_1$ :

$$f(X_1, X_2) = \bigoplus_{\sigma \in \mathbb{B}^r} X_1^\sigma \cdot f_\sigma(X_2), \quad X_1^\sigma = \prod_{i=1}^r x_{1,i}^{\sigma_i}.$$

Пусть в нашем распоряжении имеются схемы, независимо вычисляющие  $2^r + 1$  функцию

$$g_0 = f_0, \quad g_1 = f_0 \oplus f_1, \quad \dots, \quad g_i = f_{i-1} \oplus f_i, \quad \dots, \quad g_{2^r-1} = f_{2^r-2} \oplus f_{2^r-1}, \quad g_{2^r} = f_{2^r-1}$$

(здесь индексы  $i = \sigma$  интерпретируются как двоичные числа). Заметим, что

$$g_0 \oplus \dots \oplus g_i = f_i = g_{i+1} \oplus \dots \oplus g_{2^r}.$$

Таким образом, если  $X_1 = i \leq j = Y_1$ , то мы можем определить  $f_i(X_2)$  и  $f_j(Y_2)$ , подав на входы схем, вычисляющих  $g_0, \dots, g_i$ , переменные  $X$ , а на входы схем, вычисляющих  $g_{j+1}, \dots, g_{2^r}$ , — переменные  $Y$ . В случае  $i > j$  переменные  $X$  и  $Y$  меняются местами. В согласии с указанным правилом введем функции-индикаторы

$$\eta_i^X = (i \leq X_1 \leq Y_1) \vee (i > X_1 > Y_1), \quad \eta_i^Y = (i > Y_1 \geq X_1) \vee (i \leq Y_1 < X_1),$$

выбирающие, использовать схему  $g_i$  для вычисления  $f(X)$  или для вычисления  $f(Y)$ . Окончательно значения  $f(X)$  и  $f(Y)$  определяются по формулам

$$f(X) = \bigoplus_{i=0}^{2^r} \eta_i^X g_i(Z_i), \quad f(Y) = \bigoplus_{i=0}^{2^r} \eta_i^Y g_i(Z_i), \quad Z_i = \eta_i^X X_2 \vee \eta_i^Y Y_2.$$

Сложность каждой индикаторной функции и оператора  $Z_i$  линейна, поэтому

$$C(f(X), f(Y)) \leq \sum_{i=0}^{2^r} 2^r C(g_i) + O(2^r n) \lesssim (2^r + 1) \frac{2^{n-r}}{n - r} + O(2^r n)$$

согласно теореме 3.2. Достаточно положить  $r \sim \log n$ . ■

Мы видим, что при вычислении нескольких значений функции таблица функции (специальным образом подготовленная) фактически используется однократно, что и приводит к экономии сложности.

- Метод теоремы 10.2 оставляет огромный запас для увеличения числа значений, которые могут быть вычислены асимптотически без увеличения сложности. Улиг показал, что  $(1 + o(1))2^r$  вспомогательных функций  $g_i$  достаточно, чтобы определять значения функции  $f$  на  $s = 2^{o(r/\log r)}$  независимых наборах (этот результат можно получить, рекурсивно применяя метод теоремы 10.2, см. [323]). Как следствие [322]<sup>2)</sup> (см. также [323, 49]),

$$C_{B_2}(f(X_1), \dots, f(X_s)) \lesssim 2^n/n.$$

Задача вычисления значений функции близка к задаче реализации мультиплексорных функций с независимыми адресными переменными и общими информационными. Метод Улига автоматически влечет

$$C_{B_0}(\mu_n(X_1; Y), \dots, \mu_n(X_s; Y)) \sim 2^{n+1},$$

как и в случае  $s = 1$  (теорема 3.5). В этой версии задачи Дж. Холмгрен и Р. Ротблюм [224] повысили границу числа мультиплексоров в системе, при котором сохраняется линейная сложность  $O(2^n)$ , до  $s \asymp 2^n/n^3$ .

Дж. Гальбъяти и М. Фишер [197] показали, что аналог теоремы 10.2 в монотонном базисе не имеет места:

$$C_{B_M}(f(X), g(Y)) = C_{B_M}(f) + C_{B_M}(g), \quad (10.1)$$

если  $X$  и  $Y$  имеют не более одной общей переменной.

## Умножение матриц. Метод прямых сумм $[m][\varepsilon]$

Если в предыдущих параграфах было показано, что экономия сложности при массовом производстве возможна, то в этом речь пойдет о приложении принципа массового производства к одной из центральных задач теории вычислений — быстрым умножением матриц.

Метод прямых сумм А. Шёнхаге [303] служит нетривиальным обобщением следствия 8.2 — он позволяет эффективно превращать схемы для нескольких независимых матричных умножений в схемы умножения квадратных матриц.



Арнольд Шёнхаге  
Тюбингенский университет,  
с 1972 по 1989

Через  $T_1 \oplus T_2$  будем обозначать объединение систем билинейных форм  $T_1(X, Y)$  и  $T_2(X', Y')$ , построенных на непересекающихся множествах переменных,  $X \cap X' = Y \cap Y' = \emptyset$  (новая система имеет смысл прямой суммы тензоров). Пусть для краткости  $T^{\oplus s}$  означает прямую сумму  $s$  экземпляров системы  $T$ .

Какую пользу могут принести прямые суммы? Очевидно,  $\text{rk}(T_1 \oplus T_2) \leq \text{rk } T_1 + \text{rk } T_2$  и  $\underline{\text{rk}}(T_1 \oplus T_2) \leq \underline{\text{rk}} T_1 + \underline{\text{rk}} T_2$ . При этом, если в отношении первого

<sup>2)</sup>Цитируется по [330].

неравенства есть гипотеза (пока не опровергнутая), что на самом деле имеет место равенство, то для граничных рангов это не так. Изящный пример построил А. Шёнхаге [303].

**Теорема 10.3** ([303]). *В кольце  $R$  справедливо*

$$\underline{\text{rk}}^R(MM_{m,1,p} \oplus MM_{1,(m-1)(p-1),1}) = mp + 1.$$

► Для записи представления, доказывающего верхнюю оценку, удобно использовать трилинейное тождество (8.9). Пусть

$$\sum_{i=1}^m \sum_{j=1}^p x_i y_j z_{j,i}, \quad \sum_{i=1}^{m-1} \sum_{j=1}^{p-1} x_{i,j} y_{i,j} z$$

трилинейные формы, которые требуется выразить. Введем дополнительные обозначения

$$x_{m,j} = - \sum_{i=1}^{m-1} x_{i,j}, \quad y_{m,j} = 0, \quad x_{i,p} = 0, \quad y_{i,p} = - \sum_{j=1}^{p-1} y_{i,j}.$$

Подходящее представление имеет вид

$$\begin{aligned} u^2 \sum_{i=1}^m \sum_{j=1}^p (x_i y_j z_{j,i} + x_{i,j} y_{i,j} z) = \\ \sum_{i=1}^m \sum_{j=1}^p (x_i + ux_{i,j})(y_j + uy_{i,j})(z + u^2 z_{j,i}) - \left( \sum_{i=1}^m x_i \right) \left( \sum_{j=1}^p y_j \right) z \mod u^3. \end{aligned}$$

Точность оценки очевидна ввиду того, что система содержит  $mp + 1$  линейно независимых форм. ■

- При этом  $\underline{\text{rk}} MM_{m,1,p} = mp$  и  $\underline{\text{rk}} MM_{1,(m-1)(p-1),1} = (m-1)(p-1)$  (в силу размерности систем; во втором случае, с учетом леммы 8.5 — коммутативность для доказательства не требуется).

Следующий результат часто называют  $\tau$ -теоремой Шёнхаге.

**Теорема 10.4** ([303]). *Пусть в коммутативном кольце  $R$  выполнено  $\underline{\text{rk}}^R \bigoplus_{i=1}^k MM_{m_i, p_i, q_i} = r > 2$  и  $m_i p_i q_i \geq 2$  при некотором  $i$ , а  $w$  определяется из соотношения  $\sum_i (m_i p_i q_i)^{w/3} = r$ . Тогда*

$$C_{\mathcal{A}^R}(MM_n) \preccurlyeq n^{w+o(1)}.$$

► Предварительно заметим, что операция тензорного произведения дистрибутивна относительно прямой суммы:

$$(T_1 \oplus T_2) \otimes T = (T_1 \otimes T) \oplus (T_2 \otimes T), \quad T \otimes (T_1 \oplus T_2) = (T \otimes T_1) \oplus (T \otimes T_2).$$

В частности, лемма 5.4 применительно к произведению прямых сумм формулируется так: если система  $\bigoplus_{i=1}^I T_i$  имеет  $(d_1, r_1)$ -представление, а  $\bigoplus_{j=1}^J T'_j$  имеет  $(d_2, r_2)$ -представление над кольцом  $R$ , то система  $\bigoplus_{i=1}^I \bigoplus_{j=1}^J (T_i \otimes T'_j)$  имеет  $(d_1 + d_2, r_1 r_2)$ -представление.

Докажем еще одну лемму о граничном ранге тензорного произведения.

**Лемма 10.1** ([303]). Пусть система  $T_1$  имеет  $(d, r)$ -представление, а система  $T_2^{\oplus r}$  имеет  $(d', r')$ -представление над кольцом  $R$ . Тогда  $T_1 \otimes T_2$  имеет  $(d+d', r')$ -представление.

▷ При помощи данного условием  $(d, r)$ -представления для  $T_1$  запишем

$$u^d(T_1 \otimes T_2) = \sum_{l=1}^r C_l(u) \otimes (X_l(u)Y_l(u)) \bmod u^{d+1},$$

где  $C_l(u) \in R[u]^{\dim T_1}$ , а  $X_l(u)$  и  $Y_l(u)$  — линейные комбинации векторов переменных  $X^i$  и  $Y^j$  соответственно. Считая  $r$  внутренних произведений  $X_l Y_l$  независимыми, выразим их при помощи заданного  $(d', r')$ -представления (домножив на  $u^{d'}$ ):

$$\begin{aligned} u^{d+d'}(T_1 \otimes T_2) &= \sum_{l=1}^r C_l(u) \otimes \left( \sum_{s=1}^{r'} C'_{l,s}(u) X'_s(u) Y'_s(u) \right) \bmod u^{d+d'+1} = \\ &\quad \sum_{l=1}^r \sum_{s=1}^{r'} (C_l(u) \otimes C'_{l,s}(u)) X'_s(u) Y'_s(u) \bmod u^{d+d'+1}, \end{aligned}$$

где  $C'_{l,s}(u) \in R[u]^{\dim T_2}$ , а  $X'_s(u)$  и  $Y'_s(u)$  — линейные комбинации компонент векторов  $X_l(u)$  и  $Y_l(u)$  соответственно, т. е. в конечном счете просто линейные комбинации переменных. Требуемое представление построено. □

Перейдем непосредственно к доказательству теоремы, которое проведем только для случая  $k = 2$  (общий случай не отличается принципиально). Путем  $h$ -кратного применения леммы 5.4 из  $(d, r)$ -представления для  $MM_{m_1, p_1, q_1} \oplus MM_{m_2, p_2, q_2}$  получим  $(hd, r^h)$ -представление для системы  $\bigoplus_{s=0}^h MM_{m_1^s m_2^{h-s}, p_1^s p_2^{h-s}, q_1^s q_2^{h-s}}^{\oplus C_h^s}$  ввиду (5.13). Из

$$r^h = ((m_1 p_1 q_1)^{w/3} + (m_2 p_2 q_2)^{w/3})^h = \sum_{s=0}^h C_h^s (m_1 p_1 q_1)^{ws/3} (m_2 p_2 q_2)^{w(h-s)/3}$$

следует что при некотором  $s = s(h)$

$$C_h^s (m_1 p_1 q_1)^{ws/3} (m_2 p_2 q_2)^{w(h-s)/3} \geq \frac{r^h}{h+1}. \quad (10.2)$$

**Лемма 10.2.** Пусть  $g(h) = \lceil \gamma^{h^\alpha} r^{3h/w} \rceil$ . Тогда при подходящем выборе констант  $0 < \gamma, \alpha < 1$  для  $MM_{g(h)}$  существует  $(hb, r^{3h})$ -представление, где  $b = \lceil \frac{3 \log_2 r}{\log_2 r - 1} d \rceil$ .

▷ I. Докажем индуктивный переход, предполагая утверждение верным для всех значений параметра  $< h$ .

Пусть  $r^{3h_0} \leq C_h^{s(h)} < r^{3h_0+3}$ , т. е.  $h_0 = \lfloor (\log_r C_h^{s(h)})/3 \rfloor$ . Положим  $n = g(h_0)$  и  $s = s(h)$ . Таким образом, исходя из построенного выше  $(hd, r^h)$ -представления

для системы  $MM_{m_1^s m_2^{h-s}, p_1^s p_2^{h-s}, q_1^s q_2^{h-s}}^{\oplus r^{3h_0}}$  и предполагаемого  $(h_0 b, r^{3h_0})$  представления для  $MM_n$ , при помощи леммы 10.1 получаем  $(hd + h_0 b, r^h)$ -представление для  $MM_{nm_1^s m_2^{h-s}, np_1^s p_2^{h-s}, nq_1^s q_2^{h-s}}$ . Следовательно, согласно лемме 5.4 имеем  $(3hd + 3h_0 b, r^{3h})$ -представление для  $MM_{n^3(m_1 p_1 q_1)^s (m_2 p_2 q_2)^{h-s}}$ . В силу выбора  $s$  из (10.2) следует

$$n^3(m_1 p_1 q_1)^s (m_2 p_2 q_2)^{h-s} \geq \left( \frac{r^h n^w}{(h+1)C_h^s} \right)^{3/w}.$$

Для обоснования индуктивного перехода достаточно показать, что левая часть неравенства не меньше  $g(h)$ .

Обозначим  $f(h) = \gamma^{h^\alpha}$ . Заметим, что ввиду  $h_0 < (h/3) \log_r 2$

$$\frac{n^w}{C_h^s} > \frac{g^w(h_0)}{r^{3h_0+3}} \geq \frac{f^w(h_0)}{r^3} > \frac{f^w(\frac{h}{3 \log_2 r})}{r^3}.$$

Если обеспечить выполнение неравенства

$$\frac{f^3(\frac{h}{3 \log_2 r})}{(r^3(h+1))^{3/w}} \geq f(h), \quad (10.3)$$

то получим требуемое (с учетом округления до ближайшего целого сверху) соотношение

$$\left( \frac{r^h n^w}{(h+1)C_h^s} \right)^{3/w} > \frac{r^{3h/w} f^3(\frac{h}{3 \log_2 r})}{(r^3(h+1))^{3/w}} \geq f(h) r^{3h/w}.$$

Для обеспечения (10.3) выберем  $\alpha < 1$  произвольно из условия  $(3 \log_2 r)^\alpha > 3$ . Тогда (10.3) выполнено при любом  $\gamma \in (0, 1)$  при достаточно больших  $h$ , скажем, при  $h \geq h_1$  для любого  $\gamma \leq \gamma_1$ . Наконец, заметим, что в силу выбора  $b$  выполнено  $hb \geq 3hd + 3h_0 b$ .

*II.* Осталось указать базу индукции. Пусть при  $h \geq h_2$  выполняется  $r^3 \leq C_h^{s(h)}$ . Включим в базу индукции все  $h \leq \max\{h_1, h_2\}$  (из  $h \leq h_2$  следует  $h_0 \geq 1$  в индуктивном переходе). Окончательно выберем  $\gamma$  достаточно малым,  $\gamma \leq \gamma_1$ , чтобы искомое представление для  $MM_{g(h)}$  существовало при всех  $h \leq \max\{h_1, h_2\}$ . Например, для этого достаточно обеспечить  $g(h) \leq r^h$  при таких  $h$ .  $\square$

Пусть  $g^{k-1}(h) < n \leq g^k(h)$ . Из леммы 10.2 и леммы 5.3 следует  $\text{rk } MM_{g(h)} \leq ch^2 r^{3h}$ , где  $c = O(1)$ . Тогда по теореме 5.3,

$$\begin{aligned} C_{\mathcal{A}^R}(MM_n) &\leq C_{\mathcal{A}^R}(MM_{g^k(h)}) \leq g^2(h)(ch^2 r^{3h})^{k+2} \leq \\ &\leq g^2(h)r^{9h}(ch^2)^{k+2} (\gamma^{-h^\alpha} g(h))^{(k-1)w} \leq r^{c_1 h} c_2^{kh^\alpha} \cdot n^w \end{aligned}$$

при подходящих константах  $c_1, c_2$ . Утверждение теоремы следует, например, при выборе  $h \asymp \log^{1/2} n$  (при этом  $k \asymp h$ ).  $\blacksquare$

Применяя теорему 10.4 к примеру из теоремы 10.3 с выбором параметров  $m = p = 4$ , получаем

**Следствие 10.2** ([303]). *Если  $R$  — коммутативное кольцо, то*

$$\mathsf{C}_{\mathcal{A}^R}(MM_n) \prec n^{2.548}.$$

- Вероятно, наиболее сильную оценку непосредственно при помощи метода теоремы 10.4 получили Д. Копперсмит и Ш. Виноград в [177],  $\mathsf{C}_{\mathcal{A}^R}(MM_n) \prec n^{2.496}$ . Более современные теоретически быстрые методы умножения матриц используют метод прямых сумм в качестве рабочего инструмента.

### Умножение матриц. Лазерный метод $\boxed{m} \boxed{\varepsilon}$

Идея массового производства оказалась чрезвычайно востребованной в задаче быстрого умножения матриц. В «лазерном» методе Ф. Штассена [318] комбинируются сразу несколько вариантов этой идеи.

Дальнейшее изложение следует близко к работе Д. Копперсмита и Ш. Винограда [178]. Метод позволяет строить алгоритмы умножения матриц из тождеств для систем билинейных форм, которые не являются произведениями матриц. Например, система  $T_q = \{x_0y_i + x_iy_0 \mid i = 1, \dots, q\}$  может быть (приближенно) вычислена как

$$u(x_0y_i + x_iy_0) = (x_0 + ux_i)(y_0 + uy_i) - x_0y_0 \mod u^2 \quad (10.4)$$

(пример из работы [318]). Поэтому  $\underline{\text{rk}} T_q \leq q + 1$ . Система  $T_q$  получается суммированием компонент матричных умножений  $MM_{1,1,q}$  и  $MM_{q,1,1}$ , но эта сумма не является прямой, поэтому теорему 10.4 нельзя применить непосредственно. Метод Штассена объясняет, как перейти от (10.4) к тождеству с прямой суммой в левой части.

**Теорема 10.5** ([318]). *Если  $R$  — коммутативное кольцо, то*

$$\mathsf{C}_{\mathcal{A}^R}(MM_n) \prec n^{2.48}.$$

- Первым шагом тождество (10.4) приводится к более симметричному виду. Построим тензорное произведение  $T_q \otimes T'_q \otimes T''_q$ , где системы  $T'_q, T''_q$  получаются из  $T_q$  циклическим сдвигом групп переменных  $(x, y, z)$  в соответствующей  $T_q$  трилинейной форме  $\sum_{i=1}^q (x_0y_i + x_iy_0)z_i$ , см. (8.9). Таким образом,

$$T'_q = \left\{ x_0y_1, \dots, x_0y_q, \sum_{j=1}^q x_jy_j \right\}, \quad T''_q = \left\{ x_1y_0, \dots, x_qy_0, \sum_{k=1}^q x_ky_k \right\}.$$

По построению,  $\underline{\text{rk}} T_q = \underline{\text{rk}} T'_q = \underline{\text{rk}} T''_q$  (фактически доказано в лемме 8.5 для частного случая умножения матриц). В частности, система  $T'_q$  имеет  $(1, q+1)$ -представление вида

$$ux_0y_j = u(x_0 + ux_j)y_j \mod u^2, \quad u \sum_{j=1}^q x_jy_j = \sum_{j=1}^q (x_0 + ux_j)y_j - x_0 \sum_{j=1}^q y_j.$$

Система  $T_q \otimes T'_q \otimes T''_q$  состоит из билинейных форм  $x_{00k}y_{ij0} + x_{i0k}y_{0j0}$  ( $q^3$  штук),  $\sum_{k=1}^q x_{00k}y_{ijk} + x_{i0k}y_{0jk}$  ( $q^2$  штук),  $\sum_{j=1}^q x_{0jk}y_{ijo} + x_{ijk}y_{0jo}$  ( $q^2$  штук) и  $\sum_{j,k=1}^q x_{0jk}y_{ijk} + x_{ijk}y_{0jk}$  ( $q$  штук). В ней можно увидеть структуру блочного произведения матриц размера  $2 \times 2$ :

$$\begin{bmatrix} x_{00k}y_{ij0} + x_{i0k}y_{0j0} & x_{00k}y_{ijk} + x_{i0k}y_{0jk} \\ x_{0jk}y_{ijo} + x_{ijk}y_{0jo} & x_{0jk}y_{ijk} + x_{ijk}y_{0jk} \end{bmatrix} = \begin{bmatrix} x_{00k} & x_{i0k} \\ x_{0jk} & x_{ijk} \end{bmatrix} \cdot \begin{bmatrix} y_{ij0} & y_{ijk} \\ y_{0j0} & y_{0jk} \end{bmatrix} \quad (10.5)$$

(суммирование выполняется по повторяющимся индексам; знаки суммирования для краткости опущены). Осложнение здесь в том, что блоками матриц являются по сути трехмерные тензоры, которые нельзя согласованно изобразить обычными (двумерными) матрицами. Тем не менее, индивидуальные произведения блоков являются произведениями матриц: например,  $x_{00k}y_{ij0}$  изображает оператор  $MM_{q,1,q^2}$ , а  $x_{ijk}y_{0jk}$  — оператор  $MM_{q,q^2,1}$ .

Ключевое наблюдение Штассена состоит в том, что из обычного произведения (достаточно больших) матриц можно извлечь множество индивидуальных произведений компонент (или блоков).

**Лемма 10.3.** *Пусть система  $MM_n$  имеет  $(d, r)$ -представление. Тогда система*

$$B_{n,s} = \{x_{ij}y_{jk} \mid 1 \leq i, j, k \leq n, i + j + k = s\}$$

*имеет  $(d + h, r)$ -представление,  $h \preccurlyeq n^2$ .*

▷ Имея в виду (8.9), можно считать, что нам дано  $(d, r)$ -представление для трилинейной формы  $\sum_{1 \leq i, j, k \leq n} x_{ij}y_{jk}z_{ki}$ . После замены переменных

$$x_{ij} := u^{i^2+2ij} \cdot x_{ij}, \quad y_{jk} := u^{j^2+2j(k-s)} \cdot y_{jk}, \quad z_{ki} := u^{(k-s)^2+2(k-s)i} \cdot z_{ki}$$

в силу  $i^2 + 2ij + j^2 + 2j(k-s) + (k-s)^2 + 2(k-s)i = (i+j+k-s)^2$  получим почти  $(d, r)$ -представление для  $\sum_{i+j+k=s} x_{ij}y_{jk}z_{ki}$ , которое может отличаться от правильного представления вхождениями отрицательных степеней  $u$ . От отрицательных степеней можно избавиться домножением на  $u^h$  с показателем  $h \preccurlyeq n^2$ .  $\square$

Важно, что любая переменная  $x_{ij}$  или  $y_{jk}$  имеет не более одного вхождения в  $B_{n,s}$  (значения пары индексов из  $i, j, k$  однозначно определяют третий), поэтому  $B_{n,s}$  является прямой суммой отдельных форм. При выборе  $s \sim 3n/2$  система  $B_{n,s}$  содержит  $(3/4 - o(1))n^2$  форм.

Согласно лемме 5.4 система  $T_q \otimes T'_q \otimes T''_q$  имеет  $(3, (q+1)^3)$ -представление. Тогда тензорная степень  $(T_q \otimes T'_q \otimes T''_q)^{\otimes m}$  имеет  $(3m, (q+1)^{3m})$ -представление, также по лемме 5.4. Как следствие из (10.5), система  $(T_q \otimes T'_q \otimes T''_q)^{\otimes m}$  имеет структуру произведения матриц размера  $2^m \times 2^m$  из блоков  $X_{ij}$  и  $Y_{jk}$ , где произведение двух блоков  $X_{ij}Y_{jk}$  изображает умножение матриц  $MM_{q^a, q^b, q^c}$ ,  $a + b + c = 3m$ . Следовательно, согласно лемме 10.3, граничный ранг системы  $B_{2^m, s} \otimes \{X_{ij}Y_{jk} \mid 1 \leq i, j, k \leq 2^m\}$  не превосходит  $(q+1)^{3m}$ . Но эта система является прямой суммой произведений матриц  $MM_{q^a, q^b, q^c}$ . Поэтому, применяя теорему 10.4, при

выборе  $s \sim (3/2)2^m$  получаем  $C_{\mathcal{A}^R}(MM_n) \preccurlyeq n^{w+o(1)}$ , где  $w$  определяется из условия  $(3/4)2^{2m}q^{wm} = (q+1)^{3m}$ , или, после извлечения корня, из  $4q^w = (q+1)^3$ . Утверждение теоремы получается при  $q = 5$ .  $\blacksquare$

- Дальнейшее развитие метода предприняли Д. Копперсмит и Ш. Виноград в [178], где с использованием более сложных, нежели (10.4), базовых представлений получена оценка  $C_{\mathcal{A}^R}(MM_n) \prec n^{2.376}$ , которая оставалась рекордной на протяжении 20 лет. Наилучшая на данный момент<sup>3)</sup> оценка имеет вид  $C_{\mathcal{A}^R}(MM_n) \prec n^{2.372}$  [121]. Систематическое изложение теории быстрых методов умножения матриц см. также в [1, 17].

## Другие приложения

**Монотонные схемы для слой-функций.** Любопытное приложение метода массового производства нашли А. Хилтген и М. Патерсон [220]. Оно относится к совместному вычислению подобных слой-функций.  $k$ -слой-функцией называется монотонная функция вида  $T_{|X|}^k(X)f(X) \vee T_{|X|}^{k+1}(X)$  — она равна 0 на всех наборах веса  $< k$  и равна 1 на всех наборах веса  $> k$  (здесь  $f$  — произвольная булева функция).

Пусть  $X = (X_1, \dots, X_r)$ ,  $|X_1| = \dots = |X_r| = m$ ,  $n = rm$ . Рассмотрим задачу вычисления семейства  $k$ -слой-функций  $f_i(X) = X_i^{\alpha_1} \vee \dots \vee X_i^{\alpha_s} \vee T_n^{k+1}(X)$ , где  $\alpha_i$  — различные векторы веса  $k \leq m$ , а  $X_i^\sigma = \prod_{\sigma_j=1} x_{i,j}$  — мономы переменных  $X_i$ . Наличие общей части  $T_n^{k+1}(X)$  выводит эту задачу из зоны действия правила (10.1).

В обозначениях  $\dot{x}_{i,j} = x_{i,j} \cdot T_m^k(X_i)$ ,  $\dot{x}_j = \dot{x}_{1,j} \vee \dots \vee \dot{x}_{r,j}$ ,  $\dot{X} = (\dot{x}_1, \dots, \dot{x}_m)$  каждая из функций  $f_i$  может быть вычислена как

$$f_i(X) = T_m^k(X_i)(\dot{X}^{\alpha_1} \vee \dots \vee \dot{X}^{\alpha_s}) \vee T_n^{k+1}(X),$$

следовательно,  $C_{\mathcal{B}_M}(f_1, \dots, f_r) < n + ks + rC_{\mathcal{B}_M}(T_m^k) + C_{\mathcal{B}_M}(T_n^{k+1}) = ks + O(n \log n)$ . При достаточно больших  $r$  этот способ вычисления экономнее независимой реализации функций  $f_i$  или сумм  $X_i^{\alpha_1} \vee \dots \vee X_i^{\alpha_s}$ .

