Институт прикладной математики им. М. В. Келдыша
Российской Академии Наук
Московский государственный университет им. М.В. Ломоносова
Механико-математический факультет

ДИСКРЕТНАЯ МАТЕМАТИКА И ЕЕ ПРИЛОЖЕНИЯ

СБОРНИК ЛЕКЦИЙ МОЛОДЕЖНЫХ НАУЧНЫХ ШКОЛ ПО ДИСКРЕТНОЙ МАТЕМАТИКЕ И ЕЕ ПРИЛОЖЕНИЯМ

V

Москва 2009

М34
УДК 519.7

Издание осуществлено при поддержке Российского фонда фундаментальных исследований по проекту 09–01–06027

М34 Дискретная математика и ее приложения: Сборник лекций молодежных научных школ по дискретной математике и ее приложениям. Выпуск V. Под редакцией А. В. Чашкина. — М.: ИПМ им. М. В. Келдыша РАН, 2009. — 68 с.

Пятый выпуск лекций содержит содержит лекции, прочитанные на VII молодежной научной школе по дискретной математике и ее приложениям, проходившей в Москве с 18 по 23 мая 2009 г. при поддержке Российского фонда фундаментальных исследований (проект 09–01–06027). Для студентов, аспирантов и научных работников в области дискретной математики и математической кибернетики.

Научное издание

ДИСКРЕТНАЯ МАТЕМАТИКА
И ЕЕ ПРИЛОЖЕНИЯ

Сборник лекций
Выпуск V

Под общей редакцией А. В. ЧАШКИНА

Ответственный за выпуск *А. Д. Яшунский*

АЛГОРИТМЫ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ

С. Б. Гашков*, И. С. Сергеев**

Московский государственный университет им. М. В. Ломоносова, механико-математический факультет, 119992 Москва, Ленинские горы
e-mail: gashkov@gmail.com

***НПО КВАНТ, 129626, Москва, 3-я Мытищинская, 16**
e-mail: isserg@gmail.com

В работе рассматривается построение быстрых алгоритмов дискретного преобразования Фурье в некоторых кольцах и их применение к умножению многочленов. В качестве показателя быстродействия алгоритма используется его сложность, определяемая как число выполняемых двухвходовых операций сложения, вычитания, умножения, а также операций умножения на константы кольца.

1. Дискретное преобразование Фурье

Пусть $\mathbf{K}$ — коммутативное кольцо с единицей. Элемент $\zeta \in \mathbf{K}$ называется *примитивным* (*первообразным*) *корнем степени* $N \in \mathbb{N}$, если $\zeta^N = 1$, и никакой из элементов $\zeta^{N/p} - 1$, где p — простой делитель числа N, не является делителем нуля в $\mathbf{K}$. (Напомним, что элемент a называется делителем нуля, если существует ненулевой элемент b, такой, что $ab = 0$.)

Дискретным преобразованием Фурье (*ДПФ*) *порядка* N называется $(\mathbf{K}^N \to \mathbf{K}^N)$-преобразование

$$\text{ДПФ}_{N,\zeta}[\mathbf{K}](\gamma_0, \ldots, \gamma_{N-1}) = (\gamma_0^*, \ldots, \gamma_{N-1}^*), \qquad \gamma_j^* = \sum_{i=0}^{N-1} \gamma_i \zeta^{ij}. \qquad (1)$$

где ζ — примитивный корень степени N.

Фундаментальное свойство ДПФ формулируется следующим образом:

Лемма 1. *Пусть элементы γ_j^* определяются из* (1). *Тогда*

$$\text{ДПФ}_{N,\zeta^{-1}}[\mathbf{K}](\gamma_0^*,\ldots,\gamma_{N-1}^*) = (N\gamma_0,\ldots,N\gamma_{N-1}),$$

где под N в правой части формулы понимается сумма N единиц кольца.

Перед тем, как перейти к доказательству леммы, установим несколько вспомогательных фактов.

Заметим, что если элемент $a \in \mathbf{K}$ не является делителем нуля, и $a = cd$, то множители c и d также не являются делителями нуля. Действительно, если, скажем, $ce = 0$ и $e \neq 0$, то $ae = (ce)d = 0$, откуда следует, что a — делитель нуля.

Лемма 2. *Если ζ — примитивный корень степени N, то при любом $l = 1,\ldots,N-1$*

$$\sum_{i=0}^{N-1} \zeta^{il} = 0.$$

Доказательство. Рассмотрим разложение

$$0 = \zeta^{lN} - 1 = (\zeta^l - 1)\sum_{i=0}^{N-1} \zeta^{il}.$$

Из определения примитивного корня следует, что N — это минимальный натуральный показатель степени n, при котором $\zeta^n = 1$, поэтому $\zeta^l - 1 \neq 0$. Следовательно, либо $\zeta^l - 1$ является делителем нуля, либо $\sum_{i=0}^{N-1} \zeta^{il} = 0$. Покажем, что первое невозможно.

Пусть $m = \text{НОД}(l, N)$. Как известно, существуют целые q, s, такие, что $m = ql + sN$ (числа q, s называются коэффициентами Безу), при этом можно считать, что q — положительно. В таком случае $\zeta^m - 1 = \zeta^{ql} - 1$ делится на $\zeta^l - 1$. С другой стороны, поскольку $m < N$, найдется простое p, такое, что $m \mid (N/p)$. Тогда $(\zeta^m - 1) \mid (\zeta^{N/p} - 1)$. Окончательно, имеем $(\zeta^l - 1) \mid (\zeta^{N/p} - 1)$. Поскольку элемент $\zeta^{N/p} - 1$ не является делителем нуля, то и $\zeta^l - 1$ не может быть делителем нуля. Следовательно, $\sum_{i=0}^{N-1} \zeta^{il} = 0$. Лемма доказана.

Доказательство леммы 1. В векторе $\text{ДПФ}_{N,\zeta^{-1}}[\mathbf{K}](\gamma_0^*,\ldots,\gamma_{N-1}^*)$ рассмотрим произвольную j-ю компоненту:

$$\sum_{i=0}^{N-1} \gamma_i^* \zeta^{-ij} = \sum_{i=0}^{N-1}\sum_{k=0}^{N-1} \gamma_k \zeta^{ki}\zeta^{-ij} = \sum_{k=0}^{N-1}\sum_{i=0}^{N-1} \gamma_k \zeta^{i(k-j)} = \sum_{k=0}^{N-1} \gamma_k \sum_{i=0}^{N-1} (\zeta^{k-j})^i.$$

Внутренняя сумма, как следует из леммы 2, равна нулю во всех случаях, за исключением случая $k - j = 0$, в котором эта сумма равна N. Поэтому, продолжая выкладку, получаем $N\gamma_j$, что и требовалось. Лемма 1 доказана.