Более тонко применяя указанный прием, авторы [220] получили асимптотически точную оценку  $C_{\mathcal{B}_M}(\{f_{a,b} \mid 0 \leq a, b < p\}) \sim 3n$  сложности семейства 1-слой-функций Нечипорука,  $n = p^2$  и  $p \in \mathbb{P}$ . Здесь  $X = (x_{i,j})$  и  $f_{a,b}(X) = \bigvee_{i=0}^{p-1} x_{i,ai+b} \vee T_n^2(X)$  (операции с индексами выполняются по модулю  $p$ ). Конструкция функций следует матрице инциденций точек и прямых конечной аффинной плоскости над  $\mathbb{F}_p$ . Как установил Э. И. Нечипорук в [58], линейный над  $(\mathbb{B}, \vee)$  оператор с такой матрицей<sup>4)</sup> имеет монотонную сложность  $(p-1)n^2 \sim n^{3/2}$ . Гипотеза [329] о том, что и слой-версия оператора должна иметь сложность порядка  $n^{3/2}$ , как раз и была опровергнута в [220].

---

<sup>3</sup>Апрель 2024 г.

<sup>4</sup>Компонентами оператора являются булевые суммы  $\bigvee_{i=0}^{p-1} x_{i,ai+b}$  — любые две таких суммы имеют не более одной общей переменной.

# Глава 11

## Комбинаторные методы

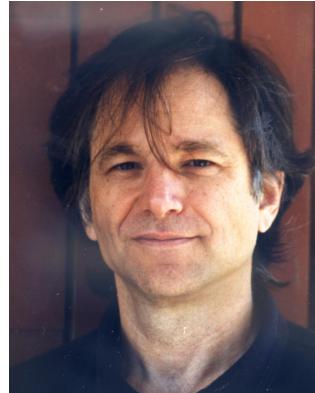


В ряде задач теории быстрых вычислений оказывается удобным организовать переменные или промежуточные данные в виде структуры с определенными комбинаторными свойствами.

### Монотонные схемы для симметрической функции с порогом 2



Для функции  $T_n^2$  нижняя оценка сложности  $C_{B_2}(T_n^2) \geq 2n - O(1)$  была получена одной из первых (Б. М. Клоссом в [23]), причем сразу в полном базисе. Вопрос о возможности вычисления функции с такой сложностью оставался открытым, пока Л. Адлеман в 1970-х гг. не нашел изящный способ<sup>1)</sup>, причем сразу для монотонного базиса. В методе Адлемана переменные соответствуют узлам двумерной решетки. Существенно то, что любые два узла находятся либо в разных строках, либо в разных столбцах решетки.



Леонард Макс  
Адлеман

Университет  
Южной Калифорнии, с 1980

**Теорема 11.1.**  $C_{B_M}(T_n^2) \leq 2n + O(\sqrt{n})$ .

► Пусть множество из  $n$  переменных занумеровано двумя индексами  $x_{i,j}$ ,  $i = 1, \dots, p$ ,  $j = 1, \dots, q$ , где  $pq \geq n$ . Обозначим  $u_i = \bigvee_j x_{i,j}$  и  $v_j = \bigvee_i x_{i,j}$ . Тогда

$$T_n^2(X) = T_p^2(u_1, \dots, u_p) \vee T_q^2(v_1, \dots, v_q),$$

поэтому

$$C(T_n^2) \leq 2n - p - q + 1 + C(T_p^2) + C(T_q^2),$$

поскольку все суммы  $u_i$  и  $v_j$  вычисляются со сложностью  $2n - p - q$ . Теперь достаточно выбрать  $p = q \approx \sqrt{n}$ . ■

<sup>1)</sup>Не опубликован автором, цитируется по [147, 330].

- Более точно, оценка, извлекаемая из теоремы 11.1, имеет вид  $C_{\mathcal{B}_M}(T_n^2) \leq 2n + (2 + o(1))\sqrt{n}$ . Доказанная автором в [93] нижняя оценка имеет вид  $C_{\mathcal{B}_M}(T_n^2) \geq 2n + \sqrt{2n/3} - O(1)$ .

Метод Адлемана допускает обобщение и позволяет получить оценку  $C_{\mathcal{B}_M}(T_n^k) \leq kn + o(n)$  при любом постоянном  $k$ , см. [330], однако в этом случае приоритет имеет адаптированный метод Яо, см. [93].

### Асимптотическая сложность формул $\vdash c$

В теории асимптотически оптимального синтеза применяются разнообразные покрытия булева куба, обладающие полезными в рамках заданной модели вычислений свойствами. Метод покрытий куба сферами был предложен О. Б. Лупановым [43] для построения экономных контактных схем и впоследствии нашел применение в ряде других задач. Мы рассмотрим приложение метода к синтезу асимптотически минимальных формул в базисе  $\mathcal{B}_0$  — результат Лупанова [44].

*Сфера* (радиуса 1) в пространстве  $\mathbb{B}^r$  определяется как множество булевых векторов длины  $r$ , отличающихся от заданного (центра сферы) ровно в одной координате. Важнейшим свойством сферы является возможность выделения любой точки сферы при помощи одной переменной в следующем смысле. Пусть  $\varphi_\alpha(X)$  — характеристическая функция<sup>2)</sup> сферы  $S$  с центром  $\alpha$ . Тогда для произвольного набора  $\sigma \in S$  выполнено  $X^\sigma = \varphi_\alpha(X)x_i^{\sigma_i}$ , где  $i$  — номер позиции, отличающей  $\sigma$  от  $\alpha$ .

Отталкиваясь от совершенного кода Хэмминга с расстоянием 3, легко проверить, что при  $r = 2^k$  булев куб  $\mathbb{B}^r$  распадается на  $2^r/r$  сфер, см., например, [48].

- При  $r = 2^k$  код Хэмминга порождает разбиение куба  $\mathbb{B}^{r-1}$  на  $s = 2^{r-1}/r$  непересекающихся шаров радиуса 1 с центрами  $\alpha_1, \dots, \alpha_s$ . Как следствие, множество сфер с центрами  $\{(\alpha_i, \beta) \mid 1 \leq i \leq s, \beta \in \mathbb{B}\}$  образует разбиение куба  $\mathbb{B}^r$ . В общем случае  $r \neq 2^k$  куб  $\mathbb{B}^r$  можно покрыть  $(1 + o(1))2^r/r$  сферами [18].

Прежде чем переходить к доказательству основного результата, получим вспомогательную оценку.

**Лемма 11.1** ([44]). *Пусть  $|X| = p$ ,  $|Y| = r$  и*

$$f(X, Y) = y_1 f_1(X) \vee y_2 f_2(X) \vee \dots \vee y_r f_r(X). \quad (11.1)$$

*Тогда  $\Phi_{\mathcal{B}_0}(f) \leq 2^p \left(\frac{r}{s} + p2^s\right)$  при любом  $s \in \mathbb{N}$ .*

▷ Разобьем булевые векторы длины  $r$  на группы  $I_1, \dots, I_q$  по  $s$  штук в каждой (для простоты полагаем  $2^p = qs$ ). Пусть  $\chi_{j,\tau}(X)$  — характеристическая функция множества векторов группы  $I_j$ , определяемого вектором<sup>3)</sup>  $\tau \in \mathbb{B}^s$ . Тогда

$$f(X, Y) = \bigvee_{k=1}^q \bigvee_{\tau \in \mathbb{B}^s} \chi_{k,\tau}(X) D_{k,\tau}(Y), \quad D_{k,\tau}(Y) = \bigvee_{f_i(I_k) = \chi_{k,\tau}(I_k)} y_i. \quad (11.2)$$

<sup>2</sup>Характеристическая функция множества  $S$  принимает значение 1 в точках  $S$ , и значение 0 — за пределами множества.

<sup>3</sup> $i$ -й вектор принадлежит множеству, если  $\tau_i = 1$ .

Поскольку  $\sum_{\tau} \Phi(D_{k,\tau}) = r$  при любом  $k$ , и  $\Phi(\chi_{j,\tau}) \leq ps$  (используем совершенную ДНФ), формула (11.2) имеет сложность

$$\Phi(f) \leq qr + q2^sps = 2^p \left( \frac{r}{s} + p2^s \right).$$

□

**Теорема 11.2** ([44]). Для произвольной булевой функции  $f$  от  $n$  переменных

$$\Phi_{B_0}(f) \leq \left( 1 + O \left( \frac{\log \log n}{\log n} \right) \right) \frac{2^n}{\log_2 n}.$$

► Пусть  $X = (X_1, X_2, X_3)$ ,  $|X_1| = p$ ,  $|X_2| = r = 2^k$ ,  $|X_3| = n - p - r$ . Используя разбиение куба  $\mathbb{B}^r$  на сферы с характеристическими функциями  $\varphi_i(X_2)$ , запишем

$$f(X) = \bigvee_{\sigma \in \mathbb{B}^{n-p-r}} f_{\sigma}(X_1, X_2) \cdot X_3^{\sigma} = \bigvee_{i=1}^{2^r/r} \varphi_i(X_2) \bigvee_{\sigma \in \mathbb{B}^{n-p-r}} f_{i,\sigma}(X_1, X_2) \cdot X_3^{\sigma}, \quad (11.3)$$

где каждая из функций  $f_{i,\sigma}(X_1, X_2)$  имеет вид (11.1) (в роли переменных  $Y$  выступают переменные  $X_2$  или их отрицания).

Оценивая грубо  $\Phi(\varphi_i) \leq r^2$  и применяя лемму 11.1, для сложности вычисления функции по формуле (11.3) получаем

$$\Phi(f) \leq \frac{2^r}{r} \left( r^2 + 2^{n-p-r} \left( n - p - r + 2^p \left( \frac{r}{s} + p2^s \right) \right) \right) < 2^r r + 2^{n-p} \cdot \frac{n}{r} + \frac{2^n}{s} + 2^{n+s} \cdot \frac{p}{r}.$$

Требуемая оценка получается при выборе параметров  $n/4 \leq r \leq n/2$ ,  $p \approx 2 \log_2 \log n$ ,  $s \approx \log_2 n - 2 \log_2 \log n$ . Отметим, что основная сложность приходится на переменные, выделяющие точки сфер. ■

То, что результат теоремы асимптотически точен, показано еще Дж. Риорданом и К. Шенноном в [293] (это первое приложение мощностного метода к получению нижних оценок сложности в теории синтеза).

- В работе [35] С. А. Ложкин повысил точность оценки формульной сложности до

$$\Phi_{B_0}(\mathcal{P}^n) = \left( 1 \pm O \left( \frac{1}{\log n} \right) \right) \frac{2^n}{\log_2 n}. \quad (11.4)$$

Из асимптотической оценки сложности формул автоматически вытекает оценка глубины  $D_{B_0}(\mathcal{P}^n) \geq n - \log_2 \log_2 n - o(1)$ . Используя модификацию представления (11.3), также основанную на сферическом разбиении булева куба, С. Б. Гашков [7] получил верхнюю оценку в виде  $D_{B_0}(\mathcal{P}^n) \leq \lceil n - \log_2 \log_2 n + o(1) \rceil + 2$ . Практически окончательный результат

$$D_{B_0}(\mathcal{P}^n) \leq \lceil n - \log_2 \log_2 n + o(1) \rceil \quad (11.5)$$

установил Ложкин [34], используя разбиение куба на производные от сфер специальные множества. Простой способ вывода чуть более слабой оценки изложен ниже, см. следствие 12.1. Ложкин также показал [37], что оценка сложности вида (11.4) и оценка глубины (11.5) с точностью до аддитивного слагаемого  $O(1)$  достигаются на одной формуле.

Гашков [8] адаптировал метод Лупанова к вычислению многочленов с коэффициентами 0 и 1 формулами в арифметическом базисе  $\{*, \pm, 1\}$ . Сложность класса многочленов с индивидуальными ограничением  $d_i - 1$  на степени каждой переменной  $x_i$  асимптотически равна  $d_1 \cdot \dots \cdot d_n / \log_2 n$ . Доказательство вместо покрытий сферами использует покрытия параллелепипедов в  $\mathbb{N}^n$  полусферами.

## Мультиликативная сложность многочленов $\square \vdots$

Напомним, что многочлен одной переменной степени  $d$  можно вычислить арифметической схемой, использующей  $\asymp \sqrt{d}$  нескалярных умножений, и эта оценка неулучшаема [280] (см. стр. 33). Рассмотрим более общую задачу реализации многочленов  $n$  переменных. Известно, что при  $d \geq 2$  порядка  $\sqrt{C_{n+d}^d}$  умножений необходимо в этом случае [164]<sup>4)</sup>. Ш. Ловетт [259] показал, что указанная оценка почти достижима. В основе метода лежит идея накрытия множества векторов степеней мономов суммой двух множеств примерно вдвое меньшей размерности.

**Теорема 11.3** ([259]). *Любой многочлен  $f \in R[x_1, \dots, x_n]$  степени  $d$  можно вычислить схемой над  $\mathcal{A}^R$  с нескалярной мультиликативной сложностью  $(dn)^{O(1)} \sqrt{C_{n+d}^d}$ .*

► В центре доказательства лежит наблюдение о том, что любой моном  $X^e$  степени  $\leq d$  можно представить в виде  $X^a X^b$ , где  $a \in A$ ,  $b \in B$  при подходящем выборе множеств  $A$  и  $B$ , мощность которых близка к  $\sqrt{C_{n+d}^d}$ .

Достаточно рассмотреть случай, когда  $n$  нечетно, а  $d$  четно. Выберем в качестве  $A = B$  множество всех векторов веса<sup>5)</sup>  $\leq d/2$  с нулевыми компонентами в каких-то  $m = (n-1)/2$  циклически смежных<sup>6)</sup> позициях  $i+1, i+2, \dots, i+m \subset \mathbb{Z}_n$ .

**Лемма 11.2** ([259]). *Любой вектор  $e = (e_1, \dots, e_n) \in \mathbb{N}_0^n$  веса  $d$  принадлежит сумме Минковского<sup>7)</sup>  $A + A$ .*

▷ Обозначим через  $w_i^-$  и  $w_i^+$  суммы компонент вектора  $e$  в циклически смежных позициях  $i-1, i-2, \dots, i-m$  и соответственно  $i+1, i+2, \dots, i+m$ . Заметим, что  $w_i^- + w_i^+ + e_i = d$ .

Сначала покажем, что при некотором  $i$  выполнено  $w_i^-, w_i^+ \leq d/2$ . Предположим, что  $w_j^+ > d/2$  при некотором  $j$ . Значит,  $w_j^- = w_{j+m}^+ \leq d/2$ . Как следствие, при некотором  $i$  выполнено  $w_{i-1}^+ > d/2$  и  $w_i^+ \leq d/2$ . Но тогда  $w_i^- \leq d - w_{i-1}^+ \leq d/2$ .

Теперь  $e = a + b$ , где

$$\begin{aligned} a_{i+1} &= e_{i+1}, \dots, a_{i+m} = e_{i+m}, & a_{i-1} = \dots = a_{i-m} = 0, & a_i = d/2 - w_i^+, \\ b_{i-1} &= e_{i-1}, \dots, b_{i-m} = e_{i-m}, & b_{i+1} = \dots = b_{i+m} = 0, & b_i = d/2 - w_i^-. \end{aligned}$$

Таким образом,  $a, b \in A$ .  $\square$

Как следует из леммы, многочлен  $f = \sum f_e X^e$  можно представить в виде

$$f(X) = \sum_{a \in A} X^a \cdot \sum_{b \in A} c_{a,b} X^b, \quad c_{a,b} \in \{0, f_{a+b}\}. \quad (11.6)$$

<sup>4)</sup>По крайней мере, если рассматривать многочлены над полем.

<sup>5)</sup>Здесь вес вектора определяется как сумма его компонент.

<sup>6)</sup>Т.е. идущих подряд в нумерации  $\dots, n, 1, 2, \dots, n, 1, 2, \dots$

<sup>7)</sup>Сумма Минковского  $A + B$  определяется как  $\{a + b \mid a \in A, b \in B\}$ .

Все мономы  $X^a$  вычисляются последовательно со сложностью  $|A|$  умножений, еще  $|A|$  умножений выполняются на внутренние суммы в формуле (11.6). Осталось заметить<sup>8)</sup>, что  $|A| \leq nC_{(n+d-1)/2}^{d/2-1} \leq n\sqrt{C_{n+d-1}^{d-2}} \leq d\sqrt{C_{n+d}^d}$  в силу простого неравенства  $C_{2n}^{2k} \geq (C_n^k)^2$ .

Случай четного  $n$  и нечетного  $d$  сводятся к рассмотренному (ценой дополнительного множителя в оценке сложности). ■

- Метод теоремы 11.3 очевидно позволяет вычислять и монотонные многочлены в монотонном базисе  $\mathcal{A}_+$ . Вопрос о том, можно ли подойти ближе к нижней оценке  $\sqrt{C_{n+d}^d}$ , пока открыт. Как видно из доказательства теоремы, эта оценка достигается по порядку в некоторых частных случаях, например, при четных постоянных  $d$  (что также легко проверяется непосредственно). Автор [95] показал, что при постоянных нечетных  $d$  порядок сложности равен  $n^{\lceil d/2 \rceil} \asymp \sqrt{nC_{n+d}^d}$  для полей характеристики 0. Открыт вопрос о сложности рассматриваемых классов многочленов над конечными полями.

### Сложность схем для логического перманента $\vdots \vdots s$

Выше была предъявлена вероятностная конструкция параллельных схем для функции логического перманента  $PERM_n$  полиномиальной сложности (теорема 9.5). Она была основана на алгебраических свойствах перманента. Далее мы увидим, что алгоритмы поиска максимального паросочетания в графе помогают конструктивно строить более простые, хотя уже не параллельные, схемы.

Пусть  $G = (U, V, E)$  — двудольный неориентированный граф с долями  $U = \{u_1, \dots, u_n\}$ ,  $V = \{v_1, \dots, v_n\}$  и множеством ребер  $E$ . Как обычно, вводятся булевые переменные  $x_{i,j} = [(u_i, v_j) \in E]$ . Напомним, что  $PERM_n(X) = 1$  ровно в том случае, когда граф  $G$  имеет совершенное паросочетание<sup>9)</sup>.

Пусть выбрано некоторое паросочетание  $M \subset E$ . Путь в графе  $G$  называется *M-чредующимся*, если в нем чередуются ребра из  $M$  и из  $E \setminus M$ . Вершина графа  $G$  называется *M-свободной*, если она не смежна ребрам из  $M$ . Чредующийся путь называется *M-увеличивающим*, если его крайние вершины свободны.

Легко проверить, что паросочетание  $M$  максимально в графе  $G$ , если в  $G$  нет *M-увеличивающих* путей. Более того, справедливо

**Утверждение 11.1** ([226]). *Если двудольный граф  $G$  имеет паросочетание  $N$  из  $n$  ребер, то для произвольного паросочетания  $M \subset G$  из  $m < n$  ребер найдется *M-увеличивающий* путь длины  $\leq 2m/(n-m) + 1$ .*

▷ Рассмотрим объединение паросочетаний  $M$  и  $N$ . В каждой компоненте связности графа  $M \cup N$  любая вершина смежна максимум одному ребру из  $M$  и одному ребру из  $N$ . Поэтому: (а) разность между числом ребер из  $M$  и из  $N$  в

<sup>8)</sup>Фиксируя крайнюю ненулевую позицию вектора  $a \in A$  и оценивая число способов разбить число  $d/2 - 1$  на  $m + 2$  слагаемых.

<sup>9)</sup>Паросочетание — множество несмежных по вершинам ребер. Совершенное паросочетание накрывает все вершины графа.

компоненте связности не превосходит 1 по абсолютной величине; (б) все ребра компоненты связности образуют  $M$ -чередующийся путь.

Как следствие, имеется  $\geq n - m$  компонент связности с преобладанием ребер из  $N$ , а значит столько же несмежных  $M$ -увеличивающих путей. Значит, найдется путь, имеющий не более  $m/(n - m)$  ребер из  $M$ .  $\square$

Остается заметить, что симметрическая разность<sup>10)</sup> паросочетания  $M$  и  $M$ -увеличивающего пути является паросочетанием мощности  $|M| + 1$ .

Таким образом, задачу поиска максимального паросочетания можно решить за  $n$  последовательных шагов поиска увеличивающих путей. На каждом шаге строится разбиение вершин графа на слои: на нулевом слое — свободные вершины  $U_0$  доли  $U$ , и далее на слое с номером  $d$  — вершины на расстоянии  $d$  от  $U_0$  по чередующимся путям. И так до тех пор, пока не будет охвачена свободная вершина из  $V$ . Затем восстанавливается чередующийся путь между этой вершиной и вершиной из  $U_0$ .

Таков простой алгоритм с временем работы<sup>11)</sup> порядка  $n^3$ , который изложен, например, в [4]. Представление алгоритма в виде схемы требует дополнительной проработки (примерно как в случае алгоритма для НОД, см. теорему 1.4).

**Теорема 11.4.**  $C(PERM_n) \leq n^4 \log^3 n$ .

► В схеме вычисляется последовательность паросочетаний  $M_1, M_2, \dots, M_{n+1}$ . Преобразования  $M_k \rightarrow M_{k+1}$  выполняются однотипными подсхемами  $S_k$ :

$$\mathcal{O} = M_1 \xrightarrow{S_1} M_2 \xrightarrow{S_2} \dots \xrightarrow{S_n} M_{n+1}.$$

Схема  $S_k$  ищет  $M_k$ -увеличивающий путь и в случае успеха увеличивает с его помощью мощность паросочетания. По построению,  $|M_k| \leq k - 1$ .

Паросочетание  $M$  характеризуется набором признаков  $p_{i,j} = [(u_i, v_j) \in M]$ , которые пересчитываются при изменении  $M$  в процессе вычислений. Попутно обновляются вспомогательные признаки  $p_i^u = \bigvee_j p_{i,j}$  и  $p_j^v = \bigvee_i p_{i,j}$  несвободы вершин  $u_i$  и  $v_j$ , а также признаки  $t[u, v]$  легальных<sup>12)</sup> ребер в  $M$ -чередующихся путях:

$$t[u_i, v_j] = x_{i,j} \cdot \overline{p_{i,j}}, \quad t[v_j, u_i] = p_{i,j}.$$

Это значит, что из доли  $U$  в долю  $V$  пути должны проходить по ребрам из  $E \setminus M$ , а в обратную сторону — по ребрам из  $M$ .

1) Опишем схему  $S_k$ . Согласно утверждению 11.1, если граф  $G$  обладает совершенным паросочетанием, то в нем найдется  $M_k$ -увеличивающий путь длины не более  $l_k = 2\lfloor k/(n - k) \rfloor + 1$ .

---

<sup>10)</sup> Симметрическая разность множеств  $A$  и  $B$  определяется как  $(A \setminus B) \cup (B \setminus A)$ .

<sup>11)</sup> Под временем работы алгоритма обычно понимается число инструкций, выполняемых реализацией его программой на универсальном компьютере.

<sup>12)</sup> Понятие легальности вводит ориентацию ребер в графе так, чтобы все ориентированные пути были  $M$ -чередующими. Множество чередующихся путей при этом не изменяется.

Методом динамического программирования построим кратчайшие  $M_k$ -чередующиеся пути длины  $\leq l_k$  между каждой парой вершин из  $G$ . Соответствующая схема состоит из  $s = \lceil \log_2 l_k \rceil$  слоев-подсхем  $L_1, \dots, L_s$ . Очередной  $i$ -й слой добавляет информацию о чередующихся путях длины  $\leq 2^i$ .

Далее  $\text{code}(\pi)$  обозначает код (т.е. булев вектор) чередующегося пути  $\pi$ . Мы выбираем естественный способ кодирования пути последовательностью номеров его вершин. Скажем, путь

$$u_{i_0} \rightarrow v_{i_1} \rightarrow u_{i_2} \rightarrow \dots \rightarrow v_{i_r}$$

кодируется строкой  $(i_0, i_1, i_2, \dots, i_r, 0, \dots, 0)$ . Схемы  $L_i$  оперируют путями, кодируемыми строкой из  $2^{i-1} + 1$  чисел, а коды вычисляемых ими путей содержат  $2^i + 1$  чисел. Числа записываются  $b = \lceil \log_2(n+1) \rceil$  разрядами.

Пусть  $\pi_{u,v}$  обозначает найденный чередующийся путь между  $u$  и  $v$ . Через  $\delta_{u,v}$  обозначим признак того, что путь  $\pi_{u,v}$  найден, а через  $\lambda_{u,v}$  — его длину.

1.0) Указанные величины инициализируются как  $\lambda_{u,v} = \delta_{u,v} = t[u, v]$ . Инициализация путей длины 1 выполняется со сложностью  $O(n^2 \log n)$ :

$$\text{code}(\pi_{u_i, v_j}) = t[u_i, v_j] \cdot (i, j), \quad \text{code}(\pi_{v_j, u_i}) = t[v_j, u_i] \cdot (j, i).$$

1.1) На слое  $L_i$  для каждой упорядоченной пары вершин  $(u, v)$  выполняются следующие вычисления.

Если  $\delta_{u,v} = 0$ , то для всех прочих вершин  $w \in G$  вычисляем признаки наличия и длину пути из  $u$  в  $v$  через  $w$ :

$$\delta_{u,w,v} = \delta_{u,w} \delta_{w,v}, \quad \lambda_{u,w,v} = \lambda_{u,w} + \lambda_{w,v}.$$

Некоторые пути  $\pi_{u,w} \cup \pi_{w,v}$  среди рассматриваемых могут содержать циклы и, таким образом, не быть кратчайшими. Поэтому среди всех найденных путей, т.е. при  $\delta_{u,w,v} = 1$ , следует определить путь минимальной длины  $\lambda_{u,w,v}$  — он циклов не содержит.

Схема сначала вычисляет  $\lambda_{u,v} = \min_{\delta_{u,w,v}=1} \lambda_{u,w,v}$ , полагая  $\lambda_{u,v} = 0$ , если путей не найдено. Далее вычисляются индикаторы  $I_w$  первой позиции  $w$ , для которой выполняется  $\lambda_{u,v} = \lambda_{u,w,v} \neq 0$ . Сложность этих расчетов для одной пары вершин  $u, v$  тривиально равна  $O(n \log n)$ . Наконец, искомый путь определяется как

$$\text{code}(\pi_{u,v}) = \bigvee_w I_w \cdot \text{code}(\pi_{u,w} \cup \pi_{w,v}). \quad (11.7)$$

Код объединенного пути в (11.7) вычисляется как

$$\text{code}(\pi_{u,w} \cup \pi_{w,v}) = \text{code}(\pi_{u,w}) \vee (\text{code}(\pi_{w,v}) \gg q \lambda_{u,w}),$$

где  $x \gg k$  — операция сдвига строки  $x$  на  $k$  разрядов вправо. Сдвиг строки длины  $L$  на переменную величину от 0 до  $L - 1$  позиций выполняется простой схемой сложности  $O(L \log L)$  (лемма 1.1). Умножение и дизъюнкция в формуле (11.7) выполняются со сложностью  $\preccurlyeq n 2^i b$ . Поэтому

$$C(L_i) \preccurlyeq n^3 \log n + n^3 b 2^i i \preccurlyeq 2^i n^3 \log^2 n. \quad (11.8)$$

1.2) На выходе последней подсхемы  $L_s$  имеем описания кратчайших чередующихся путей  $\pi_{u,v}$ , характеризуемых параметрами  $\delta_{u,v}$ ,  $\lambda_{u,v}$ ,  $\text{code}(\pi_{u,v})$ .

Выделим пути, соединяющие свободные вершины из  $U$  и  $V$ , признаками  $I_{i,j} = \delta_{u_i,v_j} \cdot \overline{p_i^u} \cdot \overline{p_j^v}$ . Далее выделим индикатором  $J_{i,j}$  позицию одного увеличивающего пути  $\pi$ , если такой путь найден. Эти действия тривиально выполняются<sup>13</sup> со сложностью  $O(n^2)$ . Сам путь вычисляется как

$$\text{code}(\pi) = \bigvee_{i,j} J_{i,j} \cdot \text{code}(\pi_{u_i,v_j})$$

со сложностью  $\preccurlyeq n^3 \log n$ .

1.3) Окончательно выполняется обновление паросочетания  $M_k \rightarrow M_{k+1}$ . Признаки  $q_{i,j}$  наличия (неориентированных) ребер  $(u_i, v_j)$  в пути  $\pi$  вычисляются по его коду тривиально<sup>14</sup> со сложностью  $\preccurlyeq n^3$ . Теперь положим  $p_{i,j} := p_{i,j} \oplus q_{i,j}$ . Пересчет сопутствующих признаков  $p_i^u$ ,  $p_j^v$ ,  $t[u, v]$  далее выполняется с общей сложностью  $O(n^2)$ . Используя (11.8), получаем

$$\mathsf{C}(S_k) = \sum_{i=1}^s \mathsf{C}(L_i) + O(n^3 \log n) \preccurlyeq l_k n^3 \log^2 n. \quad (11.9)$$

2) В случае, когда граф  $G$  обладает совершенным паросочетанием, паросочетание  $M_{n+1}$  является таковым. Поэтому  $\text{PERM}_n = \bigwedge_i p_i^u$ . Окончательно, ввиду  $\sum_{k=1}^n l_k \asymp n \left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}\right) \sim n \ln n$ , из (11.9) вытекает

$$\mathsf{C}(\text{PERM}_n) \leq \mathsf{C}(S_1) + \dots + \mathsf{C}(S_n) + O(n) \preccurlyeq n^4 \log^3 n.$$

■

- В теореме 11.4 мы сознательно пожертвовали сложность ради простоты конструкции схемы. Лучшие известные оценки схемной сложности перманента получаются переложением алгоритмов поиска максимальных паросочетаний в графах на детерминированные машины Тьюринга и далее на схемы. Лучшую среди известных детерминированных алгоритмов оценку времени работы  $O(n^{2.5} / \log n)$ , в том числе, в модели машин Тьюринга, имеет алгоритм Т. Федера и Р. Мотвани [187]<sup>15</sup>). Известно, что программа с временем выполнения  $T$  на машине Тьюринга моделируется булевой схемой сложности  $O(T \log T)$ , см., например, [330]. Из этого можно заключить, что  $\mathsf{C}(\text{PERM}_n) \preccurlyeq n^{2.5}$ . Соответствующая схема, правда, будет иметь весьма замысловатую структуру.

Примечательно, что согласно результату А. А. Разборова [66]  $\mathsf{C}_{\mathcal{B}_M}(\text{PERM}_n) = n^{\Omega(\log n)}$ , т.е. перманент является примером (монотонной) булевой функции полиномиальной сложности в полном базисе и сверхполиномиальной — в монотонном. Впрочем, Э. Тардош [320] привела аналогичный пример функции с почти экспоненциальной монотонной сложностью  $2^{n^{\Theta(1)}}$ .

<sup>13</sup>Можно использовать префиксные схемы.

<sup>14</sup>Можно начать с вычисления признаков  $\chi_{i,j}$  равенства  $i$ -й компоненты кода  $\text{code}(\pi)$  числу  $j$ .

<sup>15</sup>Результат [187] слегка усиливает оценку сложности  $O(n^{2.5})$ , которую обеспечивают известные алгоритмы Диница—Карзанова [16, 21] и Хопкрофта—Карпа [226].

### Монотонная сложность многочлена циклических блужданий $\vdots \square s$

Как и предыдущий, этот раздел демонстрирует связь (на этот раз, не слишком тривиальную) между комбинаторными характеристиками графов и методами быстрого вычисления связанных с графами многочленов.

Свяжем с произвольным ребром  $e = (i, j)$  полного неориентированного графа  $K_n$  переменную, для которой будем использовать обозначения  $x_e = x_{i,j} = x_{j,i}$ . Рассмотрим многочлен

$$CW_{k,n}(X) = \sum_{i_1 \neq i_2 \neq \dots \neq i_k \neq i_1} x_{i_1, i_2} \cdot x_{i_2, i_3} \cdot \dots \cdot x_{i_{k-1}, i_k} \cdot x_{i_k, i_1},$$

момоны которого соответствуют всевозможным циклическим блужданиям длины  $k$  в графе  $K_n$  (определение допускает многократное прохождение по одним и тем же ребрам).

Сложность вычисления многочленов такого вида монотонными арифметическими схемами оказывается связанной с древесной шириной графов.

*Древесным разложением* графа  $G$  на  $k$  вершинах, пронумерованных числами от 0 до  $k - 1$ , называется дерево  $T$ , каждой вершине  $u$  которого сопоставлено некоторое множество  $V_u \subset \llbracket k \rrbracket$ . При этом

- 0)  $\bigcup_{u \in T} V_u = \llbracket k \rrbracket$ ;
- 1) для любого ребра  $(i, j) \in G$  найдется вершина  $u \in T$ , удовлетворяющая свойству  $i, j \in V_u$ ;
- 2) при любом  $i \in \llbracket k \rrbracket$  множество вершин  $\{u \in T \mid i \in V_u\}$  является связным поддеревом.

Число  $\max_{u \in T} |V_u| - 1$  называется *шириной* разложения  $T$ . *Древесной шириной*  $tw(G)$  графа  $G$  называется минимальная ширина его разложения.

Рассмотрим более общую задачу вычисления многочлена, мономы которого соответствуют образам  $k$ -вершинного графа  $G$  в полном графе  $K_n^*$  на  $n$  вершинах  $\llbracket n \rrbracket$  с петлями при действии всевозможных отображений  $\varphi : \llbracket k \rrbracket \rightarrow \llbracket n \rrbracket$ . Образом вершины  $i$  является вершина  $\varphi(i)$ . Образом ребра  $e = (i, j)$  считается ребро  $\varphi(e) = (\varphi(i), \varphi(j))$ , которое может быть петлей при  $\varphi(i) = \varphi(j)$ . Образы вершин и ребер определяют образ  $\varphi(G)$  графа  $G$ .

Пусть  $\chi_G = \prod_{e \in G} x_e$  обозначает характеристическую функцию графа — произведение переменных, соответствующих его ребрам. Как обычно, полагаем  $\chi_G = 1$ , если в графе нет ребер.

Для произвольного графа  $G$  на вершинах  $\llbracket k \rrbracket$  многочлен *гомоморфизмов* определяется как

$$\text{Hom}_{G,n}(X) = \sum_{\varphi : \llbracket k \rrbracket \rightarrow \llbracket n \rrbracket} \chi_{\varphi(G)}.$$

Определенный выше многочлен циклических блужданий — это частный случай многочлена гомоморфизмов, когда граф  $G$  является циклом длины  $k$  (обозначается  $\mathcal{C}_k$ ), т.е.  $CW_{k,n} = \text{Hom}_{\mathcal{C}_k,n}|_{x_{0,0} = \dots = x_{n-1,n-1} = 0}$  (петли в блужданиях не допускаются).

При вычислении многочлена удобнее руководствоваться древесным разложением специального вида.

*Приведенным древесным разложением* графа  $G$  называется ориентированное к корню древесное разложение, которое является бинарным деревом и содержит вершины только следующих типов:

- а) лист  $u$ , не имеющий входящих ребер, для которого  $|V_u| = 1$ ;
- б) присоединяющая вершина  $u$  с одним входом (из вершины  $z$ ), для которой  $V_z \subset V_u$  и  $|V_u| = |V_z| + 1$ .
- в) исключающая вершина  $u$  с одним выходом (из вершины  $z$ ), для которой  $V_u \subset V_z$  и  $|V_u| = |V_z| - 1$ .
- г) объединяющая вершина  $u$  с двумя выходами (из вершин  $z_1, z_2$ ), при этом  $V_u = V_{z_1} = V_{z_2}$ .

Полагается, что корень  $r$  приведенного разложения является исключающей вершиной и имеет метку  $\emptyset$ .

Можно проверить, что древесное разложение можно преобразовать в приведенное древесное разложение той же ширины и с увеличением числа вершин в  $O(1)$  раз, см. также [243]. Следующий результат фактически содержится в [181].

**Теорема 11.5.** *Пусть граф  $G$  на  $k$  вершинах имеет приведенное древесное разложение  $T$  ширины  $\text{tw}(G)$  из  $t$  вершин. Тогда  $C_{\mathcal{A}_+^R}(\text{Hom}_{G,n}) \leq tk^2 n^{\text{tw}(G)+1}$ .*

► Распределим ребра графа  $G$  между подходящими вершинами дерева  $T$  без повторений, что значит: ребро  $(i, j)$  должно быть отнесено к некоторой вершине  $u \in T$ , для которой  $i, j \in V_u$ . Через  $E_u$  обозначим множество ребер, отнесенных к вершине  $u \in T$ .

Пусть  $T_u$  — поддерево дерева  $T$  с корнем в  $u$ . Положим  $W_u = \bigcup_{z \in T_u} V_z$ . Через  $G_u$  обозначим подграф графа  $G$  на вершинах  $W_u$ , образованный ребрами, отнесенными к вершинам поддерева  $T_u$ .

Далее через  $\{\varphi, \psi\}$  будем обозначать объединение отображений  $\varphi, \psi$  с непересекающимися областями определения.

Последовательно просматривая дерево  $T$  от листьев к корню, будем при каждой вершине  $u \in T$  вычислять семейство многочленов

$$P_{u,\psi}(X) = \sum_{\varphi: W_u \setminus V_u \rightarrow \llbracket n \rrbracket} \chi_{\{\varphi, \psi\}(G_u)} \quad (11.10)$$

для всех  $\psi : V_u \rightarrow \llbracket n \rrbracket$ , фактически используя метод динамического программирования.

- а) Если  $u$  — лист дерева и  $V_u = \{i\}$ , то  $P_{u,i \rightarrow j} = 1$  для всех  $j \in \llbracket n \rrbracket$ .
- б) Если  $u$  — присоединяющая вершина с предшествующей ей вершиной  $z$ , где  $V_u = \{i\} \cup V_z$ , то при каждом  $\psi : W_z \rightarrow \llbracket n \rrbracket$  и  $\psi_j = \{i \rightarrow j, \psi\} : W_u \rightarrow \llbracket n \rrbracket$  полагаем

$$P_{u,\psi_j} = P_{z,\psi} \cdot \prod_{e \in E_u} x_{\psi_j(e)}.$$

Несложно видеть, что при этом удовлетворяется определение (11.10): достаточно заметить, что множество отображений  $\varphi$ , по которым выполняется суммирование, совпадает для  $P_{u,\psi_j}$  и  $P_{z,\psi}$ . Кроме того,  $i \notin W_z$  в силу свойства 2) древесного разложения, значит, отображение  $\{\varphi, \psi_j\}$  корректно определено.

в) Если  $u$  — исключающая вершина с предшествующей ей вершиной  $z$ , где  $V_z = \{i\} \cup V_u$ , то при каждом  $\psi : W_u \rightarrow \llbracket n \rrbracket$  полагаем

$$P_{u,\psi} = \sum_{j=0}^{n-1} P_{z,\{i \rightarrow j, \psi\}} \cdot \prod_{e \in E_u} x_{\psi(e)}.$$

В данном случае (11.10) удовлетворяется непосредственно.

г) Пусть  $u$  — объединяющая вершина с предшествующими ей вершинами  $z_1, z_2$ . Напомним, что  $V_u = V_{z_1} = V_{z_2}$  и  $W_{z_1}, W_{z_2} \subset W_u$ . Тогда при каждом  $\psi : V_u \rightarrow \llbracket n \rrbracket$  положим

$$P_{u,\psi} = P_{z_1,\psi} \cdot P_{z_2,\psi} \cdot \prod_{e \in E_u} x_{\psi(e)}.$$

Для обоснования указанной формулы опять используем свойство 2) связности древесного разложения, в силу которого справедливо  $(W_{z_1} \setminus V_{z_1}) \cap (W_{z_2} \setminus V_{z_2}) = \emptyset$ . Обозначим  $Q_i = W_{z_i} \setminus V_{z_i}$ . Тогда

$$\begin{aligned} P_{z_1,\psi} \cdot P_{z_2,\psi} &= \sum_{\varphi_1: Q_1 \rightarrow \llbracket n \rrbracket} \chi_{\{\varphi_1, \psi\}(G_{z_1})} \cdot \sum_{\varphi_2: Q_2 \rightarrow \llbracket n \rrbracket} \chi_{\{\varphi_2, \psi\}(G_{z_2})} = \\ &= \sum_{\varphi_1, \varphi_2} \chi_{\{\varphi_1, \varphi_2, \psi\}(G_{z_1} \cup G_{z_2})} = \sum_{\varphi: W_u \setminus V_u \rightarrow \llbracket n \rrbracket} \chi_{\{\varphi, \psi\}(G_{z_1} \cup G_{z_2})}. \end{aligned}$$

д) В корне дерева вычисляется многочлен  $P_{r,\emptyset \rightarrow \llbracket n \rrbracket} = \text{Hom}_{G,n}$ .

По построению, в вычислениях при каждой вершине используется не более  $O(k^2 n^{\text{tw}(G)+1})$  операций. ■

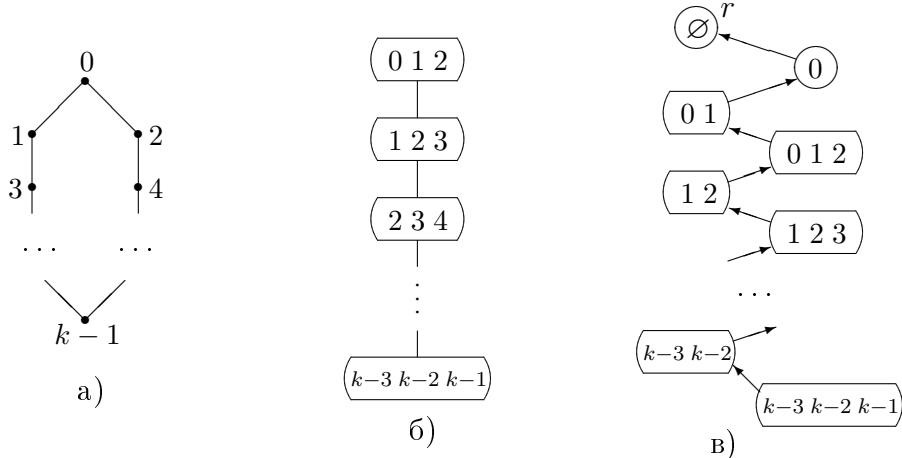


Рис. 11.1: Граф  $C_k$  (а), его древесное разложение (б) и приведенное древесное разложение (в).

Древесная ширина цикла  $C_k$  при  $k \geq 3$  равна<sup>16)</sup> 2. Минимальные по ширине древесное разложение и приведенное разложение графа  $C_k$  показаны на рис. 11.1.

<sup>16)</sup>Легко проверить, что ширину 1 среди связных графов имеют только деревья.

Они содержат  $O(k)$  вершин. Полагая  $x_{i,i} = 0$  для всех  $i \in \llbracket n \rrbracket$ , т.е. исключая петли, получаем

**Следствие 11.1.** *Если  $k \geq 3$ , то  $C_{\mathcal{A}_+^R}(CW_{k,n}) \preccurlyeq (kn)^3$ .*

- В определении многочлена гомоморфизмов, как правило, петли запрещаются. Чтобы их исключить, достаточно связанные с петлями переменные приравнять к нулю.

В работе [245] показано, что оценка теоремы 11.5 неулучшаема по порядку при постоянных  $k$ . Тропические  $(\min, +)$ -аналоги многочлена гомоморфизмов вырождены для многих  $G$ , в частности, для двудольных графов, к которым относятся циклы четной длины. При нечетных  $k$  тропическая версия многочлена  $CW_{n,k}$  содержательна. При  $k = 3$  она совпадает с многочленом выбора треугольника минимального веса и имеет сложность  $\Theta(n^3)$ , например, в силу нижней оценки К. Шнорра из [300]. Методом М. Джеррума и М. Шнира [227] оценка сложности  $\Omega(n^3)$  доказывается при любом постоянном нечетном  $k$ . Таким образом, результат следствия 11.1 в этом случае точен по порядку.

# Глава 12

## Разное

**Метод ортогональных систем. Мультиплекативная сложность булевых функций**  $\boxed{1/2}$

Метод разложения по ортогональным системам функций (в булевой алгебре) предложил Э. И. Нечипорук в работе [54] и успешно применил его сразу в нескольких задачах (минимизация числа контактов в контактно-вентильных схемах, минимизация числа отрицаний в схемах над базисом  $\{\vee, \neg\}$ ). Для иллюстрации идеи мы рассмотрим задачу минимизации числа умножений при вычислении булевых функций  $n$  переменных схемами в базисе Жегалкина  $\mathcal{B}_1 = \{\oplus, \wedge, 1\}$ . Соответствующий функционал сложности обозначим через  $C^\mu$ .

**Теорема 12.1** ([54]).  $C^\mu(\mathcal{P}^n) \lesssim (2 + o(1))2^{n/2}$ .

► Если  $n$  — четно, разобьем множество переменных  $X$  на две группы  $X_1, X_2$  поровну. Представим функцию  $f$  как многочлен Жегалкина по первой группе переменных:

$$f(X) = \bigoplus_{\sigma \in \mathbb{B}^{n/2}} X_1^\sigma \cdot f_\sigma(X_2), \quad X_1^\sigma = \prod_{\sigma_i=1} x_i.$$

Вычислим отдельно все подфункции  $f_\sigma$  как линейные комбинации всевозможных мономов переменных  $X_2$ . Мультиплекативная сложность вычисления всех мономов  $n/2$  переменных не превосходит<sup>1)</sup>  $2^{n/2}$  согласно лемме 3.1. Затем, применяя схему Горнера относительно переменных  $X_1$ , вычислим функцию  $f$  как

$$f(X) = x_1 f_1 \oplus f_0 = x_1(x_2 f_{11} \oplus f_{10}) \oplus (x_2 f_{01} \oplus f_{00}) = \dots, \quad (12.1)$$

используя еще  $2^{n/2} - 1$  умножений.

В случае нечетного  $n$  указанный способ требует примерно  $2^{\lceil n/2 \rceil} + 2^{\lfloor n/2 \rfloor} \approx 2.12 \cdot 2^{n/2}$  умножений. Однако прием разложения по ортогональным системам функций позволяет получить такой же вид асимптотики, как при четном  $n$ .

Разобьем множество переменных  $X$  на три группы  $X_1, X_2, Y$ , где  $|X_1| = |X_2| = m$ . На множестве мономов переменных  $Y$  введем нумерацию с двумя индексами,  $Y^{i,j}$ ,  $0 \leq i, j < 2^{|Y|/2} + 1$ .

<sup>1)</sup>На самом деле, просто равна  $2^{n/2} - n/2 - 1$ .

Положим  $h_i(Y) = \bigvee_j Y^{i,j}$  и  $g_j(Y) = \bigvee_i Y^{i,j}$ . Введенное множество функций позволяет выразить любой моном как  $Y^{i,j} = h_i(Y)g_j(Y)$  и удовлетворяет условиям ортогональности  $h_i(Y)h_{i'}(Y) = 0$  при  $i \neq i'$ ,  $g_j(Y)g_{j'}(Y) = 0$  при  $j \neq j'$ . Тогда

$$f(X) = \bigoplus_{\sigma, \tau \in \mathbb{B}^m} X_1^\sigma \cdot X_2^\tau \cdot f_{\sigma, \tau}(Y) = \bigoplus_i h_i(Y) \cdot \left[ \bigoplus_\sigma X_1^\sigma \cdot \bigoplus_{\tau, j: Y^{i,j} \in f_{\sigma, \tau}(Y)} X_2^\tau \cdot g_j(Y) \right], \quad (12.2)$$

где запись  $Y^{i,j} \in f_{\sigma, \tau}(Y)$  означает, что моном  $Y^{i,j}$  присутствует в многочлене Жегалкина функции  $f_{\sigma, \tau}(Y)$ .

Оценим сложность вычислений по формуле (12.2). Для вычисления всех мономов переменных  $X_2$  и  $Y$  и, как следствие, функций  $h_i$  и  $g_j$ , достаточно  $2^m + 2^{|Y|}$  умножений. Еще  $2^m(2^{|Y|/2} + 1)$  умножений требуется для вычисления всевозможных произведений  $X_2^\tau g_j(Y)$ . Далее вычисление каждой из сумм в квадратных скобках (12.2) выполняется в стиле (12.1) за  $2^m - 1$  умножений, т.е. используя  $(2^{|Y|/2} + 1)(2^m - 1)$  умножений на все суммы. Остается выполнить  $2^{|Y|/2} + 1$  умножений на  $h_i(Y)$ . Требуемая оценка получается, скажем, при  $m \asymp |Y| \asymp n$ . ■

Используя существенно более сложный метод синтеза — комбинацию многоярусного и треугольного представлений булевых функций, а также ряд других идей, — Нечипорук [54] добился асимптотически точного результата<sup>2)</sup>  $C^\mu(\mathcal{P}^n) \sim 2^{n/2}$  (подробное доказательство опубликовано в [56]).

- Ввиду экстремальной сложности асимптотически оптимального метода Нечипорука представляет интерес поиск элементарных и при этом достаточно экономных способов синтеза. Один из них представлен в теореме 12.1. Более быстрый способ предъявила С. Н. Селезнева в [308]. Он опирается на представление

$$\begin{aligned} f(X_1, X_2, X_3, x) &= \bigoplus_{\sigma \in \mathbb{B}^m} X_1^\sigma \cdot \bigoplus_{\tau \in \mathbb{B}^p} X_2^\tau (x \cdot f_{\sigma, \tau, 1}(X_3) \vee \bar{x} \cdot f_{\sigma, \tau, 0}(X_3)) = \\ &= \bigoplus_{\sigma \in \mathbb{B}^m} X_1^\sigma \cdot \bigoplus_{\tau \in \mathbb{B}^p} ((x X_2^\tau \oplus f_{\sigma, \tau, 0}(X_3)) (\bar{x} X_2^\tau \oplus f_{\sigma, \tau, 1}(X_3)) \oplus g_{\sigma, \tau}(X_3)), \end{aligned}$$

где  $|X_1| = m$ ,  $|X_2| = p$  и  $g_{\sigma, \tau} = f_{\sigma, \tau, 0} \cdot f_{\sigma, \tau, 1}$ . При подходящем выборе параметров  $m, p$  и нечетном  $n$  метод имеет сложность  $\sim \sqrt{2} \cdot 2^{n/2}$ . В четном случае оценка из [308] имеет вид  $\sim (3/2) \cdot 2^{n/2}$ , однако дополнительное разложение по ортогональным системам позволяет получить такую же асимптотику сложности, как и для нечетного  $n$ .

### Метод балансировки деревьев. Глубина булевых функций $[U]$

Напомним, что формулы в конечных базисах, как правило, допускают параллельное перестроение: существование формулы сложности  $L$  означает существование формулы глубины  $\asymp \log L$  для той же функции (см. главу 4, теоремы 4.4 и 4.5). Но оказывается, что широкий класс формул допускает практически идеальное распараллеливание: например, с глубиной, близкой к  $\log_2 L$ , в случае бинарных базисов. Таким свойством обладают формулы, в которых при движении от входов к выходам редко чередуются операции.

<sup>2)</sup>Нижняя оценка доказывается элементарно мощностным рассуждением.

*Глубиной альтернирования* пути в схеме или формуле называется число серий из одинаковых многовходовых (двух- и более) операций, на которые он распадается<sup>3)</sup>. Глубиной альтернирования схемы или формулы называется максимальная глубина альтернирования путей от входа к выходу.

С. А. Ложкин [33] заметил, что формулы с малой глубиной альтернирования распараллеливаются практически идеально.

**Теорема 12.2** ([33]). *Если функция  $f$  реализуется формулой над базисом  $\mathcal{B}_0$  сложности  $L$  и глубины альтернирования  $h$ , то  $D_{\mathcal{B}_0}(f) \leq \lceil \log_2 L \rceil + h - 1$ .*



Сергей Андреевич  
Ложкин

Московский университет,  
с 1978

- ▶ С учетом возможности опускания отрицаний на уровень входов, глубина альтернирования формулы над  $\mathcal{B}_0$  — это максимальное число серий из конъюнкций или дизъюнкций в пути от входа к выходу.

Доказательство проведем индукцией по  $h$ . При  $h = 1$  утверждение очевидно. Докажем переход от  $h - 1$  к  $h$ .

Рассмотрим произвольную формулу  $F$  сложности  $L$  и глубины альтернирования  $h$ . Выделим в ней максимальную внешнюю подформулу  $G$ , состоящую из одинаковых операций; пусть  $F_1, \dots, F_s$  — подформулы на входах  $G$ . Таким образом,  $F = F_1 \circ F_2 \circ \dots \circ F_s$  с точностью до расстановки скобок, где  $\circ \in \{\vee, \wedge\}$  — операция подформулы  $G$ . По построению,  $\sum \Phi(F_i) = L$ , и глубина альтернирования формул  $F_i$  не превосходит  $h - 1$ .

Индуктивное предположение позволяет заменить каждую формулу  $F_i$  эквивалентной формулой  $F'_i$  глубины  $d_i \leq \lceil \log_2 \Phi(F_i) \rceil + h - 2$ . Осталось заметить (это центральный момент доказательства), что  $\circ$ -сумму  $F'_1 \circ F'_2 \circ \dots \circ F'_s$  можно вычислить сбалансированным деревом с глубиной корневой вершины

$$d = \left\lceil \log_2 \sum_{i=1}^s 2^{d_i} \right\rceil \leq \left\lceil \log_2 \sum_{i=1}^s 2^{h-1} \Phi(F_i) \right\rceil \leq \lceil \log_2 L \rceil + h - 1.$$

■

Аналогичное утверждение справедливо в любом базисе, бинарные операции в котором ассоциативны и коммутативны, например, в  $\mathcal{B}_1$  или  $\mathcal{A}_+$ .

**Следствие 12.1** ([33]).  $D_{\mathcal{B}_0}(\mathcal{P}_n) \leq n - \log_2 \log n + O(1)$ .

- ▷ Анализируя конструкцию формулы из теоремы 11.2, легко убедиться, что (с учетом опускания отрицаний) формула имеет ограниченную глубину альтернирования. □

- Опираясь на конструкцию О. Б. Лупанова [45] формул асимптотически оптимальной сложности и глубины альтернирования 3, Ложкин в [33] доказал оценку следствия 12.1 в форме

<sup>3</sup>Серии из унарных операций не считаются.

$D_{B_0}(\mathcal{P}_n) \leq \lceil n - \log_2 \log_2 n + o(1) \rceil + 3$ . Это совсем немного уступает его же рекордной оценке (11.5).

Независимо от Ложкина, Г. Гувер, М. Клауэ и Н. Пиппенджера [225] предложили метод минимизации глубины формул при введении ограничения на ветвление выходов элементов. Этот метод оказывается двойственным к методу Ложкина (по существу, то же самое преобразование выполняется с перевернутой формулой), что подробно разъясняется в работе С. Б. Гашкова [9].

### Градиентный метод. Глубина схем для многократного сложения $\boxed{\nabla} \boxed{U}$

Изложение было бы неполным без упоминания градиентного метода, который играет заметную роль в задачах дискретной оптимизации. К сожалению, выбранная модель (схемы и формулы без ограничения на глубину) не позволяет в должной мере проиллюстрировать разнообразие вариаций и приложений метода. Ограничимся лишь одним примером, еще один пример будет рассмотрен ниже применительно к модели схем ограниченной глубины, см. на стр. 144.