Как следствие, получаем, что если элемент $N = 1 + \ldots + 1 \in \mathbf{K}$ обратим, то определено обратное к ДПФ преобразование

$$\text{ДПФ}^{-1}_{N,\zeta}[\mathbf{K}] = N^{-1}\text{ДПФ}_{N,\zeta^{-1}}[\mathbf{K}].$$

2. Полиномиальная интерпретация ДПФ

Рассмотрим многочлен $\Gamma(x) = \gamma_0 + \ldots + \gamma_{N-1}x^{N-1}$. Тогда, по определению,

$$\text{ДПФ}_{N,\zeta}[\mathbf{K}](\gamma_0, \ldots, \gamma_{N-1}) = \left(\Gamma(\zeta^0), \ldots, \Gamma(\zeta^{N-1})\right),$$

т.е. ДПФ вычисляет значения многочлена $\Gamma(x)$ в точках ζ^i. Смысл обратного преобразования $\text{ДПФ}^{-1}_{N,\zeta}[\mathbf{K}]$ заключается в восстановлении коэффициентов единственного многочлена степени, меньшей N, имеющего заданный набор значений в точках $\zeta^0, \ldots, \zeta^{N-1}$.

Формально, связь между ДПФ и интерполяцией описывается следующей леммой:

Лемма 3. *Преобразование* $\text{ДПФ}_{N,\zeta}[\mathbf{K}]$ *задает изоморфизм:* $\mathbf{K}[x]/(x^N - 1) \to \mathbf{K}^N$.

Доказательство. Биективность отображения следует из того, что многочлен степени не выше $N-1$ однозначно определяется своими значениями на наборе из N различных точек.

Проверим, что ДПФ сохраняет операции сложения и умножения: в кольце $\mathbf{K}[x]/(x^N-1)$ эти операции выполняются как с обычными многочленами, только с последующим приведением по модулю $x^N - 1$, в кольце $\mathbf{K}^N$ операции выполняются покомпонентно.

Действительно, значение суммы многочленов $\Gamma_1(x) + \Gamma_2(x)$ в некоторой точке совпадает с суммой значений каждого из многочленов в данной точке. Представляя произведение многочленов в форме $Q(x)(x^N - 1) + R(x)$, где $R(x)$ — остаток от деления на $x^N - 1$, убеждаемся, что произведение переходит в произведение в силу:

$$\Gamma_1(\zeta^j)\Gamma_2(\zeta^j) = Q(\zeta^j)(\zeta^{jN} - 1) + R(\zeta^j) = R(\zeta^j) = (\Gamma_1\Gamma_2 \bmod (x^N - 1))(\zeta^j).$$

Лемма доказана.

Рассмотренный изоморфизм приводит к эффективному способу умножения многочленов над $\mathbf{K}$.

Теорема 1. *Пусть в кольце* $\mathbf{K}$ *определено преобразование* $\text{ДПФ}_{N,\zeta}[\mathbf{K}]$ *и обратное к нему. Тогда умножение двух многочленов суммарной степени не выше* $N-1$ *над* $\mathbf{K}$ *можно выполнить при помощи двух преобразований* $\text{ДПФ}_{N,\zeta}[\mathbf{K}]$, *одного* $\text{ДПФ}^{-1}_{N,\zeta}[\mathbf{K}]$ *и* N *умножений в* $\mathbf{K}$.

Доказательство. Обозначим перемножаемые многочлены через $A(x) = \sum a_i x^i$ и $B(x) = \sum b_i x^i$. Вычислим вектора

$$(a_0^*, \ldots, a_{N-1}^*) = \text{ДПФ}_{N,\zeta}[\mathbf{K}](a_0, \ldots, a_{N-1}),$$

$$(b_0^*, \ldots, b_{N-1}^*) = \text{ДПФ}_{N,\zeta}[\mathbf{K}](b_0, \ldots, b_{N-1}).$$

Затем коэффициенты многочлена $C(x) = \sum c_i x^i = A(x)B(x)$ в силу $C(x) = C(x) \bmod (x^N - 1)$ могут быть найдены как

$$(c_0, \ldots, c_{N-1}) = \text{ДПФ}^{-1}_{N,\zeta}[\mathbf{K}](a_0^* b_0^*, \ldots, a_{N-1}^* b_{N-1}^*),$$

откуда следует утверждение теоремы. Теорема доказана.

3. Вычисление ДПФ

Независимое вычисление компонент вектора ДПФ по формулам (1) может быть выполнено за $O(N^2)$ операций сложения, вычитания и умножения на константы в кольце $\mathbf{K}$. Для составного числа N можно предложить следующий более эффективный способ.

Прежде заметим, что если ζ — примитивный корень степени ST, то ζ^S и ζ^T — примитивные корни степени T и S соответственно (это легко проверить непосредственно из определения).

Справедлива

Лемма 4 (Кули, Тьюки [5]). *ДПФ порядка* ST *реализуется при помощи* S *ДПФ порядка* T, T *ДПФ порядка* S *и* $(S-1)(T-1)$ *операций умножения на степени* ζ *— примитивного корня степени* ST.

Доказательство. Для $s = 0, \ldots, S-1$ и $t = 0, \ldots, T-1$ запишем

$$\gamma^*_{sT+t} = \sum_{I=0}^{ST-1} \gamma_I \zeta^{I(sT+t)} = \sum_{i=0}^{T-1} \sum_{j=0}^{S-1} \gamma_{iS+j} \zeta^{(iS+j)(sT+t)} =$$

$$= \sum_{i=0}^{T-1} \sum_{j=0}^{S-1} \gamma_{iS+j} \zeta^{itS+jsT+jt} = \sum_{j=0}^{S-1} (\zeta^T)^{js} \cdot \zeta^{jt} \cdot \gamma_{(j),t}, \quad (2)$$

где

$$\gamma_{(j),t} = \sum_{i=0}^{T-1} \gamma_{iS+j} (\zeta^S)^{it}.$$

Полученная формула позволяет произвести вычисления в следующем порядке:

а) Для $j = 0, \ldots, S-1$ вычисляются вектора

$$\left(\gamma_{(j),0}, \gamma_{(j),1}, \ldots, \gamma_{(j),T-1}\right) = \text{ДПФ}_{T,\zeta^S}[\mathbf{K}](\gamma_j, \gamma_{S+j}, \ldots, \gamma_{(T-1)S+j}).$$

б) Вычисляются произведения $\omega_{(t),j} = \zeta^{jt} \cdot \gamma_{(j),t}$, $j = 0, \ldots, S-1$, $t = 0, \ldots, T-1$.