Вернемся к задаче минимизации глубины дерева из компрессоров в задаче многократного сложения. Метод теоремы 4.3 оптимален в асимптотическом смысле, но добавочная константа  $O(1)$ , скрывающаяся в оценке глубины, очень велика (схема использует в несколько раз больше компрессоров, чем минимально необходимо).

А что если просто составить схему, последовательно присоединяя компрессоры каждый раз на минимально возможную глубину (это и есть градиентный метод)? Будет ли построенная схема эффективна? По крайней мере, в случае наиболее популярного компрессора  $FA_3$  ответ утвердительный, как показано автором в [80]. Напомним, что компрессор  $FA_3$  характеризуется наборами глубин входов  $(0, 0, 1)$  и выходов  $(2, 3)$ , а его характеристический многочлен (4.10) имеет корень  $\lambda \approx 1.2056$ . Пример градиентной схемы суммирования восьми чисел показан на рис. 12.1 (сравнить со схемой рис. 4.1).

**Теорема 12.3.**  $D_{FA_3}(n) \leq \log_\lambda n + 4.6$ .

► Рассмотрим градиентную схему  $S$  на  $n$  входах, ее глубину обозначим через  $D$ . Будем считать компрессор расположенным на уровне  $d$ , если его выходы имеют глубины  $d+2$  и  $d+3$ . Для произвольного множества с повторениями  $T \subset \mathbb{N}_0$  положим  $\sigma(T) = \sum_{t \in T} \lambda^t$  (сумма потенциалов). Через  $S_r$  обозначим схему, образованную компрессорами схемы  $S$ , расположенными на уровнях, меньших  $r$ . Через  $T(S_r)$  обозначим множество (с повторениями) глубин выходов схемы  $S_r$ , в котором числа, меньшие  $r$ , заменены на  $r$  (множество содержит только числа  $r$ ,  $r+1$  и  $r+2$ ). Положим для краткости  $\sigma(S_r) = \sigma(T(S_r))$ . В силу свойства корня характеристического многочлена компрессора,

$$n = \sigma(S_0) \leq \sigma(S_1) \leq \dots \leq \sigma(S_{d-2}) = \lambda^D + \lambda^{D-1}. \quad (12.3)$$

Если бы все компрессоры можно было состыковать без зазоров, то цепочка (12.3) состояла бы сплошь из равенств, но это, конечно, невозможно. Уже

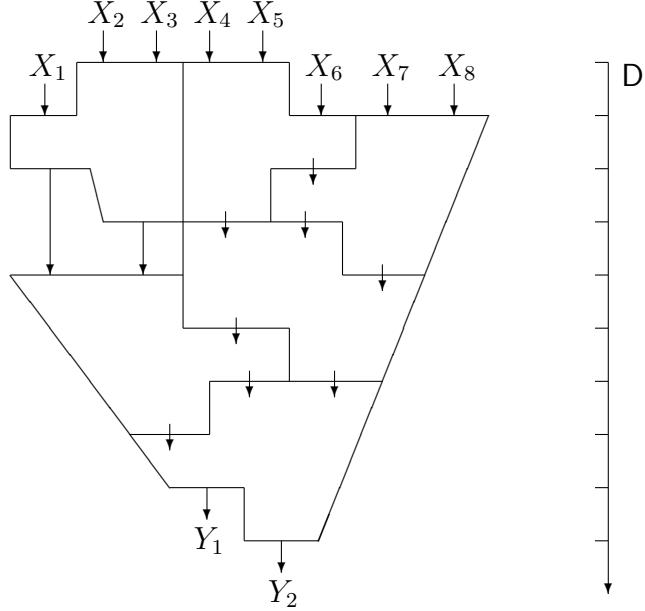


Рис. 12.1: Градиентная схема из (3, 2)-компрессоров

на первом шаге треть входов приходится «опустить» на глубину 1. Чем больше таких зазоров образуется на последующих глубинах, тем больше величина разностей  $\sigma(S_{r+1}) - \sigma(S_r)$ , и тем дальше сумма потенциалов промежуточных сумм отклоняется от  $n$ . К счастью, оказывается, что уровень 0 схемы  $S$  — особенный: на каждом из последующих уровней опускается не более двух промежуточных сумм.

**Лемма 12.1** ([80]). *Пусть  $r > 0$ , а  $m_0(r)$ ,  $m_1(r)$  и  $m_2(r)$  — соответственно количество чисел  $r$ ,  $r + 1$  и  $r + 2$  во множестве  $T(S_r)$ . Тогда выполнено:*

$$m_0(r) \leq 2m_1(r) + 2, \quad m_1(r) \leq 3m_0(r)/2 + 2m_2(r), \quad m_2(r) \leq m_1(r). \quad (12.4)$$

▷ Воспользуемся индукцией по  $r$ . Схема содержит  $k = \lfloor n/3 \rfloor$  компрессоров уровня 0, поэтому

$$m_0(1) = n \bmod 3, \quad m_1(1) = m_2(1) = k,$$

и неравенства леммы выполнены при  $r = 1$ . Для доказательства индуктивного перехода от  $r$  к  $r + 1$  рассматривается два случая. Далее  $m_i$  обозначает  $m_i(r)$ .

В случае  $m_0 \leq 2m_1$  на уровне  $r$  размещается  $k = \lfloor m_0/2 \rfloor$  компрессоров, поэтому

$$m_0(r + 1) = m_1 - k + (m_0 \bmod 2), \quad m_1(r + 1) = m_2 + k, \quad m_2(r + 1) = k.$$

В случае  $m_0 \geq 2m_1$  на уровне  $r$  расположено  $k = \lfloor (m_0 + m_1)/3 \rfloor$  компрессоров, поэтому

$$m_0(r + 1) = (m_0 + m_1) \bmod 3, \quad m_1(r + 1) = m_2 + k, \quad m_2(r + 1) = k.$$

Несложно проверить, что в обоих случаях выполнение неравенств (12.4) для величин  $m_i(r+1)$  автоматически следует из выполнения неравенств для величин  $m_i$ .  $\square$

Для оценки глубины схемы  $S$  существенно только первое неравенство леммы 12.1. Оно означает, что на любом уровне  $r \geq 1$  схема содержит  $\min\{\lfloor m_0(r)/2 \rfloor, m_1(r)\}$  компрессоров, следовательно,  $\sigma(S_{r+1}) - \sigma(S_r) \leq 2(\lambda^{r+1} - \lambda^r)$ . Таким образом,

$$\begin{aligned} \lambda^D + \lambda^{D-1} = \sigma(S_{D-2}) &= \sum_{r=1}^{D-3} (\sigma(S_{r+1}) - \sigma(S_r)) + (\sigma(S_1) - \sigma(S_0)) + \sigma(S_0) \leq \\ &\leq 2(\lambda^{D-2} - \lambda) + (\lambda - 1)(n/3 + 2) + n < 2\lambda^{D-2} + (\lambda + 2)n/3. \end{aligned}$$

Окончательно имеем  $\lambda^{D-2} \leq \frac{\lambda+2}{3(\lambda^2+\lambda-2)} n$ , откуда  $D \leq \log_\lambda n + 4.6$ .  $\blacksquare$

Указанную верхнюю оценку полезно сравнить с вытекающей из леммы 4.1 нижней оценкой  $D_{FA_3}(n) > \log_\lambda n - 3.8$ . Отметим, что сложность градиентной схемы составляет приблизительно  $5tn + O(n)$ , и не уступает стандартной схеме из сумматоров. Слагаемое  $O(n)$  здесь отвечает росту разрядности промежуточных сумм по мере увеличения глубины. Малость этого слагаемого следует из того, что число компрессоров убывает от уровня к уровню в геометрической прогрессии, а разрядность растет линейно.

- Рассуждая чуть более аккуратно, автор в [80] получил верхнюю оценку глубины градиентной схемы схемы  $D_{FA_3} < \log_\lambda n - 0.8$ . Вкупе с уточненной нижней оценкой  $D_{FA_3}(n) > \log_\lambda n - 2.7$  это означает, что градиентный метод заведомо проигрывает по глубине не более единицы оптимальному методу, а при многих  $n$  и вовсе доказуемо оптимален. Вопрос об эффективности градиентных схем для других типов компрессоров, по-видимому, не исследован.

# Глава 13

## Схемы ограниченной глубины

### Введение

Схемы ограниченной глубины — это схемы над бесконечным базисом, включающим функции с неограниченным числом аргументов<sup>1)</sup>. Такие схемы стали активно изучаться приблизительно с 1980-х гг., прежде всего в направлении получения высоких нижних оценок сложности. Уже из пионерских работ [97, 118, 195] выяснилось, что модель позволяет получать сверхполиномиальные нижние оценки сложности конкретных функций в полных булевых базисах<sup>2)</sup>.

К наиболее активно изучаемым моделям схем ограниченной глубины относятся *AC*-схемы — схемы над базисом  $\{\vee, \wedge, \neg\}$  (дизъюнкции и конъюнкции с неограниченным числом входов)<sup>3)</sup>. Они обобщают схемы над стандартным базисом  $\mathcal{B}_0$ . Можно полагать, что все отрицания в *AC*-схеме применяются только ко входам переменных и в подсчете глубины не учитываются. *AC*[ $\oplus$ ]-схемы — это схемы в более широком базисе  $\{\vee, \wedge, \oplus, \neg, 1\}$  — обобщение схем над базисом  $\mathcal{B}_2$ . Также рассматриваются схемы, в которых дополнительно используются элементы, реализующие суммы по числовому модулю, симметрические, пороговые функции. Сложность булева оператора  $F$  при реализации *AC*-схемами и, соответственно, *AC*[ $\oplus$ ]-схемами глубины  $d$  будем обозначать через  $C_d^{AC}(F)$  и  $C_d^{AC[\oplus]}(F)$ . В случае схем с чередованием слоев операций, например,  $\vee\wedge\vee$ -схем<sup>4)</sup>, для сложности будем использовать обозначение наподобие  $C_{\vee\wedge\vee}(F)$ .

Многовходовый аналог арифметических схем — схемы над базисом  $\{\Sigma, \Pi\}$ , где  $\Sigma$  — линейные комбинации над кольцом  $R$ , а  $\Pi$  — произведения неограниченного числа переменных. Обозначение для сложности вычисления оператора  $F$  такими схемами будем строить в виде  $C_{\Sigma\Pi\Sigma}^R(F)$  (пример для схем глубины 3).

<sup>1)</sup> Схемами ограниченной глубины также называют схемы над конечными базисами, имеющие ограниченную глубину альтернирования (число перемен операции в цепочке вычислений).

<sup>2)</sup> Вскоре аналогичные результаты были получены для сложности арифметических схем ограниченной глубины над конечными полями, а совсем недавно в работе [256] — для схем над бесконечными полями.

<sup>3)</sup> Название апеллирует к тому факту, что такие схемы используются в определении классов сложности  $AC^k$ .

<sup>4)</sup> Это монотонные схемы глубины 3 с дизъюнкторами на первом и третьем слоях и конъюнкторами на среднем слое.

Простейший и исторически самый первый<sup>5)</sup> вид схем из многовходовых элементов — *линейные (вентильные) схемы*. В таких схемах используется единственная операция сложения (в некоторой коммутативной полугруппе). Поскольку число элементов в качестве меры сложности линейных схем не имеет смысла, подсчитывается число ребер в графе схемы. Сложность линейного оператора с матрицей  $A$  при реализации линейными схемами глубины  $d$  в базисе  $\{+\}$  обозначается через  $W_d^+(A)$ . Обозначение  $W_d(A)$  применяется для сложности универсальных схем, правильно вычисляющих матрицу<sup>6)</sup>  $A$  вне зависимости от выбора полугруппы.

Модель линейных схем эквивалентна модели аддитивных схем: аддитивная схема перестраивается в линейную с сохранением порядка сложности: для произвольной  $m \times n$  матрицы  $A$  выполнено  $W(A) \asymp L(A) + m + n$ , где обозначение  $W(A)$  относится к сложности схем без ограничения на глубину. Именно ограничение глубины делает модель линейных схем содержательной.

В более общих булевых и арифметических моделях схемы глубины 2 представляют вырожденный случай, соответствующий нормальным формам (ДНФ, КНФ) или стандартной записи многочлена. Но для линейных схем нетривиальные задачи начинаются уже в глубине 2. В булевом случае вычисление матрицы схемой глубины 2 интерпретируется как построение покрытия матрицы *прямоугольниками* (сплошь единичными подматрицами). Подробнее о сложности линейных схем см. в [233].

Далее мы приведем несколько результатов, иллюстрирующих универсальные методы синтеза и отражающих особенности модели схем ограниченной глубины, и также рассмотрим еще один общий метод, метод сечения.

### Линейные схемы глубины 2 для матрицы Серпинского. Градиентный метод

Начнем с двух результатов о сложности вычисления матриц Серпинского линейными схемами глубины 2. Первый из них использует идею деления пополам для построения эффективных универсальных схем, а во втором применяется довольно общий комбинаторный вариант градиентного метода и строятся практически оптимальные линейные  $\nabla$ -схемы.

Последовательность матриц Серпинского<sup>7)</sup> рекурсивно определяется как

$$S_1 = [1], \quad S_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad S_{2n} = \begin{bmatrix} S_n & 0 \\ S_n & S_n \end{bmatrix}. \quad (13.1)$$

Матрицы Серпинского — популярный объект в комбинаторике и теории сложности, о некоторых их свойствах см., например, в [233].

Минимальная линейная схема строится прямо по определению (13.1):  $W(S_n) = W^\vee(S_n) = n \log_2(n/2)$ . Точность оценки доказана С. Н. Селезневой [77]

<sup>5)</sup> Введен в рассмотрение О. Б. Лупановым в работе [41] 1956 г.

<sup>6)</sup> Как и в случае аддитивных схем, удобно считать, что линейная схема вычисляет матрицу преобразования.

<sup>7)</sup> В англоязычной литературе больше распространен термин *disjointness matrix*.

и независимо Дж. Бойар и М. Файндом [153]. Простой способ построения схем глубины 2 указан в работе С. Юкны и автора [233].

**Теорема 13.1** ([233]).  $W_2(S_n) \preccurlyeq n^{\log_2(\sqrt{2}+1)} \prec n^{1.28}$ .

► Задача состоит в том, чтобы предъявить покрытие матрицы прямоугольниками малого суммарного веса<sup>8)</sup>. Следуя (13.1), составим покрытие матрицы  $S_{2n}$  из (трех идентичных) покрытий подматриц  $S_n$ . Покрытие будет состоять из прямоугольников с соотношением сторон 1 : 1 (квадратов) и 1 : 2 (брикетов). В каждой тройке подобных прямоугольников, мы объединяем два вдоль длинной стороны, см. рис. 13.1<sup>9)</sup>.

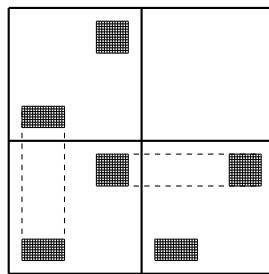


Рис. 13.1: Покрытие матрицы Серпинского

Так квадрат размера  $u \times u$  покрытия матрицы  $S_n$  порождает такой же квадрат и брикет размера  $u \times 2u$  покрытия матрицы  $S_{2n}$ . Брикет размера  $v \times 2v$  из покрытия матрицы  $S_n$  порождает такой же брикет и квадрат размера  $2v \times 2v$  в покрытии матрицы  $S_{2n}$ . Пусть  $u_n$  и  $v_n$  означают сумму длин сторон квадратов и сумму длин коротких сторон брикетов из покрытия матрицы  $S_n$ . Отталкиваясь от  $u_2 = v_2 = 1$ , при  $n = 2^r$  получаем

$$\begin{bmatrix} u_{2n} \\ v_{2n} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_n \\ v_n \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}^r \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = P \cdot \begin{bmatrix} 1 + \sqrt{2} & 0 \\ 0 & 1 - \sqrt{2} \end{bmatrix}^r \cdot P^{-1} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

где  $P$  — некоторая обратимая  $2 \times 2$  матрица, поскольку матрица  $\begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}$  имеет собственные значения  $1 \pm \sqrt{2}$ . Как следствие,  $u_n + v_n \asymp n^{\log_2(\sqrt{2}+1)}$ . ■

- В работе [122] предложено очень нетривиальное обобщение метода теоремы 13.1, позволившее понизить оценку сложности до  $W_2(S_n) \prec n^{1.26}$ ; см. также [310], где получена несколько более аккуратная оценка.

Авторы работы [166] заметили, что в классе линейных  $\mathcal{V}$ -схем глубины 2 верхняя оценка теоремы 13.1 может быть улучшена при помощи градиентного метода.

<sup>8</sup>Вес прямоугольника — эта суммарное число строк и столбцов в нем.

<sup>9</sup>Конечно, прямоугольники не обязательно состоят из смежных строк и столбцов.

Рассмотрим семейство  $\mathcal{F}$  подмножеств конечного множества  $X$ . По определению, семейство  $\mathcal{F}$  содержит покрытие размера  $t$ , если  $X = \bigcup_{i=1}^t F_i$  при некоторых  $F_i \in \mathcal{F}$  (т.е. все элементы  $X$  накрываются какими-то  $t$  множествами из  $\mathcal{F}$ ). Следующий результат фактически доказан А. А. Сапоженко в [75], но больше известен как лемма Ловаса—Стейна. Нам удобнее использовать вариант Ш. Стейна [314], см. также [230].

**Лемма 13.1** ([314]). *Если  $|F| \leq s$  для любого  $F \in \mathcal{F}$ , и любой элемент  $x \in X$  принадлежит не менее чем  $r$  множествам из  $\mathcal{F}$ , то семейство  $\mathcal{F}$  содержит покрытие размера  $t \leq (1 + \ln s)|\mathcal{F}|/r$ .*

▷ Рассмотрим градиентный метод построения покрытия. На каждом очередном шаге мы присоединяем к покрытию множество  $F \in \mathcal{F}$ , покрывающее максимальное число еще не покрытых элементов.

Пусть в начальный момент  $C_0 = \emptyset$  и  $X_0 = X$ . Если на  $i$ -м шаге выбрано множество  $F_i$ , то полагаем  $C_i = C_{i-1} \cup \{F_i\}$  и  $X_i = X_{i-1} \setminus F_i$ . Процесс заканчивается на шаге  $t$  при условии  $X_t = \emptyset$ . Тогда  $C_t$  — искомое покрытие.

Заметим, что  $X_j = \bigcup_{i=j+1}^t (F_i \cap X_{i-1})$ . По условию, число покрываемых на каждом шаге новых элементов не возрастает:

$$s \geq |F_1 \cap X_0| \geq \dots \geq |F_i \cap X_{i-1}| \geq \dots$$

Обозначим через  $t_k$  число множеств  $F_i \cap X_{i-1}$  мощности  $k$ . Тогда  $t = \sum_{k=1}^s t_k$ . Введем обозначение  $w_k = |X_{j_k}|$  для  $j_k = t_s + t_{s-1} + \dots + t_k$  и формально положим  $j_{s+1} = 0$ . Поскольку ни одно из множеств  $F \in \mathcal{F}$  не содержит более  $k - 1$  элементов из  $X_{j_k}$ , справедливо

$$rw_k \leq (k - 1)|\mathcal{F}|. \quad (13.2)$$

Ввиду  $t_k = (w_{k+1} - w_k)/k$ , применяя (13.2), окончательно получаем

$$\begin{aligned} t &= \sum_{k=1}^s t_k = \sum_{k=1}^s \frac{w_{k+1} - w_k}{k} = \frac{w_{s+1}}{s} + \sum_{k=2}^s \frac{w_k}{(k-1)k} - w_1 \leq \\ &\leq \frac{|\mathcal{F}|}{r} \left( 1 + \frac{1}{2} + \dots + \frac{1}{s} \right) \leq \frac{|\mathcal{F}|}{r} (1 + \ln s). \end{aligned}$$

Последний переход использует известное соотношение между гармоническими числами и натуральными логарифмами<sup>10</sup>.  $\square$

Лемма показывает, что градиентный метод гарантирует результат, не слишком сильно уступающий оптимистической оценке  $|\mathcal{F}|/r$ .

Далее нам удобно использовать способ определения матрицы Серпинского как матрицы непересечений. Строки и столбцы  $n \times n$  матрицы  $S_n$ ,  $n = 2^k$ , индексируются всевозможными подмножествами множества  $X$  из  $k$  элементов. Положим  $S_n[A, B] = (A \cap B = \emptyset)$ . Легко проверить, что это определение эквивалентно (13.1).

---

<sup>10</sup>А именно,  $1 + \frac{1}{2} + \dots + \frac{1}{s} \leq \ln s + \gamma + \frac{1}{2s}$ , где  $\gamma = 0.577\dots$  — постоянная Эйлера.

**Теорема 13.2** ([166]).  $W_2^\vee(S_n) \preccurlyeq n^{\log_2(9/4)} \log^{7/2} n \prec n^{1.17}$ .

► Разобьем матрицу  $S_n$  на  $(k+1)^2$  подматриц  $S_n^{a,b}$ ,  $0 \leq a, b \leq k$ , где  $S_n^{a,b}$  образована строками, занумерованными множествами мощности  $a$ , и столбцами, занумерованными множествами мощности  $b$ . Построим покрытия матриц  $S_n^{a,b}$  независимо. Далее полагаем  $a+b \leq k$ , т.к. в остальных случаях подматрицы состоят из нулей.

Рассмотрим семейство  $\mathcal{F} = \{F_R\}$  прямоугольников в  $S_n^{a,b}$ , образованных строками  $A \subset R$  и столбцами  $B \subset \bar{R}$ , где  $|R| = (a-b+k)/2$  и  $\bar{R} = X \setminus R$ .

Любой прямоугольник  $F_R$  имеет размер  $C_{(a-b+k)/2}^a \times C_{(b-a+k)/2}^b$ , и каждый элемент матрицы  $S_n^{a,b}$  накрывается  $r = C_{k-a-b}^{(k-a-b)/2}$  прямоугольниками из  $\mathcal{F}$ . При этом  $|\mathcal{F}| = C_k^{(a-b+k)/2}$ . Тогда согласно лемме 13.1

$$t \preccurlyeq \ln(C_{(a-b+k)/2}^a \cdot C_{(b-a+k)/2}^b) \cdot |\mathcal{F}|/r \preccurlyeq k^{3/2} \cdot C_k^{(a-b+k)/2} \cdot 2^{a+b-k}$$

прямоугольников образуют покрытие матрицы  $S_n^{a,b}$ . Таким образом, при  $a \geq b$

$$W_2^\vee(S_n^{a,b}) \preccurlyeq k^{3/2} \cdot C_k^{(a-b+k)/2} \cdot 2^{a+b-k} \cdot C_{(a-b+k)/2}^a = k^{3/2} \cdot 2^{-2u} \cdot C_k^u \cdot C_{k-u}^a, \quad (13.3)$$

где  $u = (k-a-b)/2$ .

Напомним известное соотношение  $C_n^m \leq 2^{nH(m/n)}$ , где  $H(x) = -x \log_2 x - (1-x) \log_2(1-x)$  — функция двоичной энтропии, определенная на отрезке  $x \in [0, 1]$ <sup>11</sup>). Вводя обозначение  $\alpha = u/k$ , оценка (13.3) продолжается как

$$W_2^\vee(S_n^{a,b}) \preccurlyeq k^{3/2} \cdot 2^{(H(\alpha)-2\alpha)k} \cdot C_{k-u}^{(k-u)/2} \preccurlyeq k^{3/2} \cdot 2^{(1+H(\alpha)-3\alpha)k}.$$

Несложно проверить, что функция  $H(x) - 3x$  принимает максимальное значение  $\log_2(9/8)$  при  $x = 1/9$ . ■

- Верхняя оценка теоремы 13.2 исключительно близка к нижней оценке  $W_2^\vee(S_n) \succ n^{1.16}$  из [233]. Более того, методы доказательства обеих оценок позволяют установить значение сложности  $W_2^\vee(S_n) \approx n^\alpha$  с точностью до множителя  $(\log n)^{O(1)}$ , где показатель  $\alpha$  определяется как решение задачи поиска экстремума конкретной функции [166].

### $\oplus \wedge \oplus$ -схемы для булевых функций $\boxed{\cdot} \vdash \boxed{c}$

Путь к быстрому вычислению нередко лежит через построение экономных покрытий (например, булева куба). Но в модели схем ограниченной глубины, в силу присущего ей параллелизма, так происходит значительно чаще. В качестве примера рассмотрим задачу синтеза  $\oplus \wedge \oplus$ -схем, решенную С. Н. Селезневой [79] в части определения порядка сложности класса функций  $n$  переменных. Нижняя оценка  $\Omega(2^n/n^2)$  устанавливается простым мощностным рассуждением [78]. Доказательство верхней связано с построением покрытия булева куба фрагментами сфер радиуса 2.

<sup>11</sup>На концах отрезка полагается по непрерывности  $H(0) = H(1) = 0$ .



Светлана  
Николаевна  
Селезнева  
Московский университет,  
с 1998

Напомним, что единичная сфера в булевом кубе с центром  $\alpha$  — это множество векторов, отличающихся от  $\alpha$  значением в одной позиции. Те из наборов, которые получаются заменой 0 на 1, образуют *положительную полусферу*, остальные — *отрицательную*. Как обычно, семейство  $T$  подмножеств  $Q \subset \mathbb{B}^n$  называется *покрытием* (под)множества  $S \subset \mathbb{B}^n$ , если  $S \subset \bigcup_{Q \in T} Q$ .

Дж. Купер, Р. Эллис и Э. Канг [174] построили покрытия булева куба полусферами оптимальной по порядку мощности. Обозначим через  $\mathbb{B}_+^n$  (соответственно  $\mathbb{B}_-^n$ ) булев куб  $\mathbb{B}^n$  без нулевого (единичного) набора.

**Лемма 13.2** ([174]). *Существует покрытие  $T_n$  булева куба  $\mathbb{B}_+^n$  положительными полусферами мощности  $|T_n| \asymp 2^n/n$ .*

▷ Сначала установим существование частичного покрытия  $T \subset \mathbb{B}^n$  булева куба с несколько худшими характеристиками. Обозначим через  $\bar{T}$  множество точек, не накрываемых частичным покрытием  $T$ . Удобно ввести специальную меру,  $\delta$ -мощность частичного покрытия:  $|T|_\delta = |T| + \delta|\bar{T}|$  для произвольного  $\delta \geq 1/n$ .

*I.* Докажем существование частичного покрытия  $T$ , для которого

$$|T|_\delta \leq (2 \ln(\delta n) + 1)2^n/n. \quad (13.4)$$

Зададим случайное множество точек  $T_0$  условием: вероятность того, что точка  $j$ -го слоя куба принадлежит  $T_0$ , равна  $p_j = \min\{\ln(\delta n)/(j+1), 1\}$ . Рассмотрим покрытие  $T$ , состоящее из полусфер с центрами из  $T_0$ .

По построению, все точки  $\ln(\delta n)$  нижних слоев куба, кроме нулевого, накрыты покрытием  $T$ . При  $j+1 > \ln(\delta n)$  вероятность того, что точка  $j$ -го слоя куба не принадлежит никакой из полусфер, оценивается как  $(1 - p_{j-1})^j \leq 1/(\delta n)$  ввиду справедливого при  $x < 1$  неравенства  $(1-x)^{1/x} \leq 1/e$ . Тогда для математического ожидания  $\delta$ -мощности покрытия справедливо

$$\begin{aligned} \mathbf{E}[|T|_\delta] &= \mathbf{E}[|T|] + \delta \mathbf{E}[|\bar{T}|] \leq \sum_{j=0}^n p_j C_n^j + \delta 2^n \frac{1}{\delta n} \leq \\ &\leq \frac{\ln(\delta n)}{n+1} \cdot \sum_{j=0}^n C_{n+1}^{j+1} + \frac{2^n}{n} \leq (2 \ln(\delta n) + 1) \frac{2^n}{n}. \end{aligned}$$

*II.* Заметим, что если  $T_1$  — покрытие куба  $\mathbb{B}_+^{n_1}$ , а  $T_2$  — частичное покрытие куба  $\mathbb{B}_+^{n_2}$ , то множество  $T_1 \Delta T_2 := (\mathbb{B}^{n_1} \times T_2) \cup (T_1 \times \bar{T}_2)$  является покрытием куба  $\mathbb{B}_+^{n_1+n_2}$ .

Руководствуясь этим правилом, построим индуктивно покрытие куба  $\mathbb{B}_+^n$  мощности  $\leq b2^n/n$  при подходящей константе  $b \geq 1$ . База индукции  $n = 1$  trivialно выполнена. Докажем переход от  $n - 1$  к  $n$ .

Положим  $n_1 = \lfloor n/2 \rfloor$  и  $n_2 = \lceil n/2 \rceil$ . Рассмотрим покрытие  $T_1 \Delta T_2$ , где в качестве  $T_1$  выбрано существующее по предположению индукции покрытие куба  $\mathbb{B}_+^{n_1}$ , а  $T_2$  — частичное покрытие куба  $\mathbb{B}_+^{n_2}$ , удовлетворяющее (13.4) при  $\delta = b/n_1$ . Тогда

$$|T_1 \Delta T_2| = 2^{n_1} |T_2| + |T_1| |\overline{T_2}| \leq 2^{n_1} |T_2|_\delta \leq (2 \ln(\delta n_2) + 1) \frac{2^n}{n_2} \leq 2(2 \ln(2b) + 1) \frac{2^n}{n}.$$

При выборе  $b = 16$  выполнено  $2(2 \ln(2b) + 1) < b$ , тем самым индуктивный переход доказан.  $\square$

Обозначим через  $S^+(\alpha)$  и  $S^-(\alpha)$  соответственно положительную и отрицательные полусфера радиуса 1 с центром  $\alpha$ .

Пусть  $t = \lfloor n/2 \rfloor$ . Представим булев куб в виде  $\mathbb{B}^n = \mathbb{B}^t \times \mathbb{B}^{n-t}$ . Для произвольного набора  $\alpha \in \mathbb{B}^n$  положим  $\alpha = (\alpha^0, \alpha^1)$ , где  $\alpha^0 \in \mathbb{B}^t$  и  $\alpha^1 \in \mathbb{B}^{n-t}$ . Обозначим  $Q(\alpha) = S^-(\alpha^0) \times S^+(\alpha^1)$ ; назовем это множество *квадрантом* с центром  $\alpha \in \mathbb{B}^n$  (по существу, это четверть сферы радиуса 2). Из леммы 13.2 вытекает

**Следствие 13.1.** *Существует покрытие  $T_n$  булева куба  $\mathbb{B}_\pm^n = \mathbb{B}_-^t \times \mathbb{B}_+^{n-t}$  квадрантами мощности  $|T_n| \asymp 2^n/n^2$ .*

▷ Искомое покрытие получается как прямое произведение покрытия куба  $\mathbb{B}_-$  отрицательными полусферами и покрытия куба  $\mathbb{B}_+^{n-t}$  положительными полусферами.  $\square$

**Теорема 13.3** ([79]).  $C_{\oplus \wedge \oplus}(\mathcal{P}_n) \preccurlyeq 2^n/n^2$ .

► Представим произвольную булеву функцию  $f(X, Y)$ ,  $|X| = t$ ,  $|Y| = n - t$  ее многочленом Жегалкина

$$f(X, Y) = \bigoplus_{\sigma \in \mathbb{B}^t} \bigoplus_{\tau \in \mathbb{B}^{n-t}} f_{\sigma, \tau} X^\sigma Y^\tau, \quad X^\sigma = \prod_{\sigma_i=1} x_i, \quad Y^\tau = \prod_{\tau_i=1} y_i.$$

Пусть  $f^*$  — часть функции  $f$ , состоящая из мономов, вектор-степени которых принадлежат  $\mathbb{B}_\pm^n$ :  $f^*(X, Y) = \bigoplus_{\sigma \in \mathbb{B}_-^t, \tau \in \mathbb{B}_+^{n-t}} f_{\sigma, \tau} X^\sigma Y^\tau$ .

Возьмем покрытие  $T$  булева куба  $\mathbb{B}_\pm^n$  квадрантами, гарантированное следствием 13.1. Пусть  $\{\alpha_1, \dots, \alpha_{|T|}\}$  — множество центров квадрантов, занумерованных в порядке уменьшения веса, и при равенстве весов — в порядке уменьшения веса младшего поднабора  $\alpha^0$ .

Далее выполним  $|T|$  шагов преобразования представления функции  $f^*$ . После каждого  $j$ -го шага будет выполнено  $f^* = P_j \oplus R_j$ , где  $P_j$  — сумма  $2j$  мультиаффинных функций<sup>12)</sup>, а «остаток»  $R_j$  имеет вид

$$R_j = \bigoplus_{\gamma \in \mathbb{B}_\pm^n \setminus \bigcup_{i=1}^j Q(\alpha_j)} c_{j, \gamma} X^{\gamma^0} Y^{\gamma^1}. \quad (13.5)$$

<sup>12</sup>Мультиаффинные функции — это произведения аффинных булевых функций, т.е. ровно такие функции, которые допускают  $\wedge \oplus$ -представление.

(Заметим, что для всех наборов  $\gamma$  под знаком суммы выполнено  $|\gamma| \leq |\alpha_j|$ .) Перед началом процедуры  $P_0 = 0$  и  $R_0 = f^*$ .

Очередной  $j$ -й шаг заключается в следующем. Отталкиваясь от представления  $f^* = P_{j-1} \oplus R_{j-1}$ , выделим из многочлена  $R_{j-1}$  мономы, вектор-степени которых принадлежат  $Q(\alpha_j)$ , и положим

$$R'_j = \bigoplus_{\gamma \in Q(\alpha_j)} c_{j-1,\gamma} X^{\gamma^0} Y^{\gamma^1}.$$

Если для любого  $\sigma \in S_-(\alpha_j^0)$  определить линейную функцию

$$g_\sigma(Y) = \bigoplus_{\tau \in S_+(\alpha_j^1)} c_{j-1,(\sigma,\tau)} Y^{\tau \oplus \alpha_j^1},$$

то получим

$$R'_j = Y^{\alpha_j^1} \bigoplus_{\sigma \in S_-(\alpha_j^0)} X^\sigma g_\sigma(Y).$$

Положим

$$A_j = X^{\alpha_j^0} Y^{\alpha_j^1} \oplus Y^{\alpha_j^1} \cdot \prod_{\sigma \in S_-(\alpha_j^0)} (X^{\sigma \oplus \alpha_j^0} \oplus g_\sigma(Y)), \quad \begin{aligned} P_j &= P_{j-1} \oplus A_j, \\ R_j &= R_{j-1} \oplus A_j \end{aligned} \quad (13.6)$$

По построению,  $A_j = R'_j \oplus \bigoplus_{\gamma} X^{\gamma^0} Y^{\gamma^1}$  для некоторого множества векторов  $\gamma$ , удовлетворяющих условиям  $|\gamma| \leq |\alpha_j|$  и  $|\gamma^0| \leq |\alpha_j^0| - 2$ . Как следствие,  $\gamma \notin \bigcup_{i=1}^j Q(\alpha_j)$ . Поэтому  $P_j$  является суммой  $2j$  мультиаффинных функций, а  $R_j$  имеет вид (13.5).

После шага  $|T|$  получаем  $f^* = P_{|T|}$ . Присоединяя к  $\bigoplus \Lambda \oplus$ -представлению функции  $f^*$  мономы функции  $f \oplus f^*$  (их не более  $2^t + 2^{n-t}$ ), получаем  $\bigoplus \Lambda \oplus$ -схему для функции  $f$  из  $\leq 2|T| + 2^t + 2^{n-t} \asymp 2^n/n^2$  элементов умножения на втором слое и  $\leq t2^{n-t}$  элементов сложения на первом слое в силу (13.6). ■

- Порядок операций в схемах ограниченной глубины имеет значение. В частности, сложность  $\Lambda \oplus \Lambda$ -схем функций  $n$  переменных нельзя оценить лучше, чем  $2^n$  (пример: дизъюнкция  $n$  переменных).

Начиная с глубины 4, сложность схем в базисе  $\{\Lambda, \oplus, 1\}$  имеет порядок  $2^{n/2}$  (как легко проверить). В модели  $AC$ -схем, благодаря двойственности операций  $\Lambda$  и  $\vee$ , эффект исключительности глубины 3 не возникает: асимптотика сложности  $2 \cdot 2^{n/2}$  класса функций  $n$  переменных достигается сразу на схемах глубины 3, как показано автором в [91].

### $AC[\oplus]$ -схемы глубины 4 для функции голосования $[P][\varepsilon]$

Известно, что минимальные  $AC$ -схемы глубины  $d$  для функции голосования  $n$  переменных имеют сложность  $2^{n^{1/(d-1)\pm o(1)}}$ : нижняя оценка доказана Й. Хостадом [214], а верхняя достигается на простых монотонных схемах, построенных Р. Боппаной [149]. Недавно И. Оливейра, Р. Сантанам и С. Сринивасан [273]

установили, что модель  $AC[\oplus]$ -схем, хотя и не позволяет улучшить результат в глубине 3, уже на схемах глубины 4 (в какой-то степени неожиданно) демонстрирует более высокую вычислительную мощь. Перед тем как перейти к изложению метода [273], комбинирующего два вероятностных аргумента с идеей приближенных вычислений, напомним для сравнения конструкцию схем глубины 3.

**Теорема 13.4 ([149]).**  $C_{\vee\wedge\vee}(\text{maj}_n) \leq 2^{\sqrt{n \log n}}$ .

► Разделим множество переменных на  $r$  групп  $X_1, \dots, X_r$  одинакового размера  $|X_i| = m = \lceil n/r \rceil$ . Можно записать

$$\text{maj}_n(X) = \bigvee_{k_1+\dots+k_r=n/2} T_m^{k_1}(X_1) \cdot \dots \cdot T_m^{k_r}(X_r). \quad (13.7)$$

Схема строится по формуле (13.7), в которой пороговые функции  $T_m^k$  выражаются при помощи КНФ. На первом слое  $\vee\wedge\vee$ -схемы вычисляются всевозможные дизъюнкции переменных в каждой из групп  $X_i$ : всего  $\leq r2^m$  штук. На втором слое вычисляются внутренние произведения в (13.7) — их не более  $C_{n/2+r-1}^{r-1} < n^r$  штук. Требуемая оценка получается при  $r \approx \sqrt{n/\log n}$  и  $m \approx \sqrt{n \log n}$ . ■

**Теорема 13.5 ([273]).**  $C_4^{AC[\oplus]}(\text{maj}_n) \leq 2^{n^{1/4+o(1)}}$ .

► Метод синтеза комбинирует знакомые по формульной реализации функции голосования идеи: вычисление арифметических сумм переменных (для этого используются операции  $\oplus$ ) и построение монотонных приближений, только в отличие от метода Вэльянта [325] повышение точности приближения достигается не последовательными, а параллельными шагами.

Ввиду

$$\text{maj}_n = \bigvee_{k \geq n/2} E_n^k, \quad E_n^k = T_n^k \cdot \overline{T_n^{k+1}}, \quad (13.8)$$

вычисление функции  $\text{maj}_n$  сводится к вычислению элементарных симметрических функций  $E_n^k$  схемами глубины 4 с внешним элементом дизъюнкции. Достаточно построить схему для<sup>13)</sup>  $E_n^{n/2}$ .



Срикант Сринивасан  
Индийский технологический  
институт, Бомбей,  
с 2012 по 2020

I. Сначала построим монотонную  $\wedge\vee\wedge$ -схему, вычисляющую приближенно функцию  $\text{maj}_m$ . Для удобства полагаем  $m$  четным. Следующий результат фактически получен К. Амано в [125].

**Лемма 13.3 ([273]).** Пусть  $0 < \delta \leq 1/(4 \ln m)$  и  $m$  достаточно велико. При некотором вероятностном распределении  $\Delta$  на множестве  $\wedge\vee\wedge$ -формул  $m$  переменных сложности  $2\sqrt{\log m/\delta}$ , формула  $G(X) \in \Delta$  обладает свойствами:

- (i) Для любого вектора  $\sigma \in \mathbb{B}^m$  веса  $\leq (1/2 - \delta)m$  выполнено  $G(\sigma) = 0$ ;
- (ii) Для любого вектора  $\sigma \in \mathbb{B}^m$  веса  $m/2$  выполнено  $\mathbf{P}(G(\sigma) = 1) \geq 1/2$ .

<sup>13)</sup>Любая функция  $E_n^k$  является подфункцией функции  $E_{2n}^n$ .

▷ Определим последовательность распределений  $\Delta_k$  формул глубины  $k$ , параметризованную числами  $l_k \in \mathbb{N}$ . На множестве переменных зададим равномерное распределение  $\Delta_0$ : для  $G \in \Delta_0$  положим  $\mathbf{P}(G \equiv x_i) = 1/m$ . Распределение  $\Delta_k$  содержит формулы  $G_1 \circ \dots \circ G_{l_k}$ , где  $G_i$  случайно выбраны из  $\Delta_{k-1}$ , а  $\circ = \wedge$  при нечетных  $k$  и  $\circ = \vee$  — при четных.

Пусть  $|\sigma| = m/2$ . Тогда в силу  $1 - x \leq e^{-x}$  при  $x \in \mathbb{R}$ ,

$$\begin{aligned} p_1 &= \mathbf{P}(G(\sigma) = 1 \mid G \in \Delta_1) = 2^{-l_1}, \\ p_2 &= \mathbf{P}(G(\sigma) = 0 \mid G \in \Delta_2) = (1 - p_1)^{l_2} \leq e^{-l_2 p_1}, \\ p_3 &= \mathbf{P}(G(\sigma) = 0 \mid G \in \Delta_3) \leq l_3 p_2. \end{aligned}$$

Пусть теперь  $|\sigma| \leq (1/2 - \delta)m$ . В предположении  $\delta l_1 \leq 1/2$  выполнено

$$\begin{aligned} q_1 &= \mathbf{P}(G(\sigma) = 1 \mid G \in \Delta_1) \leq ((1 - 2\delta)/2)^{l_1} \leq (1 - \delta l_1)p_1, \\ q_2 &= \mathbf{P}(G(\sigma) = 0 \mid G \in \Delta_2) = (1 - q_1)^{l_2} \geq e^{-l_2(q_1 + q_1^2)}, \\ q_3 &= \mathbf{P}(G(\sigma) = 1 \mid G \in \Delta_3) = (1 - q_2)^{l_3} \leq e^{-q_2 l_3}. \end{aligned}$$

При оценке  $q_1$  мы воспользовались справедливым при  $ax \leq 1$  неравенством  $(1 - x)^a \leq 1 - ax/2$ , а при оценке  $q_2$  — справедливым при  $0 \leq x \leq 1/2$  неравенством  $1 - x \geq e^{-x - x^2}$ .

Положим<sup>14)</sup>  $l_1 = \sqrt{\ln m/\delta}$ ,  $l_2 = cl_1 2^{l_1}$ ,  $l_3 = e^{cl_1 - 2}$ , где  $c \in \mathbb{R}$ . При этом выполнено  $\delta l_1 = \sqrt{\delta \ln m} \leq 1/2$ .

Тогда  $p_2 \leq e^{-cl_1}$  и  $p_3 \leq e^{-2} < 1/4$ . Далее,

$$\begin{aligned} q_2 &\geq e^{-l_2(q_1 + q_1^2)} \geq e^{-l_2 p_1(1 - \delta l_1 + (1 - \delta l_1)^2 p_1)} \geq e^{-cl_1(1 + p_1/4) + c\delta l_1^2} \geq m^c e^{-cl_1(1 + p_1/4)}, \\ q_3 &\leq e^{-q_2 l_3} \leq e^{-m^c e^{-cl_1 p_1/4 - 2}} < e^{-m^c e^{-c/4 - 2}}, \end{aligned}$$

поскольку  $x 2^{-x} < 1$  при  $x \geq 1$ . Если выбрать  $c = 2$ , то  $q_3 \leq e^{-m^2/13}$ .

Теперь вероятность того, что для какого-то из векторов  $\sigma$  веса  $\leq (1/2 - \delta)m$  выполняется  $G(\sigma) = 1$ , можно оценить как  $q_4 < 2^m q_3 \leq 2^m e^{-m^2/13}$ . Эта величина не превосходит  $1/4$  при  $m \geq 11$ .

Определим распределение  $\Delta$  как ограничение  $\Delta_3$  на множество формул, для которых выполнено условие (i). Тогда для любого вектора  $\sigma \in \mathbb{B}^m$  веса  $m/2$  справедливо  $\mathbf{P}(G(\sigma) = 1 \mid G \in \Delta) \geq 1 - p_3 - q_4 \geq 1/2$ .  $\square$

II. Теперь функция  $E_m^{m/2}(X)$  может быть вычислена приближенно как  $\tilde{E} = \Psi_m(X) \cdot \Psi'_m(\bar{X}) \cdot \text{MOD}_m^{s,r}(X)$ , где  $r = m/2 \bmod s$ ,  $\Psi_m, \Psi'_m$  — случайные функции, реализуемые схемами из леммы 13.3, а  $\bar{X}$  — вектор отрицаний переменных. При условии  $s \geq \delta m$  приближающая (случайная) функция  $\tilde{E}$  равна 0 за пределами среднего слоя булева куба  $\mathbb{B}^m$ , а в любой точке среднего слоя равна 1 с вероятностью  $\geq 1/4$ . В случае  $s = 2^k$  функция  $\text{MOD}_m^{s,r}$  имеет простое представление в виде многочлена Жегалкина.

**Лемма 13.4.** *При  $n \geq 2^k$  и любом  $r$  справедливо  $C_{\oplus \wedge}(\text{MOD}_n^{2^k, r}) \leq n^{2^k}$ .*

---

<sup>14</sup>Округлениями в последующих (довольно грубых) расчетах пренебрегаем.

▷ Обозначим  $X^S = \prod_{i \in S} x_i$ . Заметим, что число  $C_n^{2^k-1}$  нечетно только при  $n \equiv -1 \pmod{2^k}$ . Как следствие,  $\text{MOD}_n^{2^k,-1}(X) = \bigoplus_{|S|=2^k-1} X^S$ . Тогда произвольную функцию  $\text{MOD}_n^{2^k,r}(X)$  можно вычислить как  $\text{MOD}_{n+t}^{2^k,-1}(X, 1^t)$ , где  $1^t$  — набор из  $t = 2^k - r - 1$  единиц. Выражающий функцию многочлен степени  $2^k - 1$  заведомо содержит не более  $C_n^{2^k-1} + C_n^{2^k-2} + \dots + C_n^0 < n^{2^k}$  мономов. □

(Если на этом остановиться и реализовать функцию  $E_{2m}^m$  как дизъюнкцию подходящего числа независимых случайных функций  $\tilde{E}$ , то при надлежащем выборе параметров  $\delta$  и  $s$  получится схема сложности  $2^{m^{1/3+o(1)}}$  — но такой результат можно получить гораздо проще обобщением метода теоремы 13.4 даже без использования операций  $\oplus$  [149].)

*III.* Пожалуй, ключевая идея метода [273] основана на наблюдении: если множество, состоящее из равного числа элементов двух типов, случайно разбить на подмножества четного размера, то каждое из подмножеств также будет содержать поровну элементов обоих типов с достаточно высокой вероятностью.

Пусть  $\tilde{X}_1, \dots, \tilde{X}_r$  — случайное разбиение множества из  $n$  переменных на группы по  $m$  переменных<sup>15</sup>). Рассмотрим случайную функцию  $\Phi = \prod_{i=1}^r \tilde{E}_i(\tilde{X}_i)$ , в которой разбиение и все внутренние функции  $\tilde{E}_i$  выбираются независимо; функции  $\tilde{E}_i$  имеют вид  $\tilde{E}$ . По построению,  $\Phi \leq E_n^{n/2}$ . При этом

$$p = \mathbf{P}(\Phi(\sigma) = 1 \mid |\sigma| = n/2) \geq \frac{1}{2^{2r}} \cdot \frac{(C_m^{m/2})^r}{C_n^{m/2}} \geq \frac{1}{4^r} \cdot \frac{(2^m/m)^r}{2^n} = \frac{1}{(4m)^r}.$$

Тогда дизъюнкция  $\bigvee_{i=1}^{n/p} \Phi_i$  независимых случайных функций типа  $\Phi$  принимает значение 0 хотя бы на одном наборе веса  $n/2$  с вероятностью  $\leq C_n^{n/2} (1-p)^{n/p} < 2^n e^{-n} < 1$ . Следовательно, некоторая формула глубины 4 вида  $\bigvee_{i=1}^{n/p} \Phi_i$  вычисляет функцию  $E_n^{n/2}$ .

При выборе параметров  $r, s \asymp \sqrt[4]{n/\log n}$ ,  $\delta = s/m$  сложность построенной формулы оценивается как  $(n/p)r \left( m^s + 2\sqrt{\log m/\delta} \right) \asymp 2\sqrt[4]{n \log^3 n}$ . ■

- Для  $AC[\oplus]$ -схем произвольной глубины  $d$  авторы [273] получили оценки

$$2^{n^{1/(2d-4)} - o(1)} \leq C_d^{AC[\oplus]}(\text{maj}_n) \leq 2^{n^{2/(3d-12)} + o(1)}.$$

Как отмечено выше, при  $d = 3, 4$  нижняя оценка точна; из доказательства [273] следует чуть лучшая верхняя оценка, которая усиливает стандартную оценку сложности  $AC$ -схем [149] при остальных  $d \geq 5$ .

### Метод сечения. Линейные схемы для матрицы Сильвестра //

Метод сечения используется в параллельных моделях вычисления, к которым относятся и схемы ограниченной глубины. Идея состоит в том, чтобы в исходной, вообще говоря, не параллельной схеме, провести разрез (один или несколько),

<sup>15</sup>Для простоты полагаем, что  $n$  делится на  $2r$ .

руководствуясь подходящим правилом, и далее выполнить параллельное перестроение частей схемы до и после разреза независимо.

Далее мы рассмотрим два примера применения этого метода, простой и посложнее.

Булева версия матрицы Сильвестра определяется как

$$H_1 = [0], \quad H_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & \overline{H}_n \end{bmatrix}. \quad (13.9)$$

Эта матрица является двоичным аналогом матрицы ДПФ и обладает многими замечательными свойствами (см., например, [233]).

Рекурсивное определение матрицы Сильвестра согласовано с модифицированным определением операции кронекерова произведения булевых матриц  $A \overline{\otimes} B$ , в котором на позиции нулей матрицы  $A$  подставляются матрицы  $B$ , а на позиции единиц — матрицы  $\overline{B}$ . Тогда справедливо

$$H_{n_1 n_2} = H_{n_1} \overline{\otimes} H_{n_2}, \quad \overline{H}_{n_1 n_2} = \overline{H}_{n_1} \overline{\otimes} H_{n_2}. \quad (13.10)$$

Стандартная схема для матрицы  $H_n$  имеет сложность  $\asymp n \log n$  и строится прямо по определению (13.9). Схема состоит из  $\log_2 n$  слоев: на очередном слое пары матриц  $H_{2m}$ ,  $\overline{H}_{2m}$  для групп из  $2m$  переменных собираются из матриц размера  $m \times m$  для групп из  $m$  переменных.

**Теорема 13.6** ([233]). Для  $d \geq 2$  и  $n = 2^k$  выполнено  $W_d(H_n) \leq dn^{1+1/d}$ .

► На самом деле, мы будем строить схему для пары матриц  $H_n$ ,  $\overline{H}_n$ . Для этого возьмем стандартную схему и разобьем ее слои на  $d$  групп примерно равного размера. Затем слои внутри каждой группы склеим, т.е. заменим их схемой глубины 1. При  $d = 1$  имеем  $W_1(H_n, \overline{H}_n) = |H_n| + |\overline{H}_n| \leq n^2$ .

Ввиду (13.10), имеет место

$$W_{d+1}(H_{n_1 n_2}, \overline{H}_{n_1 n_2}) \leq 2n_1^2 n_2 + n_1 W_d(H_{n_2}, \overline{H}_{n_2}).$$

Отсюда для  $n = n_1 \cdot \dots \cdot n_d$ , где  $n_i \in \{2^{\lfloor k/d \rfloor}, 2^{\lceil k/d \rceil}\}$ , получаем

$$W_d(H_n, \overline{H}_n) \leq 2n(n_1 + \dots + n_d) = 2n[(1+x)2^{-x}]d2^{k/d}, \quad x = \frac{k}{d} - \left\lfloor \frac{k}{d} \right\rfloor. \quad (13.11)$$

Функция в квадратных скобках на отрезке  $[0, 1]$  принимает максимальное значение  $2/(e \ln 2) \approx 1.06$  при  $x = \log_2(e/2)$ . ■

- Результат теоремы точен по порядку, т.к. нижняя оценка, доказанная в [233], имеет вид  $W_d(H_n) \geq W_d^\vee(H_n) \gtrsim dn(n/2)^{1/d}$ . Если заметить, что на последнем слое не нужно вычислять дополнительную матрицу, оценку теоремы можно получить в более аккуратной форме  $W_d(H_n) \leq 2(d-1)n^{1+1/d}$ , используя вместо (13.11) соотношение  $W_d(H_n) \leq n(n_1 + 2n_2 + \dots + 2n_{d-1} + n_d)$ .

## Перестроение арифметических схем в $\Sigma\Pi\Sigma\Pi$ -схемы //

В работе [117] М. Агравал и В. Винай получили в какой-то степени неожиданный и резонансный результат: оказывается, арифметические схемы над кольцами достаточно общего вида могут быть промоделированы арифметическими схемами глубины 4 ненамного большей сложности. Оценка сложности впоследствии уточнялась и приняла окончательный вид по результатам работы С. Тавенаса [321].

Идея метода очень проста: схема глубины 4 строится сечением исходной схемы приблизительно пополам, и каждая половина приводится к глубине 2. В результате многочлен записывается как композиция многочленов. Содержательно задача состоит только в том, чтобы подобрать подходящее сечение и аккуратно оценить сложность новой схемы.

Следующие утверждения сформулированы для колец характеристики 0: в действительности, требуется, чтобы базовое кольцо  $R$  удовлетворяло условию  $\deg(fg) = \deg f + \deg g$  для любых многочленов  $f, g \in R[x]$ . Арифметическую схему назовем *однородной*, если на входах и выходах элементов умножения в ней вычисляются однородные многочлены<sup>16)</sup>.

**Лемма 13.5** ([317]). *Пусть  $R$  — кольцо характеристики 0. Если многочлен  $f \in R[X]$  степени  $d$  вычисляется арифметической схемой с  $s$  нескалярными элементами умножения, то он также может быть вычислен однородной арифметической схемой с  $sd^2$  нескалярными элементами умножения.*

▷ Любой многочлен, получающийся в процессе вычислений, разобьем на однородные компоненты степени  $\leq d$  (компоненты более высокой степени не нужны). Умножения исходной схемы заменим умножениями однородных компонент степени от 1 до  $d-1$ . Так получим схему, вычисляющую все однородные компоненты многочлена  $f$ . □

Следующая лемма описывает переход от обычной (однородной) арифметической схемы к  $\Sigma\Pi\Sigma\Pi$ -схеме. Доказательство проводится путем сечения исходной схемы пополам: линия разреза проходит через элементы, в которых степени промежуточных многочленов преодолевают заданный порог  $h$ . Схема рассуждения близка к [326]. Через  $\text{mon } f$  обозначаем множество мономов многочлена  $f$ .