в) Заметим, что

$$\gamma^*_{sT+t} = \sum_{j=0}^{S-1} \omega_{(t),j} (\zeta^T)^{js}.$$

Это позволяет окончательно найти компоненты вектора ДПФ по формулам

$$\left(\gamma^*_t, \gamma^*_{T+t}, \ldots, \gamma^*_{(S-1)T+t}\right) = \text{ДПФ}_{S,\zeta^T}[\mathbf{K}](\omega_{(t),0}, \omega_{(t),1}, \ldots, \omega_{(t),S-1}),$$

где $t = 0, \ldots, T-1$.

Утверждение леммы немедленно следует из вида действий, выполняемых на шагах *а*–*в*), если заметить, что среди ST умножений на шаге *б*) есть $S + T - 1$ умножений на $\zeta^0 = 1$. Лемма доказана.

Замечание (Гуд, Томас [6]). *Если числа S и T взаимно просты, то для вычисления ДПФ порядка ST достаточно выполнить S ДПФ порядка T и T ДПФ порядка S, т.е. дополнительных умножений не требуется.*

Обозначим через $F(N) = F_A(N) + F_C(N)$ сложность схемы ДПФ порядка N, построенной методом леммы 4, где $F_A(N)$ — число аддитивных элементов (сложений и вычитаний) в схеме, а $F_C(N)$ — число скалярных умножений (т.е. умножений на константы кольца $\mathbf{K}$). При этом необходимые схемы ДПФ простых порядков должны быть построены отдельно.

Несложно построить схему для ДПФ порядка 2, если оно существует. Компоненты ДПФ определяются формулами

$$\gamma^*_0 = \gamma_0 + \gamma_1, \qquad \gamma^*_1 = \gamma_0 - \gamma_1,$$

т.к. в качестве примитивного корня степени 2 можно взять -1. Поэтому положим $F(2) = 2$, $F_A(2) = 2$, $F_C(2) = 0$.

Для вычисления сложности схемы ДПФ порядка 2^k, где $k > 1$, воспользуемся рекуррентными соотношениями, вытекающими из леммы 4 при значениях параметров $S = 2^{k-1}$ и $T = 2$:

$$F_A(2^k) = 2F_A(2^{k-1}) + 2^{k-1}F_A(2), \quad F_C(2^k) = 2F_C(2^{k-1}) + 2^{k-1}F_C(2) + 2^{k-1} - 1.$$

Легко проверить, что указанные соотношения разрешаются как

$$F_A(2^k) = k2^k, \qquad F_C(2^k) = (k-2)2^{k-1} + 1. \tag{3}$$

Доказана

Теорема 2. *ДПФ порядка 2^k может быть выполнено за $k2^k$ операций сложения-вычитания и $(k-2)2^{k-1}+1$ операций скалярного умножения.*

Эта оценка является асимптотически наилучшей из известных верхних оценок сложности ДПФ порядка 2^k.

Сложность схемы обратного ДПФ порядка 2^k с точностью до 2^k умножений на константы 2^{-k} совпадает со сложностью «прямого» ДПФ, причем при $k \geqslant 2$ умножения на 2^{-k} можно совместить с умножениями на шаге *б*).

Приведенный алгоритм является примером алгоритма *быстрого преобразования Фурье* (*БПФ*). Вообще, под алгоритмами БПФ понимаются такие алгоритмы, в которых используется прием сведения ДПФ составного порядка N к ДПФ порядка множителей числа N. Иногда термин БПФ прилагается к любым алгоритмам сложности $O(N \log N)$.

4. Вещественная сложность комплексного ДПФ

Ситуация, когда аддитивные и мультипликативные операции в кольце $\mathbf{K}$ нельзя считать равноценными, приводит к возникновению специальных алгоритмов БПФ. Проиллюстрируем это на примере поля комплексных чисел $\mathbb{C}$.

Комплексное число обычно представляется парой вещественных чисел — действительной и мнимой частью: $z = x + \mathbf{i}y$. Пересчитаем операции с комплексными числами на операции с вещественными. Комплексное сложение (вычитание) равноценно двум вещественным сложениям (вычитаниям). Комплексное умножение можно выполнить, используя четыре вещественных умножения и два сложения-вычитания. Кроме того, для умножения на константу достаточно трех вещественных умножений и трех сложений-вычитаний.

Для описанной выше схемы комплексного ДПФ порядка 2^k получаем оценки $F_A^{\mathbb{R}}(2^k) < 3,5k2^k$ для числа вещественных сложений-вычитаний и $F_C^{\mathbb{R}}(2^k) < 1,5k2^k$ — для числа вещественных скалярных умножений и суммарно $F^{\mathbb{R}}(2^k) < 5k2^k$.

Можно, однако, получить лучшую оценку, если заметить, что некоторые умножения в алгоритме леммы 4 выгодно не выполнять сразу, а перенести на следующий этап вычислений (реализация ДПФ порядка S). На этом наблюдении основан известный алгоритм БПФ «с расщепленным основанием».

Теорема 3. *ДПФ порядка 2^k над полем $\mathbb{C}$ можно реализовать, используя не более $3k2^k$ сложений-вычитаний и не более $k2^k$ умножений в поле $\mathbb{R}$.*

Доказательство. Вычислим по формуле (2), в которой $S = 2^{k-1}$ и $T = 2$, только компоненты ДПФ порядка 2^k с четными индексами, т.е. при $t = 0$. Для этого достаточно вычислить по одной компоненте $\gamma_{(j),0} = \gamma_j + \gamma_{S+j}$ каждого из 2^{k-1} внутренних ДПФ порядка 2 и реализовать внешнее ДПФ порядка 2^{k-1}.

Для вычисления компонент с нечетными индексами положим $S = 2^{k-2}$ и $T = 4$ и вновь воспользуемся формулой (2). У каждого из 2^{k-2} внутренних ДПФ порядка 4 требуется вычислить по две компоненты $\gamma_{(j),1}$ и $\gamma_{(j),3}$. Каждая такая пара в силу $\zeta^S = \mathbf{i}$ может быть вычислена по формулам

$$\gamma_{(j),1} = (\gamma_j - \gamma_{2S+j}) + \mathbf{i}(\gamma_{S+j} - \gamma_{3S+j}),$$

$$\gamma_{(j),3} = (\gamma_j - \gamma_{2S+j}) - \mathbf{i}(\gamma_{S+j} - \gamma_{3S+j}).$$

Поскольку умножение на $\pm\mathbf{i}$ сводится к перестановке действительной и мнимой части со сменой знака у одной из них, вычисление одной пары $\gamma_{(j),1}$, $\gamma_{(j),3}$ по этим формулам выполняется за 8 вещественных сложений-вычитаний. Окончательно, выполняется 2^{k-1} умножений на степени ζ^{jt} и два ДПФ порядка 2^{k-2}.

Для числа сложений-вычитаний $F_A^{\mathbb{R}}(2^k)$ и числа скалярных умножений $F_C^{\mathbb{R}}(2^k)$ в построенной схеме имеем рекуррентные соотношения:

$$F_A^{\mathbb{R}}(2^k) \leqslant F_A^{\mathbb{R}}(2^{k-1}) + 2F_A^{\mathbb{R}}(2^{k-2}) + 4,5 \cdot 2^k,$$

$$F_C^{\mathbb{R}}(2^k) \leqslant F_C^{\mathbb{R}}(2^{k-1}) + 2F_C^{\mathbb{R}}(2^{k-2}) + 1,5 \cdot 2^k,$$

которые с учетом начальных данных

$$F_A^{\mathbb{R}}(2) = 4,\ F_C^{\mathbb{R}}(2) = 0,\ F_A^{\mathbb{R}}(4) = 16,\ F_C^{\mathbb{R}}(4) = 0,$$

разрешаются так, как заявлено в утверждении теоремы. Теорема доказана.

Более аккуратный учет позволяет получить оценки сложности метода в виде: $3k2^k - 3\cdot 2^k + 4$ сложений-вычитаний и $k2^k - 3\cdot 2^k + 4$ скалярных умножений над $\mathbb{R}$. Эта оценка была получена не позднее 1968 г., но только в 2004 г. ван Бускирк обнаружил, что она может быть улучшена (см. обзор [3]). Его метод учитывает, 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.

Теорема 4. *ДПФ порядка 2^k над полем $\mathbb{C}$ можно реализовать, используя не более $(8/3)k2^k$ сложений-вычитаний и не более $(10/9)k2^k + 2^{k+1}$ умножений в поле $\mathbb{R}$.*

Доказательство. При $k \in \mathbb{N}$ и $j \in \mathbb{Z}$ определим вещественные коэффициенты

$$\sigma_{k,j} = \prod_{l\geqslant 0} \max\left\{\left|\cos\frac{4^l 2\pi j}{2^k}\right|, \left|\sin\frac{4^l 2\pi j}{2^k}\right|\right\}.$$

Эти коэффициенты обладают свойствами симметрии $\sigma_{k,j} = \sigma_{k,-j}$ и периодичности $\sigma_{k,j} = \sigma_{k,j+2^{k-2}}$, что вытекает из известных соотношений

$$\sin x = -\sin(-x), \quad \cos x = \cos(-x),$$

$$\left\{\left|\sin\left(x + \frac{\pi n}{2}\right)\right|, \left|\cos\left(x + \frac{\pi n}{2}\right)\right|\right\} = \{|\sin x|, |\cos x|\},$$

где $x \in \mathbb{R}$, $n \in \mathbb{Z}$. Кроме того, для примитивного корня $\zeta = e^{\frac{2\pi\mathbf{i}}{2^k}}$ степени 2^k справедливо: $(\sigma_{k-2,j}/\sigma_{k,j})\zeta^j$ имеет вид $\pm 1 + a\mathbf{i}$ или $a \pm \mathbf{i}$, поскольку

$$\zeta^j = \cos\frac{2\pi j}{2^k} + \mathbf{i}\sin\frac{2\pi j}{2^k}, \qquad \frac{\sigma_{k,j}}{\sigma_{k-2,j}} = \max\left\{\left|\cos\frac{2\pi j}{2^k}\right|, \left|\sin\frac{2\pi j}{2^k}\right|\right\}.$$

Будем строить схемы для преобразований

$$\Phi_{2^k}(\gamma_0, \ldots, \gamma_{2^k-1}) = \text{ДПФ}_{2^k,\zeta}[\mathbb{C}](\sigma_{k,0}^{-1}\gamma_0, \sigma_{k,1}^{-1}\gamma_1, \ldots, \sigma_{k,2^k-1}^{-1}\gamma_{2^k-1}).$$

Согласно формуле (2) с выбором параметров $S = 2^{k-2}$ и $T = 4$ и свойству периодичности коэффициентов $\sigma_{k,j}$ для компонент φ_i преобразования Φ_{2^k} выполнено:

$$\varphi_{4s+t} = \sum_{j=0}^{S-1} (\zeta^4)^{js}\cdot\zeta^{jt}\cdot\sigma_{k,j}^{-1}\gamma_{(j),t}, \qquad \gamma_{(j),t} = \sum_{l=0}^{3}\gamma_{lS+j}\cdot\mathbf{i}^{lt}.$$

Компоненты $\gamma_{(j),t}$, $t = 0, \ldots, 3$, каждого из 2^{k-2} ДПФ порядка 4 можно вычислить, используя 16 вещественных сложений-вычитаний. Последующие вычисления при $t = 0,\ 1,\ 3$ выполняются по формулам

$$\varphi_{4s} = \sum_{j=0}^{S-1} (\zeta^4)^{js} \sigma_{k-2,j}^{-1} \cdot (\sigma_{k-2,j}/\sigma_{k,j}) \cdot \gamma_{(j),0},$$

$$\varphi_{4s+1} = \sum_{j=0}^{S-1} (\zeta^4)^{js} \sigma_{k-2,j}^{-1} \cdot (\sigma_{k-2,j}/\sigma_{k,j}) \zeta^j \cdot \gamma_{(j),1},$$

$$\varphi_{4s+3} = \sum_{j=0}^{S-1} (\zeta^4)^{j(s+1)} \sigma_{k-2,j}^{-1} \cdot (\sigma_{k-2,j}/\sigma_{k,j}) \zeta^{-j} \cdot \gamma_{(j),3}.$$

Эти вычисления состоят в выполнении 2^{k-2} умножений на действительные константы, 2^{k-1} умножений на константы вида $\pm 1 + a\mathbf{i}$ или $a \pm \mathbf{i}$ и трех преобразований $\Phi_{2^{k-2}}$.