**Лемма 13.6.** *Пусть многочлен  $f \in R[x_1, \dots, x_n]$  степени  $d$  вычисляется однородной арифметической схемой  $S$  с  $s$  элементами умножения, где  $R$  — кольцо характеристики 0. Тогда при любом  $h \leq d$  его можно представить в виде*

$$f(x_1, \dots, x_n) = p(q_1, \dots, q_r), \quad p \in R[y_1, \dots, y_r], \quad q_j \in R[\text{mon } f], \quad (13.12)$$

<sup>16)</sup>Определение допускает вычисление однородной схемой произвольного многочлена как суммы однородных компонент.



Маниндра Агравал  
Индийский технологический институт, Канпур, с 1996

$$\text{где } \deg q_j \leq h, \deg p < 6d/h, |\text{mon } p| \leq s^{\deg p} u r \leq s^2 + n.$$

▷ Классифицируем ребра схемы  $S$ , входящие в элементы умножения. Входящее ребро, доставляющее в элемент умножения множитель более высокой степени, назовем *сильным*, другое ребро — *слабым*. В случае равенства степеней перемножаемых многочленов определим сильное и слабое ребро в паре произвольно. Ориентированный путь, соединяющий две вершины в схеме, назовем *легальным*, если он не содержит слабых ребер.

Обозначим через  $g(v)$  многочлен, вычисляемый в вершине  $v \in S$ . Для элемента умножения  $v$  положим  $g(v) = g_1(v) \cdot g_2(v)$ , где  $g_1(v)$  — множитель, приходящий по сильному ребру,  $g_2(v)$  — по слабому. Для элемента сложения  $v$  полагаем  $g_1(v) = g(v)$  и  $g_2(v) = 1$ . Для легального пути  $\rho = (v_1, \dots, v_l)$  определим  $g(\rho) = g_2(v_2) \cdot \dots \cdot g_2(v_l)$ . В случае  $l = 1$  формально положим  $g(\rho) = 1$ . Наконец, определим

$$g(v, w) = \sum_{\rho=(v, \dots, w)} g(\rho),$$

где суммирование выполняется по всем легальным путям из  $v$  в  $w$ ; в случае отсутствия таких путей полагается  $g(v, w) = 0$ .

Пусть  $V_t = \{v \in S \mid \deg g(v) \geq t > \deg g_1(v)\}$  — множество элементов умножения, в которых степень вычисляемых многочленов впервые преодолевает порог  $t$ . Очевидно, множество  $V_t$  является антицепью.

**Утверждение 13.1.** (i) Пусть  $\deg g(w) \geq t$ . Тогда  $g(w) = \sum_{v \in V_t} g(v) \cdot g(v, w)$ .  
(ii) Пусть  $\deg g(u, w) \geq t - \deg g(u) > 0$ . Тогда  $g(u, w) = \sum_{v \in V_t} g(u, v) \cdot g(v, w)$ .

▷ Доказательство п. (i) проводится индукцией по расстоянию, которое берется по легальным ориентированным путям, до  $w$  от ближайшей вершины из множества  $V_t$ . Если  $w \in V_t$ , то в схеме нет других вершин из  $V_t$ , предшествующих  $w$ , поэтому проверяемое равенство является тождеством  $g(w) = g(w) \cdot g(w, w)$ .

Иначе, для предшествующей  $w$  по сильному ребру вершины  $w_1$  выполнено  $\deg g(w_1) \geq t$ , при этом вершина  $w_1$  находится ближе к  $V_t$ , следовательно, по предположению индукции верно  $g(w_1) = \sum_{v \in V_t} g(v) \cdot g(v, w_1)$ . Учитывая, что  $g(v, w) = g(v, w_1) \cdot g(w_2)$  ввиду того, что все легальные пути в  $w$  проходят через  $w_1$ , окончательно получаем

$$g(w) = g(w_1) \cdot g(w_2) = \sum_{v \in V_t} g(v) \cdot g(v, w_1) \cdot g(w_2) = \sum_{v \in V_t} g(v) \cdot g(v, w).$$

Доказательство п. (ii) полностью аналогично. □

Для выражения многочленов  $g(v)$  и  $g(u, w)$  воспользуемся предоставляемыми утверждением 13.1 формулами:

$$g(w) = \sum_{v \in V_t} g_1(v) \cdot g_2(v) \cdot g(v, w), \quad (13.13)$$

$$g(u, w) = \sum_{v \in V_t} g(u, v') \cdot g_2(v) \cdot g(v, w), \quad (13.14)$$

где вершина  $v'$  предшествует  $v$  по сильному ребру. В формуле (13.13) выбираем  $t = (\deg g(w))/2$ , тогда степени всех множителей в ней не превосходят  $t$ . В формуле (13.14) выбираем  $t = \deg g(u) + (\deg g(u, w))/2$  (поэтому  $u \notin V_t$ ), тогда  $\deg g(u, v'), \deg g(v, w) \leq (\deg g(u, w))/2$ .

Теперь опишем преобразование схемы к виду (13.12). Последовательно, в порядке убывания степеней, выразим все многочлены  $g(v)$  и  $g(u, w)$  по правилам (13.13), (13.14) вплоть до многочленов степени  $\leq h$ , которые будем считать формальными переменными  $y_1, \dots, y_r$ . По построению,  $r \leq s + C_s^2 + n \leq s^2 + n$ . Будем обозначать через  $\deg^*$  степень многочлена  $g(v)$  или  $g(u, w)$  как многочлена переменных  $y_i$ .

### Утверждение 13.2.

- (i) Если  $d' = \deg g(v) > h$ , то  $\deg^* g(v) \leq 6\lfloor d'/h \rfloor - 3$ .
- (ii) Если  $d' = \deg g(u, w) > h$ , то  $\deg^* g(u, w) \leq 6\lfloor d'/h \rfloor - 1$ .

▷ Неравенства (i), (ii) доказываются совместно индукцией по  $d'$ , исходя из (13.13), (13.14). База индукции  $d' \leq 4h$  проверяется непосредственно.

Ввиду простого неравенства  $\lfloor a+b \rfloor \geq \lfloor a \rfloor + \lfloor b \rfloor$ , требуется разобраться только в ситуациях с множителями степени  $\leq h$  в формулах (13.13), (13.14).

Заметим, что при  $d' > 4h$  не более одного множителя в формуле (13.13) и не более двух множителей (если двух, то крайних) в формуле (13.14) имеют степени  $\leq h$ . Отсюда сразу следует индуктивный переход.  $\square$

**Утверждение 13.3.** Для  $g(y_1, \dots, y_r) \in \{g(v), g(u, v)\}$  выполнено  $|\operatorname{mon} g| \leq s^{\deg^* g - 1}$ .

▷ Тривиальное доказательство по индукции. При  $\deg^* g = 1$  утверждение очевидно. Индуктивный переход обеспечивается правилами (13.13), (13.14), поскольку в силу выбора порога  $t$  не менее двух множителей в каждом внутреннем произведении отличны от 1.  $\square$

Утверждения 13.2 и 13.3 дают требуемые оценки на  $\deg p$  и  $|\operatorname{mon} p|$ . Лемма доказана.  $\square$

- Число мономов в утверждении 13.3 можно оценить точнее как  $s^{3\lfloor d'/h \rfloor - 1}$ , где  $d' = \deg g(x_1, \dots, x_n) \geq h$  (доказывается так же, как утверждение 13.2). Тогда вывод леммы можно уточнить до  $|\operatorname{mon} p| \leq s^{3d/h}$ .

**Теорема 13.7** ([321]). Пусть  $R$  — кольцо характеристики 0. Если многочлен  $f \in R[x_1, \dots, x_n]$  степени  $d$  имеет мультипликативную сложность  $s$ , то  $C_{\Sigma\Pi\Sigma\Pi}^R(f) = 2^{O(\sqrt{d \log(sd) \log n})}$ .

- В доказательстве мы используем простое неравенство  $C_{n+k}^k \leq (k+1)n^k$ .

К преобразованной при помощи леммы 13.5 схеме применим лемму 13.6 с выбором  $h \approx \sqrt{\frac{d \log(sd)}{\log(3n)}}$ . Заметим, что при этом обеспечено  $h \leq d$ , поскольку  $sd \leq dC_{n+d}^d \leq d(d+1)n^d \leq (3n)^d$ .

Построенная по формуле (13.12) схема имеет не более  $C_{n+h}^h \leq (h+1)n^h = 2^{O(\sqrt{d \log(sd) \log n})}$  элементов на первом слое (всевозможные мономы степени  $\leq h$  переменных  $x_i$ ), не более  $s^2 d^4 + n = 2^{O(\log(sd))}$  элементов на втором слое и не более  $(sd^2)^{6d/h} = 2^{O(\sqrt{d \log(sd) \log n})}$  элементов на третьем слое. ■

Отметим, что переход к схемам глубины 4 выполняется перестроением топологии схемы, поэтому результат теоремы 13.7 справедлив для вычислений в кольцах  $R$  достаточно общего вида, даже, например, для схем над тропическим полукольцом  $(\mathbb{R}, \min, +)$ .

Теорема 13.7 показывает, что если сложность  $s$  многочлена не очень велика, то не слишком велика и его сложность в классе схем глубины 4. Скажем, если  $s = 2^{O(d \log n)}$ , то сложность в глубине 4 также будет величиной  $2^{o(d \log n)}$ . Разнообразные следствия из этого результата приведены в [117, 207, 321]. Одно из наиболее интересных — рекордно простые схемы для определителя  $n \times n$  матрицы. Напомним, что обычная сложность вычисления определителя полиномиальна, она близка к сложности матричного умножения<sup>17)</sup>.

**Следствие 13.2** ([321]). *Определитель  $n \times n$  матрицы над кольцом характеристики 0 может быть вычислен  $\Sigma\Pi\Sigma$ -схемой сложности  $2^{O(\sqrt{n \cdot \log n})}$ .*

Построить такие схемы другим способом не удавалось. До появления работ [117, 207] для определителя даже не были известны схемы ограниченной глубины и сложности  $2^{o(n)}$ .

## Перестроение арифметических схем в $\Sigma\Pi\Sigma$ -схемы



Нирадж Кайял

Исследовательская  
лаборатория Майкрософт,  
Бенгалуру, с 2008

Результаты предыдущего параграфа получили продолжение, уже действительно неожиданное. Группе индийских математиков [207] удалось построить переход от схем глубины 4 к схемам глубины 3 практически без изменения оценки сложности, если вычисления выполняются над полями  $\mathbb{Q}, \mathbb{R}, \mathbb{C}$ . Перестроение в схему глубины 3 осуществляется при помощи двух алгебраических (и отчасти теоретико-числовых) приемов: на промежуточном шаге получается схема глубины 5, в которой все умножения являются возведениями в степень.

Используемые алгебраические инструменты сужают множество допустимых колец, поэтому последующее изложение ограничено случаем комплексных многочленов.

**Лемма 13.7.** *В поле характеристики 0 выполнено*

$$2^{n-1} n! x_1 \cdot \dots \cdot x_n = \sum_{\varepsilon_2, \dots, \varepsilon_n = \pm 1} \varepsilon_2 \cdot \dots \cdot \varepsilon_n (x_1 + \varepsilon_2 x_2 + \dots + \varepsilon_n x_n)^n.$$

<sup>17)</sup> Пожалуй, легче всего полиномиальность сложности определителя проверить при помощи формулы Н. Пиппенджера [286], приводящей к оценке  $O(n^4 \log n)$ .

▷ Доказательство проводится «раскрытием скобок» в правой части и приведением подобных слагаемых. Несложно проверить, что моном  $x_1 \cdot \dots \cdot x_n$  входит в любое слагаемое под знаком суммы с коэффициентом  $n!$ . В любом другом мономе степени  $n$  некоторая переменная  $x_i$  присутствует в четной степени (возможно, в нулевой). Тогда этот моном входит в любую пару слагаемых суммы, отличающихся только значением  $\varepsilon_i$ , с противоположными коэффициентами. Как следствие, его коэффициент в общей сумме равен 0.  $\square$

- Формула леммы известна достаточно давно. Согласно [14] различные ее варианты предлагались на школьных и студенческих олимпиадах в СССР в 1980-е гг. Формула опубликована И. Фишером в [188], а С. Б. Гашков и Е. Т. Шавгуидзе [14] доказали ее оптимальность: меньшим числом степеней линейных форм в правой части, чем  $2^{n-1}$ , обойтись нельзя. Р. Саптариши [297] заметил, что альтернативное представление в виде суммы  $2^n$  степеней линейных форм также дает известная формула Г. Райзера [296] для алгебраического перманента:

$$n! x_1 \cdot \dots \cdot x_n = \text{per} \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & \cdots & x_n \end{pmatrix} = \sum_{T \subset \llbracket n \rrbracket} (-1)^{n-|T|} \left( \sum_{i \in T} x_i \right)^n.$$

Следующая лемма позволяет от  $\Sigma\Pi\Sigma\Pi$ -представления многочлена  $f$  (13.12) перейти к  $\Sigma E \Sigma E \Sigma$ -представлению, где  $E$  означает слой элементов возведения в степень.

**Лемма 13.8** ([207, 321]). *В условиях леммы 13.6 многочлен  $f \in \mathbb{C}[x_1, \dots, x_n]$  степени  $d$  при любом  $h \leq d$  можно представить в виде*

$$f(X) = \sum_{i=1}^{s_1} \left( \sum_{j=1}^{s_2} (l_{i,j}(X))^{h_{i,j}} \right)^{d_i}, \quad d_i \leq 6d/h, \quad h_{i,j} \leq h, \quad \deg l_{i,j} \leq 1, \quad (13.15)$$

где  $s_1 \leq (2s)^{6d/h}$  и  $s_2 \leq h \cdot (2n)^h$ .

▷ В формуле (13.12) выразим каждый моном многочлена  $p$  при помощи леммы 13.7 как  $\sum_{i=1}^{2^{k-1}} \lambda_i^k(y_1, \dots, y_r)$ , где  $k$  — степень монома,  $\deg \lambda_i = 1$ .

В каждый многочлен  $\lambda_i$  подставим вместо переменных  $Y$  соответствующие им многочлены переменных  $X$  и затем при помощи леммы 13.7 каждый моном переменных  $X$  перепишем в виде  $\sum_{i=1}^{2^{k-1}} l_{i,j}^k(x_1, \dots, x_n)$ , где  $k$  — степень монома.

По построению,  $s_1 \leq 2^{\deg p} \cdot |\text{mon } p| \leq (2s)^{\deg p}$  и  $s_2 \leq 2^{h-1} C_{h+n}^h \leq h(2n)^h$ .  $\square$

Ключом к синтезу схем глубины 3 служит элегантный результат Н. Саксены [298], позволяющий выразить степень линейной формы суммой сравнительно малого числа произведений многочленов одной переменной.

**Лемма 13.9** ([298]). *Для любых  $d, n > 0$  и различных чисел  $a_1, \dots, a_{dn+1} \in \mathbb{C}$  выполнено*

$$(x_1 + \dots + x_n)^d = \sum_{i=1}^{dn+1} b_i \prod_{j=1}^n E_d(a_i x_j), \quad E_d(x) = 1 + x + \frac{x^2}{2} + \dots + \frac{x^d}{d!}.$$

при некоторых  $b_i \in \mathbb{C}$ .

▷ Положим  $u = x_1 + \dots + x_n$ . В силу

$$e^{uz} = 1 + uz + \dots + (uz)^d/d! + \dots$$

$u^d$  является коэффициентом при  $z^d$  степенного ряда  $e^{uz} \in \mathbb{C}[[z]]$ . Ввиду

$$e^{uz} = e^{x_1 z} \cdot \dots \cdot e^{x_n z} \equiv F(z) = E_d(x_1 z) \cdot \dots \cdot E_d(x_n z) \pmod{z^{d+1}},$$

искомая величина  $u^d$  — это коэффициент многочлена  $F(z)$  при  $z^d$ .

Согласно интерполяционной формуле (Лагранжа) коэффициенты многочлена степени  $m$  могут быть выражены в виде линейных комбинаций значений многочлена в  $m+1$  точках, в данном случае,  $F(a_1), \dots, F(a_{dn+1})$ .  $\square$

**Теорема 13.8** ([207, 321]). *Если многочлен  $f \in \mathbb{C}[x_1, \dots, x_n]$  степени  $d$  имеет мультипликативную сложность  $s$ , то  $C_{\Sigma\Pi\Sigma}(f) = 2^{O(\sqrt{d \log(sd) \log n})}$ .*



Нитин Саксена  
Индийский технологический институт, Канпур, с 2013

► При помощи леммы 13.9 любое слагаемое внешней суммы в (13.15) можно переписать в виде

$$\left( \sum_{j=1}^{s_2} (l_{i,j}(X))^{h_{i,j}} \right)^{d_i} = \sum_{k=1}^{s_2 d_i + 1} b_k \prod_{j=1}^{s_2} E_{d_i}(a_k (l_{i,j}(X))^{h_{i,j}}).$$

В поле  $\mathbb{C}$  многочлен  $E_m(ax^q)$  раскладывается в произведение линейных множителей  $\prod_{i=1}^{qm} \lambda_{a,m,q,i}(x)$ . Окончательно получаем

$$f(X) = \sum_{i=1}^{s_1} \sum_{k=1}^{s_2 d_i + 1} b_k \prod_{j=1}^{s_2} \prod_{u=1}^{d_i h_{i,j}} \lambda_{a_k, d_i, h_{i,j}, u}(l_{i,j}(X)).$$

Формула содержит (согласно оценкам леммы 13.8 и с учетом леммы 13.5) порядка

$$s_1 s_2^2 d^2 / h \leq (2sd^2)^{6d/h} (2n)^{2h} d^2 h$$

линейных множителей. При выборе  $h \approx \sqrt{\frac{d \log(sd)}{\log(3n)}}$  (напомним, что при этом  $h \leq d$ ) получаем требуемую оценку.  $\blacksquare$

Теперь следствие 13.2 может быть усилено.

**Следствие 13.3** ([321]). *Определитель комплексной  $n \times n$  матрицы может быть вычислен  $\Sigma\Pi\Sigma$ -схемой сложности  $2^{O(\sqrt{n} \cdot \log n)}$ .*

- Авторы [207] доказали, что результат теоремы 13.8 справедлив и над полем  $\mathbb{Q}$ . Из работы [256] следует, что оценка теоремы 13.8 близка к оптимальной: приводится пример многочлена  $f$  степени  $d$  от  $n$  переменных, для которого  $C_{\Sigma\Pi\Sigma}(f) = 2^{\Omega(\sqrt{d} \cdot \log n)}$ , правда, при ограничении  $d \prec \log n$ .

Аналог теоремы 13.8 для конечных полей не имеет места. Так, Д. Григорьев и А. Разборов [206] получили по сути экстремально возможную нижнюю оценку  $2^{\Omega(n)}$  сложности вычисления симметрических функций  $\text{MOD}_q^n$  над  $\mathbb{F}_p$  схемами глубины 3 при условии  $p, q \in \mathbb{P}$  и  $q \neq p$ , а также функции голосования  $\text{maj}_n$  над  $\mathbb{F}_2$ . Как известно [154], мультипликативная сложность любой нелинейной симметрической функции  $n$  переменных над  $\mathbb{F}_2$  равна  $\Theta(n)$ . Совсем несложно убедиться, что при  $p = O(1)$  выполнено  $C_{\mathcal{A}_{\mathbb{F}_p}}(\text{MOD}_2^n) = n^{O(1)}$ .

Теорему 13.8 нельзя распространить и на тропические схемы. В частности, в работе [261] показано, что тропический вариант многочлена  $\text{CONN}_n$  степени  $n$  от  $C_n^2$  переменных (соответствующий задаче поиска кратчайшего пути в графе) имеет сложность  $2^{\Theta(n \log n)}$  при реализации схемами глубины 3 над полукольцом  $(\mathbb{R}, \min, +)$ .

# Литература

- [1] Алексеев В. Б. *Сложность умножения матриц*. Кибернетический сборник. Вып. 25. М.: Мир, 1988, 189–236. [123](#)
- [2] Алексеев В. Е. *О некоторых алгоритмах сортировки с минимальной памятью*. Кибернетика. 1969. (5), 99–103. [32](#), [94](#)
- [3] Андреев А. Е. *О синтезе схем из функциональных элементов в полных монотонных базисах*. Математические вопросы кибернетики. Вып. 1. М.: Физматлит, 1988, 114–139. [39](#), [41](#)
- [4] Ахо А. В., Хопкрофт Дж. Э., Ульман Дж. Д. Структуры данных и алгоритмы. М.: Вильямс, 2000. [Ориг.: Aho A. V., Hopcroft J. E., Ullman J. D. *Data structures and algorithms*. Reading: Addison-Wesley, 1983.] [129](#)
- [5] Белага Э. Г. *О вычислении многочленов от одного переменного с предварительной обработкой коэффициентов*. Проблемы кибернетики. Вып. 5. М.: Физматлит, 1961, 7–16. [33](#)
- [6] Болотов А. А., Гашков С. Б. *О быстром умножении в нормальных базисах конечных полей*. Дискретная математика. 2001. **13**(3), 3–31. [81](#)
- [7] Гашков С. Б. *О глубине булевых функций*. Проблемы кибернетики. Вып. 34. М.: Наука, 1978, 265–268. [126](#)
- [8] Гашков С. Б. *О параллельном вычислении некоторых классов многочленов с расщущим числом переменных*. Вестник Московского университета. Серия 1. Математика. Механика. 1990. (2), 88–92. [126](#)
- [9] Гашков С. Б. *Замечание о минимизации глубины булевых схем*. Вестник Московского университета. Серия 1. Математика. Механика. 2007. (3), 7–9. [139](#)
- [10] Гашков С. Б., Гашков И. Б. *О сложности вычисления дифференциалов и градиентов*. Дискретная математика. 2005. **17**(3), 45–67. [100](#)
- [11] Гашков С. Б., Гринчук М. И., Сергеев И. С. *О построении схем сумматоров малой глубины*. Дискретный анализ и исследование операций. Серия 1. 2007. **14**(1), 27–44; 2008. **15**(4), 92–93. [30](#)
- [12] Гашков С. Б., Кочергин В. В. *Об аддитивных цепочках векторов, вентилярных схемах и сложности вычисления степеней*. Сб. трудов ИМ СО РАН. Методы дискретного анализа в теории графов и сложности. Вып. 52. Новосибирск: 1992, 22–40. [37](#)
- [13] Гашков С. Б., Сергеев И. С. *Умножение*. Чебышевский сборник. 2020. **21**(1), 101–134. [74](#)
- [14] Гашков С. Б., Шавгулидзе Е. Т. *О представлении произведений в виде суммы степеней линейных форм*. Вестник Московского университета. Серия 1. Математика. Механика. 2014. (2), 9–14. [158](#)

- [15] Гринчук М. И. Уточнение верхней оценки глубины сумматора и компаратора. Дискретный анализ и исследование операций. Серия 1. 2008. **15**(2), 12–22. **29, 96, 97, 99**
- [16] Диниц Е. А. Алгоритм решения задачи о максимальном потоке в сети со степенной оценкой. Доклады АН СССР. 1970. **194**(4), 754–757. **131**
- [17] Жданович Д. В. Экспонента сложности матричного умножения. Фундаментальная и прикладная математика. 2012. **17**(2), 107–166. **123**
- [18] Кабатянский Г. А., Панченко В. И. Упаковки и покрытия пространства Хэмминга шарами единичного радиуса. Проблемы передачи информации. 1988. **24**(4), 3–16. **125**
- [19] Карацуба А. А. Комментарии к моим работам, написанные мной самим. Современные проблемы математики. 2013. **17**, 7–29. **74**
- [20] Карацуба А. А., Офман Ю. П. Умножение многозначных чисел на автоматах. Доклады АН СССР. 1962. **145**(2), 293–294. **23**
- [21] Карзанов А. В. Точная оценка алгоритма нахождения максимального потока, примененного к задаче о «представителях». Вопросы кибернетики. Труды семинара по комбинаторной математике (Москва, 1971). М.: Сов. радио, 1973, 66–70. **131**
- [22] Ким К. В., Нестеров Ю. Е., Черкасский Б. В. Оценка трудоемкости вычисления градиента. Доклады АН СССР. 1984. **275**(6), 1306–1309. **103**
- [23] Клосс Б. М., Малышев В. А. Оценки сложности некоторых классов функций. Вестник Московского университета. Серия 1. Математика. Механика. 1965. (4), 44–51. **124**
- [24] Кнут Д. Искусство программирования. Т. 2. Получисленные алгоритмы. М.: Вильямс, 2004. [Ориг: Knuth D. E. *The art of computer programming. Vol. 2. Seminumerical algorithms*. Reading: Addison-Wesley, 1997.] **2, 24, 30**
- [25] Кнут Д. Искусство программирования. Т. 3. Сортировка и поиск. М.: Вильямс, 2007. [Ориг: Knuth D. E. *The art of computer programming. Vol. 3. Sorting and searching*. Reading: Addison-Wesley, 1998.] **32, 61**
- [26] Коробков В. К. Реализация симметрических функций в классе  $\pi$ -схем. Доклады АН СССР. 1956. **109**(2), 260–263. **105**
- [27] Коровин В. В. О сложности реализации универсальной функции схемами из функциональных элементов. Дискретная математика. 1995. **7**(2), 95–102. **42**
- [28] Коршунов А. Д. О числе монотонных булевых функций. Проблемы кибернетики. Вып. 38. М.: Наука, 1981, 5–108. **39**
- [29] Коршунов А. Д. Монотонные булевые функции. Успехи математических наук. 2003. **58**(5), 89–162. **39**
- [30] Кочергин В. В. Задачи Беллмана, Кнута, Лупанова, Пиппенджера и их вариации как обобщения задачи об аддитивных цепочках. Математические вопросы кибернетики. Вып. 20. М.: Физматлит, 2022, 119–256. **37**
- [31] Кочергин В. В., Кочергин Д. В. Уточнение нижней оценки сложности возведения в степень. Прикладная дискретная математика. 2017. (38), 119–132. **35**
- [32] Кричевский Р. Е. О сложности параллельно-последовательных контактных схем, реализующих однушажность булевых функций. Проблемы кибернетики. Вып. 12. М.: Наука, 1964, 45–55. **105**
- [33] Ложкин С. А. О связи между глубиной и сложностью эквивалентных формул и о глубине монотонных функций алгебры логики. Проблемы кибернетики. Вып. 38. М.: Наука, 1981, 269–271. **138**