Для вычисления оставшихся компонент φ_{4s+2} используем формулу (2) с параметрами $S' = 2^{k-3}$ и $T' = 8$:

$$\varphi_{8s+2} = \sum_{j=0}^{S'-1} (\zeta^8)^{js} \sigma_{k-3,j}^{-1} \cdot (\sigma_{k-3,j}/\sigma_{k-1,j}) (\zeta^2)^j \cdot \gamma'_{(j),2},$$

$$\varphi_{8s+6} = \sum_{j=0}^{S'-1} (\zeta^8)^{j(s+1)} \sigma_{k-3,j}^{-1} \cdot (\sigma_{k-3,j}/\sigma_{k-1,j}) (\zeta^2)^{-j} \cdot \gamma'_{(j),6},$$

где

$$\gamma'_{(j),2} = (\sigma_{k-1,j}/\sigma_{k,j}) \gamma_{(j),2} + (\sigma_{k-1,j+S'}/\sigma_{k,j+S'}) \mathbf{i} \gamma_{(j+S'),2},$$

$$\gamma'_{(j),6} = (\sigma_{k-1,j}/\sigma_{k,j}) \gamma_{(j),2} - (\sigma_{k-1,j+S'}/\sigma_{k,j+S'}) \mathbf{i} \gamma_{(j+S'),2}.$$

Заметим, что $\sigma_{k-1,j} = \sigma_{k-1,j+S'}$. Эти вычисления выполняются при помощи 2^{k-2} умножений на действительные или мнимые константы, 2^{k-2} умножений на константы вида $\pm 1 + a\mathbf{i}$ или $a \pm \mathbf{i}$ и двух преобразований $\Phi_{2^{k-3}}$.

Итого, для чисел $\hat{F}_A^{\mathbb{R}}(2^k)$ аддитивных вещественных операций и $\hat{F}_C^{\mathbb{R}}(2^k)$ вещественных скалярных умножений имеем рекуррентные соотношения:

$$\hat{F}_A^{\mathbb{R}}(2^k) \leqslant 3\hat{F}_A^{\mathbb{R}}(2^{k-2}) + 2\hat{F}_A^{\mathbb{R}}(2^{k-3}) + 6 \cdot 2^k,$$

$$\hat{F}_C^{\mathbb{R}}(2^k) \leqslant 3\hat{F}_C^{\mathbb{R}}(2^{k-2}) + 2\hat{F}_C^{\mathbb{R}}(2^{k-3}) + 2,5 \cdot 2^k,$$

которые в согласии с начальными данными для $k \leqslant 3$, полученными предыдущим методом, разрешаются как

$$\hat{F}_A^{\mathbb{R}}(2^k) \leqslant (8/3)k2^k, \qquad \hat{F}_C^{\mathbb{R}}(2^k) \leqslant (10/9)k2^k.$$

Осталось учесть, что схема для ДПФ порядка 2^k достраивается из схемы для Φ_{2^k} при помощи 2^k умножений на вещественные константы $\sigma_{k,j}$. Теорема доказана.

Более аккуратный подсчет операций позволяет уточнить оценки теоремы 4 в остаточном члене (см., например, [2]).

Сложность обратного ДПФ (с точностью до умножений на 2^{-k}, которые могут быть совмещены со внутренними умножениями) в обоих рассмотренных алгоритмах оценивается так же, как и сложность «прямого», поскольку примитивный корень ζ^{-1} является комплексно-сопряженным к ζ числом.

Рассмотрим случай, когда ДПФ применяется к вектору с действительными компонентами γ_j — этот случай представляет интерес при умножении многочленов над $\mathbb{R}$. Оказывается, сложность ДПФ (будем называть его действительным ДПФ) в данном случае может быть понижена примерно вдвое по сравнению с общим случаем.

Заметим, что если все $\gamma_j \in \mathbb{R}$, то $\gamma^*_{N-j} = \overline{\gamma^*_j}$ для любого j, где γ^*_j определяются из (1), а $\overline{\ }$ обозначает операцию комплексного сопряжения.

Лемма 5. *Действительное ДПФ порядка $4N$ может быть реализовано при помощи действительного ДПФ порядка $2N$, комплексного ДПФ порядка N, $7N$ операций сложения-вычитания и $3N$ операций скалярного умножения в $\mathbb{R}$.*

Доказательство. Для определения компонент γ^*_k с четными индексами $k = 2s$ воспользуемся формулой (2), полагая $S = 2N$ и $T = 2$:

$$\gamma^*_{2s} = \sum_{j=0}^{2N-1} (\zeta^2)^{js} \cdot (\gamma_j + \gamma_{2N+j}).$$

Эти вычисления сводятся к реализации ДПФ порядка $2N$ с вещественными аргументами и $2N$ сложениям в $\mathbb{R}$.

Среди компонент с нечетными индексами достаточно вычислить только γ^*_{4s+1}, т.к. $\gamma^*_{4s+3} = \overline{\gamma^*_{4(N-s-1)+1}}$. Для этого применяется (2) с параметрами $S = N$ и $T = 4$:

$$\gamma^*_{4s+1} = \sum_{j=0}^{N-1} (\zeta^4)^{js} \cdot \zeta^j \cdot (\gamma_j - \gamma_{2N+j} + \mathbf{i}(\gamma_{N+j} - \gamma_{3N+j})).$$

Для вычисления указанных компонент достаточно одного ДПФ порядка N, не более $2N$ вычитаний в $\mathbb{R}$ и N скалярных умножений в $\mathbb{C}$. Лемма доказана.

Другой способ сведения действительного ДПФ к комплексному ДПФ вдвое меньшего порядка можно посмотреть в [1].

Аналогичное утверждение можно доказать про сложность обратного к действительному ДПФ преобразования — его входом является вектор $(\gamma_0, \ldots, \gamma_{N-1})$ такой, что $\gamma_0 \in \mathbb{R}$ и $\gamma_{N-j} = \overline{\gamma_j}$ для любого $j = 1, \ldots, N-1$. Такое преобразование назовем действительнозначным ДПФ. В отношении этого ДПФ справедлив результат, аналогичный доказанному выше.