- [34] Ложкин С. А. *О глубине функций алгебры логики в некоторых базисах*. Annales Univ. Budapest. Sec. Computatorica. 1983. IV, 113–125. **126**
- [35] Ложкин С. А. *Оценки высокой степени точности для сложности управляющих систем из некоторых классов*. Математические вопросы кибернетики. Вып. 6. М.: Физматлит, 1996, 189–214. **37, 126**
- [36] Ложкин С. А. *О минимальных  $\pi$ -схемах для монотонных симметрических функций с порогом 2*. Дискретная математика. 2005. **17**(4), 108–110. **105**
- [37] Ложкин С. А. *О синтезе формул, сложность которых не превосходит асимптотически наилучших оценок высокой степени точности*. Вестник Московского университета. Серия 1. Математика. Механика. 2007. (3), 19–25. **126**
- [38] Ложкин С. А. *Уточненные оценки функции Шеннона для сложности схем из функциональных элементов*. Вестник Московского университета. Серия 1. Математика. Механика. 2022. (3), 32–40. **37**
- [39] Ложкин С. А., Власов Н. В. *О сложности мультиплексорной функции в классе  $\pi$ -схем*. Ученые записки Казанского государственного университета. Серия: Физ.-мат. науки. 2009. **151**(2), 98–106. **42**
- [40] Ложкин С. А., Семенов А. А. *Об одном методе сжатия информации и о сложности реализации монотонных симметрических функций*. Известия вузов. Математика. 1988. (7), 44–52. **109**
- [41] Лупанов О. Б. *О вентильных и контактно-вентильных схемах*. Доклады АН СССР. 1956. **111**(6), 1171–1174. **35, 36, 143**
- [42] Лупанов О. Б. *Об одном методе синтеза схем*. Известия ВУЗ. Радиофизика. 1958. **1**(1), 120–140. **35, 36, 37**
- [43] Лупанов О. Б. *О синтезе контактных схем*. Доклады АН СССР. 1958. **119**(1), 23–26. **42, 125**
- [44] Лупанов О. Б. *О сложности реализации функций алгебры логики формулами*. Проблемы кибернетики. Вып. 3. М.: Физматгиз, 1960, 61–80. **125, 126**
- [45] Лупанов О. Б. *О реализации функций алгебры логики формулами из конечных классов (формулами ограниченной глубины) в базисе  $\&, \vee, \neg$* . Проблемы кибернетики. Вып. 6. М.: Физматлит, 1961, 5–14. **138**
- [46] Лупанов О. Б. *Об одном подходе к синтезу управляющих систем — принципе локального кодирования*. Проблемы кибернетики. Вып. 14. М.: Наука, 1965, 31–110. **14, 39, 41**
- [47] Лупанов О. Б. *К вопросу о реализации симметрических функций алгебры логики контактными схемами*. Проблемы кибернетики. Вып. 15. М.: Наука, 1965, 85–99. **43, 89**
- [48] Лупанов О. Б. *Асимптотические оценки сложности управляющих систем*. М.: Изд-во МГУ, 1984. **93, 125**
- [49] Лупанов О. Б. *О сложности моделирования степеней булевых  $(n, n)$ -функций*. Математические вопросы кибернетики. Вып. 12. М.: Физматлит, 2003, 179–216. **117**
- [50] Лупанов О. Б. А. Н. Колмогоров и теория сложности схем. Математические вопросы кибернетики. Вып. 17. М.: Физматлит, 2008, 5–12. **72**
- [51] Маргулис Г. А. *Явные конструкции расширителей*. Проблемы передачи информации. 1973. **9**(4), 71–80. **63**
- [52] Марков А. А. *Об инверсионной сложности систем функций*. Доклады АН СССР. 1957. **116**(6), 917–919. **93**

- [53] Митягин Б. С., Садовский Б. Н. *О линейных булевских операторах*. Доклады АН СССР. 1965. **165**(4), 773–776. **95**
- [54] Нечипорук Э. И. *О сложности схем в некоторых базисах, содержащих нетри-виальные элементы с нулевыми весами*. Проблемы кибернетики. Вып. 8. М.: Физматлит, 1962, 123–160. **136**, **137**
- [55] Нечипорук Э. И. *О вентильных схемах*. Доклады АН СССР. 1963. **148**(1), 50–53. **37**, **38**, **114**
- [56] Нечипорук Э. И. *О синтезе логических сетей в неполных и вырожденных базисах*. Проблемы кибернетики. Вып. 14. М.: Наука, 1965, 111–160. **137**
- [57] Нечипорук Э. И. *О топологических принципах самокорректирования*. Проблемы кибернетики. Вып. 21. М.: Наука, 1969, 5–102. **37**, **114**
- [58] Нечипорук Э. И. *Об одной булевской матрице*. Проблемы кибернетики. Вып. 21. М.: Наука, 1969, 237–240. **123**
- [59] Нигматуллин Р. Г. *Сложность булевых функций*. М.: Наука, 1991. **2**
- [60] Офман Ю. П. *Алгоритмическая сложность дискретных функций*. Доклады АН СССР. 1962. **145**(1), 48–51. **28**
- [61] Пан В. Я. *Некоторые схемы для вычисления значений полиномов с вещественными коэффициентами*. Доклады АН СССР. 1959. **127**(2), 266–269. **33**
- [62] Пан В. Я. *Некоторые схемы для вычисления значений полиномов с вещественными коэффициентами*. Проблемы кибернетики. Вып. 5. М.: Физматлит, 1961, 17–30. **33**
- [63] Пан В. Я. *О некоторых способах вычисления значений многочленов*. Проблемы кибернетики. Вып. 7. М.: Физматлит, 1962, 21–30. **33**
- [64] Пан В. Я. *О схемах вычисления произведений матриц и обратной матрицы*. Успехи математических наук. 1972. **27**(5), 249–250. **99**
- [65] Пан В. Я. *Быстрое умножение матриц и смежные вопросы алгебры*. Математический сборник. 2017. **208**(11), 90–138. **100**
- [66] Разборов А. А. *Нижние оценки монотонной сложности логического перманента*. Математические заметки. 1985. **37**(6), 887–900. **131**
- [67] Редькин Н. П. *Доказательство минимальности некоторых схем из функциональных элементов*. Проблемы кибернетики. Вып. 23. М.: Наука, 1970, 83–101. **12**
- [68] Редькин Н. П. *О реализации монотонных функций контактными схемами*. Проблемы кибернетики. Вып. 35. М.: Наука, 1979, 87–110. **39**, **40**
- [69] Редькин Н. П. *О минимальной реализации двоичного сумматора*. Проблемы кибернетики. Вып. 38. М.: Наука, 1981, 181–216. **13**, **83**, **85**
- [70] Редькин Н. П. *Обобщенная сложность линейных булевых функций*. Дискретная математика. 2018. **30**(4), 88–96. **12**
- [71] Рохлина М. М. *О схемах, повышающих надежность*. Проблемы кибернетики. Вып. 23. М.: Наука, 1970, 295–301. **109**
- [72] Румянцев П. В. *О сложности реализации мультиплексорной функции схемами из функциональных элементов*. Тез. докладов XIV Междунар. конф. «Проблемы теоретической кибернетики» (Пенза, 2005). М.: изд. мех.-мат. ф-та МГУ, 2005, 133. **42**
- [73] Рычков К. Л. *О нижних оценках сложности параллельно-последовательных контактных схем, реализующих линейные булевые функции*. Сибирский журнал исследования операций. 1994. **1**(4), 33–52. **22**

- [74] Рычков К. Л. *О нижних оценках формульной сложности линейной булевой функции*. Сибирские электронные математические известия. 2014. **11**, 165–184. **22**
- [75] Сапоженко А. А. *О сложности дизъюнктивных нормальных форм, получаемых с помощью градиентного алгоритма*. Дискретный анализ. Вып. 21. Новосибирск: Ин-т математики СО АН СССР, 1972, 62–71. **145**
- [76] Сапоженко А. А. *Проблема Дедекинда и метод граничных функционалов*. Математические вопросы кибернетики. Вып. 9. М.: Физматлит, 2000, 161–220. **39**
- [77] Селезнева С. Н. *Нижняя оценка сложности нахождения полиномов булевых функций в классе схем с разделенными переменными*. Прикладная математика и информатика. Труды ф-та ВМиК МГУ. Т. 40. М.: МАКС пресс, 2012, 97–104. **143**
- [78] Селезнева С. Н. *О длине булевых функций в классе полиномиальных форм с аффинными множителями в слагаемых*. Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. 2014. (2), 34–38. **146**
- [79] Селезнева С. Н. *Порядок длины функций алгебры логики в классе псевдополиномиальных форм*. Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. 2016. (3), 27–31. **146**, **148**
- [80] Сергеев И. С. *О глубине схем для многократного сложения и умножения чисел*. Матер. VI молодежной научной школы по дискретной математике и ее приложениям (Москва, 2007). Том II. М.: Изд. ИПМ РАН, 2007, 40–45. **139**, **140**, **141**
- [81] Сергеев И. С. *О построении схем для перехода между полиномиальными и нормальными базисами конечных полей*. Дискретная математика. 2007. **19**(3), 89–101. **80**, **81**
- [82] Сергеев И. С. *О сложности градиента рациональной функции*. Дискретный анализ и исследование операций. Серия 1. 2007. **14**(4), 57–75. **101**, **103**, **104**
- [83] Сергеев И. С. *Быстрые алгоритмы для элементарных операций с комплексными степенными рядами*. Дискретная математика. 2010. **22**(1), 17–49. **57**
- [84] Сергеев И. С. *О минимальных параллельных префиксных схемах*. Вестник Московского университета. Серия 1. Математика. Механика. 2011. (5), 48–51. **28**
- [85] Сергеев И. С. *Верхние оценки глубины симметрических булевых функций*. Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. 2013. **4**, 39–44. **93**
- [86] Сергеев И. С. *Верхние оценки сложности формул для симметрических булевых функций*. Известия вузов. Математика. 2014. (5), 38–52. **93**
- [87] Сергеев И. С. *О схемной сложности стандартного метода умножения чисел и метода Карацубы*. Тр. XXII Междунар. научно-технической конф. «Информационные средства и технологии» (Москва, 2014). Том 3. М.: Изд. дом МЭИ, 2014, 180–187. **24**
- [88] Сергеев И. С. *О сложности и глубине формул для симметрических булевых функций*. Вестник Московского университета. Серия 1. Математика. Механика. 2016. (3), 53–57. **71**, **93**, **109**
- [89] Сергеев И. С. *Верхние оценки сложности и глубины формул для MOD-функций*. Дискретная математика. 2016. **28**(2), 108–116. **43**, **44**, **71**, **90**, **91**
- [90] Сергеев И. С. *Вентильные схемы ограниченной глубины*. Дискретный анализ и исследование операций. 2018. **25**(1), 120–141. **39**

- [91] Сергеев И. С. *О сложности схем и формул ограниченной глубины над базисом из многовходовых элементов*. Дискретная математика. 2018. **30**(2), 120–137. **149**
- [92] Сергеев И. С. *О соотношении между глубиной и сложностью монотонных булевых формул*. Дискретный анализ и исследование операций. 2019. **26**(4), 108–120. **52**
- [93] Сергеев И. С. *О сложности монотонных схем для пороговых симметрических булевых функций*. Дискретная математика. 2020. **32**(1), 81–109. **32**, **125**
- [94] Сергеев И. С. *Формульная сложность линейной функции в k-арном базисе*. Математические заметки. 2021. **109**(3), 419–435. **45**
- [95] Сергеев И. С. *О мультипликативной сложности многочленов*. Дискретная математика. 2022. **34**(3), 85–89. **128**
- [96] Смирнов А. В. *О билинейной сложности и практических алгоритмах умножения матриц*. Журнал вычислительной математики и математической физики. 2013. **53**(12), 1970–1984. **60**, **89**, **100**
- [97] Ткачев Г. А. *О сложности реализации одной последовательности булевых функций схемами из функциональных элементов и π-схемами при дополнительных ограничениях на структуру схем*. Комбинаторно-алгебраические методы в прикладной математике. Горький: изд-во Горьк. ун-та, 1980, 161–207. **142**
- [98] Тоом А. Л. *О сложности схемы из функциональных элементов, реализующей умножение целых чисел*. Доклады АН СССР. 1963. **150**(3), 496–498. **24**, **72**
- [99] Угольников А. Б. *О реализации монотонных функций схемами из функциональных элементов*. Проблемы кибернетики. Вып. 31. М.: Наука, 1976, 167–185. **39**
- [100] Улиг Д. *О синтезе самокорректирующих схем из функциональных элементов с малым числом надежных элементов*. Математические заметки. 1974. **15**(6), 937–944. **116**
- [101] Фурман М. Е. *О применении метода быстрого перемножения матриц в задаче нахождения транзитивного замыкания графа*. Доклады АН СССР. 1970. **194**(3), 524. **70**
- [102] Хасин Л. С. *Оценки сложности реализации монотонных симметрических функций формулами в базисе  $\vee, \&, \neg$* . Доклады АН СССР. 1969. **189**(4), 752–755. **105**, **106**
- [103] Храпченко В. М. *Об асимптотической оценке времени сложения параллельного сумматора*. Проблемы кибернетики. Вып. 19. М.: Наука, 1967, 107–120. **29**, **30**, **52**
- [104] Храпченко В. М. *Об одном методе получения нижних оценок сложности π-схем*. Математические заметки. 1971. **10**(1), 83–92. **22**, **23**, **109**
- [105] Храпченко В. М. *О сложности реализации симметрических функций формулами*. Математические заметки. 1972. **11**(1), 109–120. **93**
- [106] Храпченко В. М. *О соотношении между сложностью и глубиной формул*. Методы дискретного анализа в синтезе управляющих систем. Вып. 32. Новосибирск: ИМ СО АН СССР, 1978, 76–94. **52**
- [107] Храпченко В. М. *О соотношении между сложностью и глубиной формул в базисе, содержащем медиану*. Методы дискретного анализа в изучении булевых функций и графов. Вып. 37. Новосибирск, ИМ СО АН СССР, 1981, 77–84. **49**
- [108] Храпченко В. М. *Об одной из возможностей уточнения оценок для задержки параллельного сумматора*. Дискретный анализ и исследование операций. Серия 1. 2007. **14**(1), 86–93. **99**

- [109] Чашкин А. В. *О сложности булевых матриц, графов и соответствующих им булевых функций*. Дискретная математика. 1994, **6**(2), 43–73. [96](#)
- [110] Чашкин А. В. *Дискретная математика*. М.: Академия, 2012. [32](#), [96](#)
- [111] Чашкин А. В. *О вычислении монотонных булевых функций*. Дискретная математика и ее приложения: сборник лекций молодежных научных школ по дискретной математике и ее приложениям. Вып. VIII. М.: Изд. ИПМ РАН, 2016, 30–44. [39](#)
- [112] Черухин Д. Ю. *О реализации линейной функции формулами в различных базисах*. Вестник Московского университета. Серия 1. Математика. Механика. 2001. (6), 15–19. [44](#)
- [113] Черухин Д. Ю. *К вопросу о логическом представлении счётчика чётности*. Неформальная наука. 2008. (2), 14–23. [22](#)
- [114] Яблонский С. В. *Реализация линейной функции в классе П-схем*. Доклады АН СССР. 1954. **94**(5), 805–806. [22](#)
- [115] Яблонский С. В., Козырев В. П. *Математические вопросы кибернетики*. «Информационные материалы» Научного совета по комплексной проблеме «Кибернетика» АН СССР. Вып. 19а. М., 1968, 3–15. [49](#)
- [116] Яблонский С. В. *Введение в дискретную математику*. М.: Наука, 1986. [96](#)
- [117] Agrawal M., Vinay V. *Arithmetic circuits: A chasm at depth four*. Proc. FOCS (Philadelphia, 2008). Los Alamitos: IEEE, 2008, 67–75. [154](#), [157](#)
- [118] Ajtai M.  *$\Sigma_1^1$ -formulae on finite structures*. Annals of Pure and Applied Logic. 1983. **24**(1), 1–48. [142](#)
- [119] Ajtai M., Komlós J., Szemerédi E. *Sorting in  $c \log n$  parallel steps*. Combinatorica. 1983. **3**(1), 1–19. [32](#), [60](#), [63](#), [68](#), [94](#)
- [120] Ajtai M., Komlós J., Szemerédi E. *An  $O(n \log n)$  sorting network*. Proc. STOC (Boston, 1983). NY: ACM, 1983, 1–9. [32](#), [60](#), [61](#), [63](#), [68](#)
- [121] Alman J., Duan R., Vassilevska Williams V., Xu Y., Xu Z., Zhou R. *More asymmetry yields faster matrix multiplication*. 2024. arXiv:2404.16349v1. [77](#), [81](#), [112](#), [115](#), [123](#)
- [122] Alman J., Guan Y., Padaki A. *Smaller low-depth circuits for Kronecker powers*. Proc. SODA (Florence, 2023). SIAM, 2023, 4159–4187. [144](#)
- [123] Alman J., Rao K. *Faster Walsh-Hadamard and Discrete Fourier transforms from matrix non-rigidity*. Proc. STOC (Orlando, Florida, 2023). NY: ACM, 2023, 455–462. [88](#)
- [124] Alon N., Schieber B. *Optimal preprocessing for answering on-line product queries*. Tech. report, 1987. <http://www.math.tau.ac.il/~haimk/adv-ds-2008> [113](#)
- [125] Amano K. *Bounds on the size of small depth circuits for approximating majority*. Proc. ICALP (Rhodes, Greece, 2009). LNCS. 2009. **5555**, 59–70. [150](#)
- [126] Amano K. *Integer complexity and mixed binary-ternary representation*. Proc. ISAAC (Seoul, 2022). LIPIcs. 2022. **248**, Art. 29. [17](#)
- [127] Barrett P. *Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor*. Advances in Cryptology. Proc. CRYPTO (Santa Barbara, 1986). LNCS. 1987. **263**, 311–323. [92](#)
- [128] Batcher K. E. *Sorting networks and their applications*. Proc. AFIPS spring joint comput. conf. (Atlantic City, 1968). **32**. NY: AFIPS, 1968, 307–314. [32](#), [60](#), [68](#)
- [129] Baur W., Strassen V. *The complexity of partial derivatives*. Theor. Comput. Sci. 1983. **22**, 317–330. [Перевод: Баур В., Штрассен Ф. *Сложность частных производных*. Кибернетический сборник. Новая серия. Вып. 22. М.: Мир, 1985, 3–18.] [100](#), [102](#), [103](#)

- [130] Beals R. *Improved construction of negation-limited circuits*. DIMACS Tech. report 95-31. Princeton Univ., 1995. [93](#), [94](#)
- [131] Beals R., Nishino T., Tanaka K. *On the complexity of negation-limited boolean networks*. SIAM J. Comput. 1998. **27**(5), 1334–1347. [93](#)
- [132] Beame P. W., Cook S. A., Hoover H. J. *Log depth circuits for division and related problems*. SIAM J. Comput. 1986. **15**(4), 994–1003. [Перевод: Бим П. У., Кук С. А., Гувер Г. Дж. *Схемы логарифмической глубины для деления и связанных с ним проблем*. Кибернетический сборник. Вып. 28. М.: Мир, 1991, 134–150.] [74](#), [75](#), [76](#)
- [133] Bellman R. *On a routing problem*. Quarterly of Appl. Math. 1958. **16**, 87–90. [17](#)
- [134] Bellman R. *Dynamic programming treatment of the travelling salesman problem*. J. ACM. 1962. **9**(1), 61–63. [18](#)
- [135] Beniamini G., Cheng N., Holtz O., Karstadt E., Schwartz O. *Sparsifying the operators of fast matrix multiplication algorithms*. 2020. arXiv:2008.03759v1. [89](#)
- [136] Bernstein D. J. *Multidigit multiplication for mathematicians*. 2001. <http://cr.yp.to/papers.html#m3> [74](#)
- [137] Bernstein D. J. *Computing logarithm intervals with the arithmetic-geometric-mean iteration*. 2003. <http://cr.yp.to/papers.html#logagm> [69](#)
- [138] Bernstein D. J. *The tangent FFT*. Proc. AAECC (Bangalore, 2007). LNCS. 2007. **4851**, 291–300. [87](#)
- [139] Bernstein D. J. *Fast multiplication and its applications*. Algorithmic Number Theory, MSRI Publ. 2008. **44**, 325–384. [74](#)
- [140] Bernstein D. J. *Batch binary Edwards*. Advances in Cryptology. Proc. CRYPTO (Santa Barbara, 2009). LNCS. 2009. **5677**, 317–336. [24](#)
- [141] Bernstein D. J., Yang B.-Y. *Fast constant-time gcd computation and modular inversion*. IACR TCHES. 2019. **3**, 340–398. [31](#)
- [142] Bhargava V., Ghosh S., Kumar M., Mohapatra C. K. *Fast, algebraic multivariate multipoint evaluation in small characteristic and applications*. Proc. SODA (Rome, 2022). NY: ACM, 2022, 403–415. [79](#)
- [143] Bini D. *Relations between exact and approximate bilinear algorithms*. Applications. Calcolo. 1980. **17**, 87–97. [58](#), [60](#), [100](#)
- [144] Bini D., Capovani M., Lotti G., Romani F.  *$O(n^{2.7799})$  complexity for  $n \times n$  approximate matrix multiplication*. Inform. Process. Lett. 1979. **8**(5), 234–235. [58](#), [100](#)
- [145] Bini D., Pan V. Y. *Polynomial and matrix computations*. Vol. 1. Boston: Birkhäuser, 1994. [2](#), [110](#), [112](#)
- [146] Blelloch G. E. *Prefix sums and their applications*. in: Synthesis of parallel algorithms. San Francisco: Morgan Kaufmann, 1993, 35–60. [28](#)
- [147] Bloniarz P. A. *The complexity of monotone Boolean functions and an algorithm for finding shortest paths in a graph*. Ph.D. thesis. Tech. report no. 238. Lab. for Computer Science, MIT, 1979. [124](#)
- [148] Bodrato M. *A Strassen-like matrix multiplication suited for squaring and higher power computation*. Proc. ISSAC (Munich, 2010). NY: ACM, 2010, 273–280. [88](#), [89](#)
- [149] Boppana R. B. *Threshold functions and bounded depth monotone circuits*. Proc. STOC (Washington, 1984). NY: ACM, 1984, 475–479. [149](#), [150](#), [152](#)
- [150] Boppana R. B. *Amplification of probabilistic Boolean formulas*. Proc. FOCS (Portland, 1985). Los Alamitos: IEEE, 1985, 20–29. [109](#)
- [151] Borodin A., von zur Gathen J., Hopcroft J. *Fast parallel matrix and GCD computations*. Inform. and Control. 1982. **52**, 241–256. [111](#)

- [152] Bostan A., Schost É. *Polynomial evaluation and interpolation on special sets of points*. J. Complexity. 2005. **21**, 420–446. [79](#), [94](#)
- [153] Boyar J., Find M. *Cancellation-free circuits in unbounded and bounded depth*. Theor. Comput. Sci. 2015. **590**, 17–26. [144](#)
- [154] Boyar J., Peralta R. *Tight bounds for the multiplicative complexity of symmetric functions*. Theor. Comput. Sci. 2008. **396**, 223–246. [160](#)
- [155] Brauer A. *On addition chains*. Bull. AMS. 1939. **45**, 736–739. [34](#)
- [156] Brent R. P. *The parallel evaluation of general arithmetic expressions in logarithmic time*. J. ACM. 1974. **21**(2), 201–206. [49](#)
- [157] Brent R. P. *Multiple-precision zero-finding methods and the complexity of elementary function evaluation*. in: Analytic Computational Complexity. NY: Academic Press, 1975, 151–176. [54](#), [68](#)
- [158] Brent R. P., Kuck D. J., Maruyama K. *The parallel evaluation of arithmetic expressions without division*. IEEE Trans. Comp. 1973. C-**22**, 532–534. [49](#), [50](#)
- [159] Brent R. P., Kung H. T. *Fast algorithms for manipulating formal power series*. J. ACM. 1978. **25**(4), 581–595. [76](#), [80](#)
- [160] Brent R. P., Zimmermann P. *Modern computer arithmetic*. Cambridge University Press, 2010. [2](#), [69](#)
- [161] Bshouty N. H. *On the additive complexity of  $2 \times 2$  matrix multiplication*. Inform. Proc. Letters. 1995. **56**(6), 329–335. [24](#)
- [162] Cayley A. *A theorem on trees*. Quart. J. Pure Appl. Math. 1889. **23**, 376–378. [21](#)
- [163] Cenk M., Hasan M. A. *On the arithmetic complexity of Strassen-like matrix multiplications*. J. Symb. Comput. 2017. **80**, 484–501. [89](#)
- [164] Chen X., Kayal N., Wigderson A. *Partial derivatives in arithmetic complexity and beyond*. Found. Trends in Theor. Comput. Sci. 2011. **6**(1–2), 1–138. [127](#)
- [165] Chin A. *On the depth complexity of the counting functions*. Inform. Proc. Letters. 1990. **35**, 325–328. [43](#), [44](#)
- [166] Chistikov D., Iván S., Lubiw A., Shallit J. *Fractional coverings, greedy coverings, and rectifier networks*. Proc. STACS (Hannover, 2017). LIPIcs. 2017. **66**, Art. 23. [144](#), [146](#)
- [167] Chistov A. L. *Fast parallel calculation of the rank of matrices over a field of arbitrary characteristic*. Fundamentals of computation theory. LNCS. 1985. **199**, 63–69. [110](#)
- [168] Chockler H., Zwick U. *Which bases admit non-trivial shrinkage of formulae?* Comput. Complexity. 2001. **10**, 28–40. [44](#), [45](#)
- [169] Chvátal V. *Lecture notes on the new AKS sorting network*. Tech. report DCS-TR-94. Rutgers Univ., 1992. [68](#)
- [170] Commentz-Walter B. *Size-depth tradeoff in monotone Boolean formulae*. Acta Inf. 1979. **12**, 227–243. [99](#)
- [171] Commentz-Walter B., Sattler J. *Size-depth tradeoff in non-monotone Boolean formulae*. Acta Inf. 1979. **14**, 257–269. [99](#)
- [172] Cook S. *On the minimum computation time of functions*. Ph. D. Thesis, Harvard Univ., 1966. [30](#), [53](#), [74](#)
- [173] Cooley J. W., Tukey J. W. *An algorithm for the machine calculation of complex Fourier series*. Math. Comp. 1965. **19**, 297–301. [25](#), [26](#)
- [174] Cooper J. N., Ellis R. B., Kahng A. B. *Asymmetric binary covering codes*. J. Comb. Theory, Ser. A. 2002. **100**, 232–249. [147](#)

- [175] Coppersmith D. *Rapid multiplication of rectangular matrices*. SIAM J. Comput. 1982. **11**(3), 467–471. [116](#)
- [176] Coppersmith D., Schieber B. *Lower bounds on the depth of monotone arithmetic computations*. J. Complexity. 1999. **15**(1), 17–29. [51](#)
- [177] Coppersmith D., Winograd S. *On the asymptotic complexity of matrix multiplication*. SIAM J. Comput. 1982. **11**(3), 472–492. [121](#)
- [178] Coppersmith D., Winograd S. *Matrix multiplication via arithmetic progressions*. J. Symb. Comput. 1990. **9**, 251–280. [115](#), [121](#), [123](#)
- [179] Crandall R., Fagin B. *Discrete weighted transforms and large-integer arithmetic*. Math. Comp. 1994. **62**(205), 305–324. [80](#)
- [180] Demenkov E., Kojevnikov A., Kulikov A., Yaroslavtsev G. *New upper bounds on the Boolean circuit complexity of symmetric functions*. Inform. Proc. Letters. 2010. **110**(7), 264–267. [83](#), [84](#)
- [181] Díaz J., Serna M., Thilikos D. M. *Counting  $H$ -colorings of partial  $k$ -trees*. Theor. Comput. Sci. 2002, **281**, 291–309. [133](#)
- [182] Dubiner M., Zwick U. *Amplification and percolation (probabilistic Boolean functions)*. Proc. FOCS (Pittsburgh, 1992). Los Alamitos: IEEE, 1992, 258–267. [109](#)
- [183] Duhamel P., Hollmann H. *Split-radix FFT algorithm*. Electronics Letters. 1984. **20**, 14–16. [86](#)
- [184] Dunne P. E. *The complexity of Boolean networks*. San Diego: Academic Press, 1988. [2](#)
- [185] Edmonds J. *Systems of distinct representatives and linear algebra*. J. Research Nat. Bureau Stand. — B. Math. and Math. Phys. 1967. **71B**(4), 241–245. [111](#)
- [186] Erdős P. *Remarks on number theory III: On addition chains*. Acta Arithm. 1960. **6**, 77–81. [35](#)
- [187] Feder T., Motwani R. *Clique partitions, graph compression and speeding-up algorithms*. J. Comp. System Sci. 1995. **51**, 261–272. [131](#)
- [188] Fischer I. *Sums of like powers of multivariate linear forms*. Mathematics Magazine. 1994. **67**(1), 59–61. [158](#)
- [189] Fischer M. J. *The complexity of negation-limited networks — a brief survey*. Proc. Automata Theory and Formal Lang. Conf. (Kaiserslautern, 1975). LNCS. 1975. **33**, 71–82. [93](#)
- [190] Fischer M. J., Meyer A. R. *Boolean matrix multiplication and transitive closure*. Proc. SWAT (East Lansing, 1971). Washington: IEEE, 1971, 129–131. [70](#)
- [191] Floyd R. W. *Algorithm 97, shortest path*. Comm. ACM. 1962. **5**, 345. [21](#)
- [192] Fomin S., Grigoriev D., Koshevoy G. *Subtraction-free complexity, cluster transformations, and spanning trees*. Found. Comput. Math. 2016. **15**, 1–31. [19](#)
- [193] Ford L. R. *Network flow theory*. The Rand Corp., 1956, Report P-923. [17](#)
- [194] Fürer M. *Faster integer multiplication*. SIAM J. Comput. 2009, **39**(3), 979–1005. [74](#)
- [195] Furst M., Saxe J., Sipser M. *Parity, circuits, and the polynomial time hierarchy*. Math. Syst. Theory. 1984. **17**, 13–27. [142](#)
- [196] Gabber O., Galil Z. *Explicit constructions of linear size superconcentrators*. Proc. FOCS (San Juan, Puerto-Rico, 1979). Los Alamitos: IEEE, 1979, 364–370. [63](#)
- [197] Galbiati G., Fischer M. J. *On the complexity of 2-output Boolean networks*. Theor. Comput. Sci. 1981. **16**, 177–185. [117](#)
- [198] Galil Z., Pan V. *Parallel evaluation of the determinant and of the inverse of a matrix*. Inform. Proc. Letters. 1989. **30**, 41–45. [112](#)