Лемма 6. *Действительнозначное ДПФ порядка $4N$ может быть реализовано при помощи действительнозначного ДПФ порядка $2N$, комплексного ДПФ порядка N, $7N$ операций сложения-вычитания и $3N$ операций скалярного умножения в $\mathbb{R}$.*

Доказательство. Используя формулу (2) и обозначения из леммы 4 с выбором параметров $S = 2$ и $T = 2N$, можно записать

$$\gamma^*_{2Ns+t} = \omega_{(t),0} + (-1)^s \omega_{(t),1} = \gamma_{(0),t} + (-1)^s \zeta^t \gamma_{(1),t},$$

где $s = 0$, 1 и $t = 0, \ldots, 2N-1$,

$$\gamma_{(0),t} = \sum_{i=0}^{2N-1} \gamma_{2i} (\zeta^2)^{it}, \qquad \gamma_{(1),t} = \sum_{i=0}^{2N-1} \gamma_{2i+1} (\zeta^2)^{it}.$$

При условии, что компоненты $\omega_{(t),i}$ вычислены, γ^*_j определяются за $4N$ вещественных сложений-вычитаний.

Вектор с компонентами $\omega_{(t),0} = \gamma_{(0),t}$ является образом действительнозначного ДПФ порядка $2N$, т.к. оно применяется к вектору $(\gamma_0, \gamma_2, \ldots, \gamma_{2(2N-1)})$. Для вычисления $\omega_{(t),1}$ представим $\gamma_{(1),t}$ в форме (2) с выбором параметров $S = 2$ и $T = N$:

$$\gamma_{(1),Ns'+t'} = \gamma'_{(0),t'} + (-1)^{s'} \zeta^{2t'} \gamma'_{(1),t'},$$

где $s' = 0$, 1 и $t' = 0, \ldots, N-1$,

$$\gamma'_{(0),t'} = \sum_{i=0}^{N-1} \gamma_{4i+1} (\zeta^4)^{it'}, \qquad \gamma'_{(1),t'} = \sum_{i=0}^{N-1} \gamma_{4i+3} (\zeta^4)^{it'}.$$

В силу $\gamma_{4N-j} = \overline{\gamma_j}$ справедливо:

$$\gamma'_{(1),t'} = \sum_{i=0}^{N-1} \overline{\gamma_{4(N-i-1)+1}}(\zeta^4)^{it'} = \sum_{i'=0}^{N-1} \overline{\gamma_{4i'+1}}(\zeta^4)^{(N-1-i')t'} =$$

$$= \zeta^{-4t'} \overline{\sum_{i'=0}^{N-1} \gamma_{4i'+1}(\zeta^4)^{i't'}} = \zeta^{-4t'}\overline{\gamma'_{(0),t'}}.$$

Таким образом, компоненты $\omega_{(t),1}$ могут быть определены по формулам:

$$\omega_{(Ns'+t'),1} = \mathbf{i}^{s'}(\zeta^{t'}\gamma'_{(0),t'} + (-1)^{s'}\zeta^{-t'}\overline{\gamma'_{(0),t'}}),$$

т.е.

$$\omega_{(t'),1} = \mathrm{Re}(2\zeta^{t'}\gamma'_{(0),t'}), \qquad \omega_{(N+t'),1} = -\mathrm{Im}(2\zeta^{t'}\gamma'_{(0),t'}).$$

Для вычисления всех $\omega_{(t),1}$ достаточно одного ДПФ порядка N и N операций умножения в $\mathbb{C}$. Лемма доказана.

Разрешая рекуррентные соотношения, вытекающие из доказанных лемм, с использованием теоремы 4 заключаем:

Следствие 1. *Как действительное, так и действительнозначное ДПФ порядка 2^k можно выполнить за $(4/3)k2^k + O(2^k)$ операций сложения-вычитания и $(5/9)k2^k + O(2^k)$ операций скалярного умножения в $\mathbb{R}$.*

5. ДПФ в кольце-расширении

Если кольцо $\mathbf{K}$ не содержит корней из единицы подходящей степени, то для умножения многочленов над $\mathbf{K}$ непосредственно использовать способ теоремы 1 невозможно. В алгоритме Шёнхаге—Штрассена [7, 8] и ему подобных в таком случае предлагается использовать расширение $\mathbf{K}_{2,n}(x) = \mathbf{K}[x]/(x^{2^n}+1)$, при этом двойка должна быть обратима в $\mathbf{K}$.

В кольце $\mathbf{K}_{2,n}(x)$ определено ДПФ порядка 2^{n+1} с примитивным корнем x (здесь и далее вместо элементов фактор-кольца $\mathbf{K}_{2,n}(x)$, которыми являются классы эквивалентных по модулю $x^{2^n}+1$ многочленов, будут фигурировать многочлены-представители классов).

Лемма 7. *ДПФ порядка 2^k над кольцом $\mathbf{K}_{2,n}(x)$, $k \leqslant n+1$, может быть выполнено за $k2^{k+n}$ операций сложения-вычитания в $\mathbf{K}$.*

Доказательство. Представляя элементы кольца $\mathbf{K}_{2,n}(x)$ многочленами степени не выше $2^n - 1$, легко видеть, что сложение или вычитание в $\mathbf{K}_{2,n}(x)$ соответствует 2^n сложениям-вычитаниям в кольце $\mathbf{K}$, а умножение на x^m — к циклическому сдвигу коэффициентов со сменой знака у некоторых из них. Таким образом, если смена знака может быть учтена в последующих вычислениях, умножение на степени примитивного корня x реализуется «бесплатно». Для завершения доказательства теперь остается воспользоваться оценками (3). Лемма доказана.

Для реализации умножения в $\mathbf{K}_{2,n}(x)$ это кольцо удобно рассмотреть как расширение некоторого кольца $\mathbf{K}_{2,m}(y)$: справедлива

Лемма 8. *Пусть $m < n$. Имеет место изоморфизм*

$$\mathbf{K}_{2,n}(x) \cong \mathbf{K}_{2,m}(y)[x]/(x^{2^{n-m}} - y), \tag{4}$$

порождаемый подстановкой $x^{2^{n-m}} = y$.

Доказательство. Многочлен $f(x) \in \mathbf{K}_{2,n}(x)$ можно записать в виде $f(x) = \sum_{i=0}^{2^{n-m}-1} f_i(x^{2^{n-m}})x^i$, где $\deg f_i < 2^m$. Подстановка $x^{2^{n-m}} = y$ переводит $f(x)$ в многочлен $\sum_{i=0}^{2^{n-m}-1} f_i(y)x^i$. Положим $f_i(y) \in \mathbf{K}_{2,m}(y)$.

Очевидно, что подстановка порождает линейное взаимно однозначное отображение. Остается проверить, что это отображение сохраняет произведение, причем в силу линейности проверку можно ограничить нормированными одночленами. В кольце $\mathbf{K}_{2,n}(x)$ выполнено

$$x^{j_1 2^{n-m}+i_1} \cdot x^{j_2 2^{n-m}+i_2} = x^{j_3 2^{n-m}+i_3} = (-1)^k x^{j_4 2^{n-m}+i_3},$$

где

$$i_3 = (i_1 + i_2) \bmod 2^{n-m}, \qquad j_3 = j_1 + j_2 + (i_1 + i_2 - i_3)/2^{n-m},$$

$$j_4 = j_3 \bmod 2^m, \qquad k = (j_3 - j_4)/2^m.$$

С другой стороны, в кольце $\mathbf{K}_{2,m}(y)[x]/(x^{2^{n-m}} - y)$ также верно

$$y^{j_1} x^{i_1} \cdot y^{j_2} x^{i_2} = y^{j_3} x^{i_3} = (-1)^k y^{j_4} x^{i_3}.$$

Видно, что результаты обоих умножений переходят друг в друга при подстановке $y = x^{2^{n-m}}$. Лемма доказана.

Важно заметить, что рассматриваемое отображение реализуется простой перестановкой коэффициентов. Например, многочлену $x^3 + 2x^2 - 1 \in \mathbf{K}_{2,2}(x)$ соответствует многочлен $yx + (2y - 1) \in \mathbf{K}_{2,1}(y)[x]/(x^2 - y)$. Для реализации умножения можно использовать и другие изоморфизмы, см. [3].

Теорема 5. *Умножение в кольце* $\mathbf{K}_{2,n}(x)$ *может быть выполнено при помощи* $3 \cdot 2^n n(\log_2 n + O(1))$ *операций сложения-вычитания,* $3 \cdot 2^{n+\lceil \log_2 n \rceil - 1}$ *операций умножения и* 2^n *операций скалярного умножения в* $\mathbf{K}$.

Доказательство. Воспользуемся (4) с выбором параметра $m = \lceil n/2 \rceil$. Умножение многочленов над $\mathbf{K}_{2,m}(y)$ по модулю $x^{2^{n-m}} - y$ выполним как обычное умножение многочленов степени не выше $2^{n-m} - 1$ с последующим приведением по модулю.

Умножение выполняется при помощи трех ДПФ порядка $2^{n-m+1} = 2^{\lfloor n/2 \rfloor + 1}$ и $2^{\lfloor n/2 \rfloor + 1}$ умножений в кольце $\mathbf{K}_{2,m}(y)$, при этом обратное ДПФ реализуется с точностью до постоянного множителя. Приведение по модулю $x^{2^{n-m}} - y$ реализуется за 2^n сложений-вычитаний в $\mathbf{K}$. Окончательно результат умножается на подходящую степень 2^{-1}.

Для чисел $\mu_A(n)$ сложений-вычитаний и $\mu_M(n)$ нескалярных умножений в предложенной схеме при $n \geqslant 2$ имеем рекуррентные соотношения:

$$\mu_A(n) \leqslant 2^{\lfloor n/2 \rfloor + 1} \mu_A(\lceil n/2 \rceil) + 3(\lfloor n/2 \rfloor + 1)2^{n+1} + 2^n,$$

$$\mu_M(n) \leqslant 2^{\lfloor n/2 \rfloor + 1} \mu_M(\lceil n/2 \rceil),$$

которые разрешаются так, как заявлено в утверждении теоремы, если при $n = 1$ воспользоваться оценками $\mu_A(1) = 5$ и $\mu_M(1) = 3$. Иначе, можно положить $\mu_A(1) = 2$ и $\mu_M(1) = 4$ — в этом случае схема будет содержать $2^{n+\lceil \log_2 n \rceil + 1}$ умножений, но общее число операций будет несколько меньше. Теорема доказана.

Доказанная оценка сложности является асимптотически наилучшей из известных. Умножение многочленов над $\mathbf{K}$ теперь можно выполнить при помощи подходящей схемы умножения в $\mathbf{K}_{2,n}(x)$.

6. Применение ДПФ порядка 3^k

В кольце характеристики 2 нельзя определить ДПФ четного порядка, поэтому актуальной является задача построения и эффективной реализации (в самом кольце или в расширении) ДПФ нечетного порядка, предпочтительно порядка 3^k. Эта задача также является актуальной для колец, в которых есть примитивные корни степени 3^k, либо двойка необратима.

Компоненты ДПФ порядка 3 вычисляются по формулам

$$\gamma_0^* = \gamma_0 + \gamma_1 + \gamma_2, \quad \gamma_1^* = \gamma_0 - \gamma_2 + \zeta(\gamma_1 - \gamma_2), \quad \gamma_2^* = \gamma_0 - \gamma_1 - \zeta(\gamma_1 - \gamma_2), \quad (5)$$

где ζ — примитивный корень степени 3 в $\mathbf{K}$. Эти вычисления можно выполнить, используя семь операций сложения-вычитания и одно скалярное умножение (или шесть сложений-вычитаний и два умножения). Если $\operatorname{char} \mathbf{K} = 2$, то достаточно пяти сложений и одного умножения.

Из леммы 4 следует

Теорема 6. *ДПФ порядка 3^k можно реализовать, используя не более $(7/3)k3^k$ операций сложения-вычитания и $(k-1)3^k + 1$ операций скалярного умножения. В кольце характеристики 2 число аддитивных операций оценивается как $(5/3)k3^k$.*

Доказательство. Указанные оценки следуют из рекуррентных соотношений на числа $F_A(3^k)$ аддитивных операций и $F_C(3^k)$ операций скалярного умножения в методе леммы 4:

$$F_A(3^k) = 3F_A(3^{k-1}) + 3^{k-1}F_A(3),$$

$$F_C(3^k) = 3F_C(3^{k-1}) + 3^{k-1}F_C(3) + 2 \cdot 3^{k-1} - 2$$

и начальных условий: $F_C(3) = 1$, $F_A(3) = 7$ в общем случае или $F_A(3) = 5$ для кольца характеристики 2. Теорема доказана.