- [199] Gao S., von zur Gathen J., Panario D., Shoup V. *Algorithms for exponentiation in finite fields*. J. Symb. Comput. 2000. **29**, 879–889. [82](#)
- [200] von zur Gathen J., Gerhard J. *Modern computer algebra*. Cambridge Univ. Press, 1999. [2](#)
- [201] von zur Gathen J., Shoup V. *Computing Frobenius maps and factoring polynomials*. Comput. Complexity. 1992. **2**, 187–224. [81](#)
- [202] Gerhard J. *Modular algorithms in symbolic summation and symbolic integration*. Berlin, Heidelberg: Springer–Verlag, 2004. [94](#)
- [203] Giesbrecht M., Jamshidpey A., Schost E. *Subquadratic-time algorithms for normal bases*. Comput. Complexity. 2021. **30**, Art. 5. [82](#)
- [204] Good I. J. *The interaction algorithm and practical Fourier analysis*. J. R. Statist. Soc. B. 1958. **20**(2), 361–372; 1960. **22**(2), 372–375. [25](#)
- [205] Goodrich M. T. *Zig-zag sort: a simple deterministic data-oblivious sorting algorithm running in  $O(n \log n)$  time*. Proc. STOC (New York, 2014). NY: ACM, 2014, 684–693. [68](#)
- [206] Grigoriev D., Razborov A. A. *Exponential complexity lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields*. Applicable Algebra in Eng. Comm. Comput. 2000. **10**(6), 465–487. [160](#)
- [207] Gupta A., Kamath P., Kayal N., Saptharishi R. *Arithmetic circuits: A chasm at depth 3*. SIAM J. Comput. 2016. **45**(3), 1064–1079. [157](#), [158](#), [159](#)
- [208] Gupta A., Mahajan S. *Using amplification to compute majority with small majority gates*. Comput. Complexity. 1996. **6**, 46–63. [109](#)
- [209] Grove E. *Proofs with potential*. Ph.D. thesis. Univ. of California, Berkeley, 1993. [49](#)
- [210] Hadas T., Schwartz O. *Towards practical fast matrix multiplication based on trilinear aggregation*. Proc. ISSAC (Tromsø, 2023). NY: ACM, 2023, 289–297. [100](#)
- [211] Harvey D., van der Hoeven J. *Integer multiplication in time  $O(n \log n)$* . Annals of Math. 2021. **193**(2), 563–617. [30](#), [74](#), [76](#)
- [212] Harvey D., van der Hoeven J. *Polynomial multiplication over finite fields in time  $O(n \log n)$* . J. ACM. 2022. **69**(2), Art. 12. [74](#)
- [213] Harvey D., van der Hoeven J., Lecerf G. *Faster polynomial multiplication over finite fields*. J. ACM. 2017. **63**(6), Art. 52. [74](#)
- [214] Håstad J. *Computational limitations of small-depth circuits*. MIT Press, 1986. [149](#)
- [215] Håstad J. *Notes for the course advanced algorithms*. 2000. <http://www.csc.kth.se/~johanh/algnotes.pdf> [54](#)
- [216] Hastad J., Leighton T. *Division in  $O(\log n)$  depth using  $O(n^{1+\epsilon})$  processors*. 1986. <http://www.csc.kth.se/~johanh/paraldivision.pdf> [74](#), [76](#)
- [217] Heideman M. T. *Applications of multiplicative complexity theory to convolution and the discrete Fourier transform*. Ph.D. Thesis. Rice Univ., 1986. [26](#)
- [218] Held M., Karp R. M. *A dynamic programming approach to sequencing problems*. SIAM J. on Appl. Math. 1962. **10**, 196–210. [18](#)
- [219] Hermann A. *Faster circuits for And-Or paths and binary addition*. PhD. Thesis, Univ. Bonn, 2020. [99](#)
- [220] Hiltgen A. P., Paterson M. S.  *$PI_k$  mass production and an optimal circuit for the Nečiporuk slice*. Comput. Complexity. 1995. **5**(2), 132–154. [123](#)
- [221] van der Hoeven J. *Newton’s method and FFT trading*. J. Symb. Comput. 2010. **45**(8), 857–878. [55](#)
- [222] van der Hoeven J. *Optimizing the half-gcd algorithm*. 2022. arXiv:2212.12389v1. [31](#)

- [223] van der Hoeven J., Lecerf G. *Fast multivariate multi-point evaluation revisited*. J. Complexity. 2020. **56**, Art. 101405. [80](#)
- [224] Holmgren J., Rothblum R. *Linear-size Boolean circuits for multiselection*. Proc. CCC (Ann Arbor, USA, 2024). LIPIcs. 2024. **300**, Art. 11. [117](#)
- [225] Hoover H., Klawe M., Pippenger N. *Bounding fan-out in logical networks*. J. ACM. 1984. **31**(1), 13–18. [139](#)
- [226] Hopcroft J. E., Karp R. M. *An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs*. SIAM J. Comput. 1973. **2**(4), 225–231. [128](#), [131](#)
- [227] Jerrum M., Snir M. *Some exact complexity results for straight-line computations over semirings*. J. ACM. 1982. **29**(3), 874–897. [18](#), [19](#), [135](#)
- [228] Jimbo S., Maruoka A. *A method of constructing selection networks with  $O(\log n)$  depth*. SIAM J. Comput. 1996. **25**(4), 709–739. [32](#)
- [229] Johnson S. F., Frigo M. *A modified split-radix FFT with fewer arithmetic operations*. IEEE Trans. Signal Process. 2007. **55**(1), 111–119. [87](#)
- [230] Jukna S. *Extremal combinatorics: with applications in computer science*. Berlin, Heidelberg: Springer–Verlag, 2011. [40](#), [145](#)
- [231] Jukna S. *Boolean function complexity: advances and frontiers*. Berlin, Heidelberg: Springer–Verlag, 2012. [2](#), [23](#)
- [232] Jukna S., Seiwert H. *Greedy can beat pure dynamic programming*. Inform. Proc. Letters. 2019. **142**, 90–95. [19](#), [21](#)
- [233] Jukna S., Sergeev I. *Complexity of linear boolean operators*. Found. Trends in Theor. Comput. Sci. 2013. **9**(1), 1–123. [143](#), [144](#), [146](#), [153](#)
- [234] Kaltofen E., Shoup V. *Subquadratic-time factoring of polynomials over finite fields*. Math. Comput. 1998. **67**(223), 1179–1197. [76](#), [80](#), [81](#)
- [235] Kaltofen E., Singer M. F. *Size efficient parallel algebraic circuits for partial derivatives*. Proc. Intern. Conf. on Computer Algebra in Psysical Research (Dubna, 1990). Singapore: World Scientific, 1991, 133–145. [104](#)
- [236] Karppa M., Kaski P. *Engineering Boolean matrix multiplication for multiple-accelerator shared-memory architectures*. 2019. arXiv:1909.01554v1. [89](#)
- [237] Karstadt E., Schwartz O. *Matrix multiplication, a little faster*. J. ACM. 2020. **67**(1), Art. 1. [88](#), [89](#)
- [238] Kedlaya K. S., Umans C. *Fast polynomial factorization and modular composition*. SIAM J. Comput. 2011. **40**(6), 1767–1802. [80](#)
- [239] Kerr L. R. *The effect of algebraic structure on the computational complexity of matrix multiplication*. PhD. Thesis, Cornell Univ., Ithaca, N.Y., 1970. [24](#)
- [240] Kleiman M., Pippenger N. *An explicit conctructing of short monotone formulae for the monotone symmetric functions*. Theor. Comput. Sci. 1978. **7**(3), 325–332. [106](#)
- [241] Klein P., Paterson M. S. *Asymptotically optimal circuit for a storage access function*. IEEE Trans. on Computers. 1980. C-**29**(8), 737–738. [41](#)
- [242] Kleitman D. *On Dedekind's problem: the number of monotone Boolean functions*. Proc. AMS. 1969. **21**(3), 677–682. [Перевод: Клейтмен Д. *О проблеме Дедекинда: число монотонных булевых функций*. Кибернетический сборник. Вып. 7. М.: Мир, 1970, 43–52.] [39](#)
- [243] Kloks T. *Treewidth. Computations and approximations*. LNCS. **842**. Berlin: Springer, 1994. [133](#)
- [244] Knuth D. E. *The analysis of algorithms*. Actes du Congrès international des mathématiciens. Nice, 1970, **3**, 269–274. [30](#)

- [245] Komarath B., Pandey A., Rahul C. S. *Monotone arithmetic complexity of graph homomorphism polynomials*. Proc. ICALP (Paris, 2022). LIPIcs. 2022. **229**, Art. 83. 135
- [246] Kosaraju S. R. *Parallel evaluation of division-free arithmetic equations*. Proc. STOC (Berkeley, 1986). NY: ACM, 1986, 231–239. 29, 51
- [247] Kulikov A. S., Melanich O., Mihajin I. *A  $5n - o(n)$  lower bound on the circuit size over  $U_2$  of a linear Boolean function*. Proc. Conf. on Computability in Europe (Cambridge, 2012). LNCS. 2012. **7318**, 432–439. 96
- [248] Kulikov A. S., Mikhailin I., Mokhov A., Podolskii V. *Complexity of linear operators*. Proc. ISAAC (Shanghai, 2019). LIPIcs. 2019. **149**, Art. 17. 112, 114
- [249] Kung H. T. *Fast evaluation and interpolation*. Tech. report. Carnegie-Mellon Univ., Pittsburgh, 1973. 77
- [250] Lademan J. D. *A noncommutative algorithm for multiplying  $3 \times 3$  matrices using 23 multiplications*. Bull. AMS. 1976, **82**, 126–128. 60
- [251] Ladner R. E., Fischer M. J. *Parallel prefix computation*. J. ACM. 1980. **27**(4), 831–838. 26, 27
- [252] Lamagna E. A., Savage J. E. *Combinational complexity of some monotone functions*. Proc. SWAT (New Orleans, 1974). New Orleans: IEEE, 1974, 140–144. 60
- [253] Landsberg J. M. *The border rank of the multiplication of  $2 \times 2$  matrices is seven*. J. AMS. 2006. **19**(2), 447–459. 60
- [254] Lehmer D. H. *Euclid's algorithm for large numbers*. Amer. Math. Monthly. 1938. **45**, 227–233. 31
- [255] van Leijenhorst D. C. *A note on the formula size of the “mod k” functions*. Inform. Proc. Letters. 1987. **24**, 223–224. 71
- [256] Limaye N., Srinivasan S., Tavenas S. *Superpolynomial lower bounds against low-depth algebraic circuits*. Proc. FOCS (virtually, 2022). Los Alamitos: IEEE, 2022, 804–814. 142, 159
- [257] Linnainmaa S. *Taylor expansion of the accumulated rounding error*. Nordisk Tidskr. Informationsbehandling (BIT). 1976. **16**(2), 146–160. 103
- [258] Lovász L. *On determinants, matchings, and random algorithms*. Proc. Conf. Algebr. Arithm. Categ. Methods in Comput. Theory (Berlin/Wendisch-Rietz, 1979). Berlin: Akademie, 1979, 565–574. 111
- [259] Lovett S. *Computing polynomials with few multiplications*. Theory of Computing. 2011. **7**, 185–188. 127
- [260] Lundy T. J., van Buskirk J. *A new matrix approach to real FFTs and convolutions of length  $2^k$* . Computing. 2007. **80**, 23–45. 86
- [261] Mahajan M., Nimborkar P., Tawari A. *Shortest path length with bounded-alternation  $(\min, +)$  formulas*. Int. J. Advances in Engin. Sci. and Appl. Math. 2019. **11**, 68–74. 160
- [262] Mahler K., Popken J. *On a maximum problem in arithmetic*. Nieuw Archief voor Wiskunde (ser. 3). 1953. **1**(3), 1–15. 15
- [263] Martens J.-B. *Recursive cyclotomic factorization — a new algorithm for calculating the discrete Fourier transform*. IEEE Trans. on Acoustics, Speech, and Signal Processing. 1984. **32**, 750–761. 86
- [264] Massey J. L., Omura J. K. *Apparatus for finite fields computation*. US patent application. 1986. №4587627. 81

- [265] McColl W. F. *Some results on circuit depth*. Theory of computation. Report no. 18. Coventry, Univ. of Warwick, 1977. 71
- [266] Moenck R. T. *Fast computation of GCDs*. Proc. STOC (Austin, 1973). NY: ACM, 1973, 142–151. 31
- [267] Möller N. *On Schönhage's algorithm and subquadratic integer GCD computation*. Math. Comp. 2008. **77**, 589–607. 31
- [268] Montgomery P. L. *Modular multiplication without trial division*. Math. Comp. 1985. **44**, 519–521. 91, 92
- [269] Moore E. F. *The shortest path through a maze*. Proc. Intern. Sympos. on Theory of Switching (Cambridge, USA, 1957). Part II. Cambridge: Harvard Univ. Press, 1959, 285–292. 17
- [270] Moore E. F., Shannon C. E. *Reliable circuits using less reliable relays*. J. of the Franklin Institute. 1956. **262**(3), 191–208; **262**(4), 281–297. [Перевод: Надежные схемы из ненадежных реле. В сб. Шеннон К. Работы по теории информации и кибернетике. М.: ИЛ, 1963, 114–153.] 109
- [271] Morizumi H., Suzuki G. *Negation-limited inverters of limited size*. IEICE Trans. Inf. Syst. 2010. **E93-D**(2), 257–262. 94
- [272] Muller D. E. *Complexity in electronic switching circuits*. IRE Trans. Comput. 1956. **EC-5**(1), 15–19. 35
- [273] Oliveira I. C., Santhanam R., Srinivasan S. *Parity helps to compute majority*. Proc. Comput. Compl. Conf. (New Brunswick, 2019). LIPIcs. 2019. **137**, Art. 23. 149, 150, 152
- [274] Pan V. Ya. *Trilinear aggregating with implicit canceling for a new acceleration of matrix multiplication*. Comput. Math. Appl. 1982. **8**(1), 23–34. 100
- [275] Pan V. Y. *Structured matrices and polynomials: unified superfast algorithms*. Boston: Birkhäuser, 2001. 2
- [276] Paterson M. S. *Complexity of monotone networks for Boolean matrix product*. Theor. Comput. Sci. 1975. **1**, 13–20. [Перевод: Патерсон М. С. Сложность монотонных схем для булева умножения матриц. Кибернетический сборник. Вып. 15. М.: Мир, 1978, 28–37.] 24, 70
- [277] Paterson M. S. *Improved sorting networks with  $O(\log N)$  depth*. Algorithmica. 1990. **5**(1), 75–92. 61, 63, 68
- [278] Paterson M. S., Pippenger N., Zwick U. *Faster circuits and shorter formulae for multiple addition, multiplication and symmetric Boolean functions*. Proc. FOCS (St. Louis, 1990). Washington: IEEE, 1990, 642–650. 93
- [279] Paterson M. S., Pippenger N., Zwick U. *Optimal carry save networks*. LMS Lecture Notes Series. Boolean function complexity. Vol. 169. Cambridge Univ. Press, 1992, 174–201. 45, 46, 47, 49
- [280] Paterson M. S., Stockmeyer L. J. *On the number of nonscalar multiplications necessary to evaluate polynomials*. SIAM J. Comput. 1973. **2**, 60–66. 33, 127
- [281] Paterson M., Zwick U. *Shallow circuits and concise formulae for multiple addition and multiplication*. Comput. Complexity. 1993. **3**, 262–291. 49
- [282] Paul W. J. *Realizing Boolean functions on disjoint sets of variables*. Theor. Comput. Sci. 1976. **2**, 383–396. 115
- [283] Paul W. J. *A  $2.5n$ -lower bound on the combinational complexity of Boolean functions*. SIAM J. Comput. 1977. **6**(3), 427–443. [Перевод: Пауль В. Й. Нижняя оценка  $2.5n$  для комбинаторной сложности булевых функций. Кибернетический сборник. Вып. 16. М.: Мир, 1979, 23–44.] 42

- [284] Pippenger N. *The minimum number of edges in graphs with prescribed paths*. Math. Systems Theory. 1979. **12**, 325–346. [39](#)
- [285] Pippenger N. *On the evaluation of powers and monomials*. SIAM J. Comput. 1980. **9**(2), 230–250. [37](#), [39](#)
- [286] Pippenger N. *A formula for the determinant*. 2022. arXiv:2206.00134v1. [110](#), [157](#)
- [287] Preparata F. P., Muller D. E. *Efficient parallel evaluation of Boolean expressions*. IEEE Trans. Comp. 1976. C-**25**(5), 548–549. [51](#), [52](#)
- [288] Probert R. L. *On the additive complexity of matrix multiplication*. SIAM J. Comput. 1976. **5**(2), 187–203. [24](#)
- [289] Prüfer H. *Neuer Beweis eines Satzes über Permutationen*. Arch. Math. Phys. 1918. **27**, 742–744. [21](#)
- [290] Rabin M. O., Winograd S. *Fast evaluation of polynomials by rational preparation*. IBM Res. Rep. RC 3645. N.Y.: Yorktown Heights, 1971. [33](#)
- [291] Radhakrishnan J. *Entropy and counting*. IIT Kharagpur, Golden Jubilee Volume on Computational Mathematics, Modelling and Algorithms. New Delhi: Narosa Publ., 2001. [105](#)
- [292] Reif J., Tate S. *Optimal size integer division circuits*. SIAM J. Comput. 1990. **19**(5), 912–925. [76](#)
- [293] Riordan J., Shannon C. *The number of two-terminal series-parallel networks*. J. of Math. and Phys. 1942. **21**(2), 83–93. [Перевод: Число двухполюсных параллельно-последовательных сетей. В сб. Шеннон К. Работы по теории информации и кибернетике. М.: ИЛ, 1963, 46–58.] [126](#)
- [294] Rosser J. B., Schoenfeld L. *Approximate formulas for some functions of prime numbers*. Illinois J. Math. 1962. **6**, 64–94. [75](#), [112](#)
- [295] Roy B. *Transitivité et connexité*. C. R. Acad. Sci. Paris. 1959. **249**, 216–218. [21](#)
- [296] Ryser H. J. *Combinatorial Mathematics*. NY: Wiley, 1963. [158](#)
- [297] Saptharishi R. et al. *A survey of lower bounds in arithmetic circuit complexity*. Github survey. 2014–2021. ver. 9.0.3. <https://github.com/dasarmar/lowerbounds-survey> [158](#)
- [298] Saxena N. *Diagonal circuit identity testing and lower bounds*. Proc. ICALP (Reykjavik, 2008). LNCS. **5126**. Springer, 2008, 60–71. [158](#)
- [299] Schnorr C. P. *Zwei lineare untere Schranken für die Komplexität Boolescher Funktionen*. Computing. 1974. **13**(2), 155–171. [12](#)
- [300] Schnorr C. P. *A lower bound on the number of additions in monotone computations*. Theor. Comput. Sci. 1976, **2**, 305–315. [Перевод: Шнорр К. П. *Нижняя оценка числа сложений в монотонных вычислениях*. Кибернетический сборник. Вып. 18. М.: Мир, 1981, 5–20.] [135](#)
- [301] Schönhage A. *Schnelle Berechnung von Kettenbruchentwicklungen*. Acta Inf. 1971. **1**, 139–144. [30](#)
- [302] Schönhage A. *A lower bound for the length of addition chains*. Theor. Comput. Sci. 1975, **1**, 1–12. [35](#)
- [303] Schönhage A. *Partial and total matrix multiplication*. SIAM J. Comput. 1981. **10**(3), 434–455. [60](#), [117](#), [118](#), [119](#), [121](#)
- [304] Schönhage A. *Fast reduction and composition of binary quadratic forms*. Proc. ISSAC (Bonn, 1991). NY: ACM, 1991, 128–133. [31](#)
- [305] Schönhage A., Strassen V. *Schnelle Multiplikation großer Zahlen*. Computing. 1971. **7**(3–4), 271–282. [Перевод: Шёнхаге А., Штрассен В. *Быстрое умножение больших чисел*. Кибернетический сборник. Вып. 10. М.: Мир, 1973, 87–98.] [72](#), [74](#)

- [306] Schwartz J. T. *Fast probabilistic algorithms for verification of polynomial identities*. J. ACM. 1980. **27**(4), 701–717. [111](#)
- [307] Seiferas J. *Sorting networks of logarithmic depth, further simplified*. Algorithmica. 2009. **53**(3), 374–384. [см. также: Seiferas J. *AKS sorting networks*. Manuscript, 2017. доступно на <http://www.researchgate.net/profile/Joel-Seiferas>] [61](#), [68](#)
- [308] Selezneva S. N. *On the multiplicative complexity of Boolean functions*. Fundamenta Informaticae. 2016. **145**, 399–404. [137](#)
- [309] Sergeev I. S. *On the complexity of parallel prefix circuits*. ECCC report TR13–041. 2013. [28](#)
- [310] Sergeev I. S. *Notes on the complexity of coverings for Kronecker powers of symmetric matrices*. 2022. arXiv:2212.01776v1. [144](#)
- [311] Shannon C. E. *The synthesis of two-terminal switching circuits*. Bell Systems Technical J. 1949. **28**(1), 59–98. [Перевод: *Синтез двухполюсных переключательных схем*. В сб. Шеннон К. Работы по теории информации и кибернетике. М.: ИЛ, 1963, 59–105.] [35](#)
- [312] Stehlé D., Zimmermann P. *A binary recursive GCD algorithm*. Proc. ANTS (Burlington, USA, 2004). LNCS. 2004. **3076**, 411–425. [31](#)
- [313] Stein J. *Computational problems associated with Racah algebra*. J. Comput. Physics. 1967. **1**, 397–405. [13](#)
- [314] Stein S. K. *Two combinatorial covering problems*. J. Combin. Theory (A). 1974. **16**, 391–397. [145](#)
- [315] Stockmeyer L. J. *On the combinational complexity of certain symmetric Boolean functions*. Math. Systems Theory. 1977. **10**, 323–336. [Перевод: Стокмейер Л. Дж. *О комбинационной сложности некоторых симметрических булевых функций*. Кибернетический сборник. Вып. 16. М.: Мир, 1979, 45–61.] [84](#)
- [316] Strassen V. *Gaussian elimination is not optimal*. Numer. Math. 1969. **13**(4), 354–356. [Перевод: Штрассен Ф. *Алгоритм Гаусса не оптимальен*. Кибернетический сборник. Новая серия. Вып. 7. М.: Мир, 1970, 67–70.] [24](#), [70](#), [88](#), [103](#), [112](#)
- [317] Strassen V. *Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten*. Numer. Math. 1973. **20**, 238–251. [55](#), [154](#)
- [318] Strassen V. *Relative bilinear complexity and matrix multiplication*. J. für die reine und angewandte Math. 1987. **1987**(375–376), 406–443. [121](#)
- [319] Tanaka K., Nishino T. *On the complexity of negation-limited Boolean networks*. Proc. STOC (Montreal, 1994). NY: ACM, 1994, 38–47. [93](#)
- [320] Tardos É. *The gap between monotone and non-monotone circuit complexity is exponential*. Combinatorica. 1987. **7**(4), 141–142. [131](#)
- [321] Tavenas S. *Improved bounds for reduction to depth 4 and depth 3*. Inform. and Comput. 2015. **240**, 2–11. [154](#), [156](#), [157](#), [158](#), [159](#)
- [322] Uhlig D. *Zur Parallelberechnung Boolescher Funktionen*. TR Ing. Hochschule Mittweida, 1984. [117](#)
- [323] Uhlig D. *Networks computing Boolean functions for multiple input values*. LMS Lecture Notes Series. Boolean function complexity. Vol. 169. Cambridge Univ. Press, 1992, 165–173. [117](#)
- [324] Umans C. *Fast polynomial factorization and modular composition in small characteristic*. Proc. STOC (Victoria, Canada, 2008). NY: ACM, 2008, 481–490. [77](#), [79](#), [81](#)

- [325] Valiant L. G. *Short monotone formulae for the majority function*. J. Algorithms. 1984. **5**, 363–366. [Перевод: Вэльянт Л. *Простые монотонные формулы для функции голосования*. Кибернетический сборник. Вып. 24. М.: Мир, 1987, 97–100.] **93**, **106**, **150**
- [326] Valiant L. G., Skyum S., Berkowitz S., Rackoff C. *Fast parallel computation of polynomials using few processors*. SIAM J. Comput. 1983. **12**(4), 641–644. **154**
- [327] Vetterli M., Nussbaumer H. J. *Simple FFT and DCT algorithms with reduced number of operations*. Signal Processing. 1984, **6**, 262–278. **86**
- [328] Warshall S. *A theorem on boolean matrices*. J. ACM. 1962. **9**, 11–12. **21**
- [329] Wegener I. *More on the complexity of slice functions*. Theor. Comput. Sci. 1986. **43**, 201–211. **123**
- [330] Wegener I. *The complexity of Boolean functions*. Stuttgart: Wiley–Teubner, 1987. **2**, **23**, **117**, **124**, **125**, **131**
- [331] Winograd S. *On multiplication of  $2 \times 2$  matrices*. Linear Algebra and Appl. 1971. **4**, 381–388. **24**, **57**
- [332] Winograd S. *On the multiplicative complexity of the discrete Fourier transform*. Advances in Math. 1979. **32**(2), 83–117. **26**
- [333] Yao A. C. *A study of concrete computational complexity*. Ph.D. thesis. Tech. Report UIUCDCS-R-75-716. Urbana-Champaign, Univ. Illinois, 1975. **32**
- [334] Yavne R. *An economical method for calculating the discrete Fourier transform*. Proc. Fall Joint Computer Conf. (San Francisco, 1968). Part I. NY: ACM, 1968, 115–125. **85**, **86**
- [335] Zelinsky J. *Upper bounds on integer complexity*. 2022. arXiv:2211.02995v1. **15**