Если в кольце $\mathbf{K}$ нет примитивных корней достаточно большой степени 3^k, но тройка обратима, то можно рассмотреть расширение $\mathbf{K}_{3,n}(x) = \mathbf{K}[x]/(x^{2\cdot 3^n} + x^{3^n} + 1)$, в котором x является примитивным корнем степени 3^{n+1}.

В кольце $\mathbf{K}_{3,n}(x)$ сложение (вычитание) выполняется за $2 \cdot 3^n$ операций сложения (вычитания) в $\mathbf{K}$, а сложность умножения на x^m зависит от m: справедлива

Лемма 9. *Сложность умножения на x^m с точностью до множителя ± 1 в кольце $\mathbf{K}_{3,n}(x)$ составляет $|m|$ операций вычитания в $\mathbf{K}$, если $-3^n \leqslant m \leqslant 3^n$, и 3^n операций вычитания, иначе (т.е. если $3^n < m < 2 \cdot 3^n$).*

Доказательство. Пусть $0 \leqslant m \leqslant 3^n$. Запишем многочлен $f(x) \in \mathbf{K}_{3,n}(x)$ в виде $a(x)x^{2\cdot 3^n - m} + b(x)$, где $\deg a < m$ и $\deg b < 2 \cdot 3^n - m$. В кольце $\mathbf{K}_{3,n}(x)$ выполняется равенство

$$f(x)x^m = b(x)x^m - a(x) - a(x)x^{3^n},$$

из которого видно, что, не считая умножений на -1, вычисление коэффициентов произведения требует m вычитаний в $\mathbf{K}$.

В случае $-3^n \leqslant m < 0$ запишем $f(x) = b(x)x^{-m} + a(x)$. Тогда в силу

$$f(x)x^m = b(x) - a(x)x^{2\cdot 3^n} - a(x)x^{3^n}$$

для вычисления коэффициентов многочлена также требуется $|m|$ умножений.

Пусть $3^n < m < 2 \cdot 3^n$). Представим $f(x)$ в виде $a(x) + b(x)x^{2\cdot 3^n - m} + c(x)x^{-m}$, где $\deg a < 2 \cdot 3^n - m$, $\deg b < 3^n$ и $\deg c < m - 3^n$. Тогда для вычисления $f(x)x^m$ достаточно 3^n вычитаний, поскольку

$$f(x)x^m = a(x)x^m - b(x)x^{3^n} + c(x) - b(x).$$

Лемма доказана.

Лемма 10. *Любое из преобразований* $\text{ДПФ}_{3,\zeta}[\mathbf{K}_{3,n}(x)](\gamma_0, \zeta^{c_1}\gamma_1, \zeta^{c_2}\gamma_2)$, *где* $\zeta = x^{3^n}$, $c_1, c_2 \in \{0, 1, 2\}$, *может быть выполнено за* $13 \cdot 3^n$ *операций сложения-вычитания в* $\mathbf{K}$ *или за* $10 \cdot 3^n$ *операций сложения, если* $\operatorname{char} \mathbf{K} = 2$.

Доказательство. Несмотря на девять возможностей для выбора параметров c_1, c_2, в действительности достаточно рассмотреть три случая, например, $(c_1, c_2) \in \{(0, 0), (0, 1), (1, 1)\}$. Компоненты любого другого преобразования получаются перестановкой компонент одного из трех перечисленных.

Рассмотрим случай $c_1 = c_2 = 0$, в котором изучаемое преобразование является обычным ДПФ порядка 3. Запишем $\gamma_i \in \mathbf{K}_{3,n}(x)$ в виде $a_i(x) + x^{3^n} b_i(x)$, где a_i, b_i — многочлены степени меньше 3^n, $i = 0, 1, 2$. Тогда формулы (5) можно переписать как

$$\gamma_0^* = (a_0 + a_1 + a_2) + x^{3^n}(b_0 + b_1 + b_2),$$

$$\gamma_1^* = (a_0 - a_2 - (b_1 - b_2)) + x^{3^n}(b_0 - b_1 + a_1 - a_2),$$

$$\gamma_2^* = (a_0 - a_1 + b_1 - b_2) + x^{3^n}(b_0 - b_2 - (a_1 - a_2)).$$

Если представить γ_2^* в виде

$$\gamma_2^* = a_0 - a_2 + ((b_1 - b_2) - (a_1 - a_2)) + x^{3^n}(b_0 - b_1 + ((b_1 - b_2) - (a_1 - a_2))),$$

то станет ясно, что все три компоненты γ_i^* можно вычислить за 13 операций сложения-вычитания многочленов степени не выше $3^n - 1$.

В кольце характеристики 2 формулы для γ_i^* принимают вид

$$\gamma_0^* = (a_0 + a_1 + a_2) + x^{3^n}(b_0 + b_1 + b_2),$$

$$\gamma_1^* = (a_0 + a_2 + b_1 + b_2) + x^{3^n}(b_0 + b_1 + a_1 + a_2),$$

$$\gamma_2^* = (a_0 + a_1 + b_1 + b_2) + x^{3^n}(b_0 + b_2 + a_1 + a_2).$$

Если записать

$$\gamma_2^* = \gamma_1^* + (a_1 + a_2) + x^{3^n}(b_1 + b_2),$$

то несложно проверить, что для вычисления компонент γ_i^* можно ограничиться 10 сложениями многочленов степени не выше $3^n - 1$.

Другие два случая рассматриваются аналогично. Например, компоненты преобразования с параметрами $c_1 = c_2 = 1$ имеют вид

$$(a_0 - b_1 - b_2) + x^{3^n}(b_0 - (b_1 - a_2 + b_2 - a_1)),$$

$$(a_0 - a_1 + b_1 - a_2) + x^{3^n}(b_0 + b_2 - a_1),$$

$$(a_0 + a_1 - a_2 + b_2) + x^{3^n}(b_0 + b_1 - a_2),$$

откуда видно, что они могут быть вычислены за 13 операций сложения-вычитания многочленов степени не выше $3^n - 1$. Остальные пункты проверки предоставляются читателю. Лемма доказана.

Лемма 11. *ДПФ порядка 3^k над кольцом $\mathbf{K}_{3,n}(x)$, где $k \leqslant n+1$, может быть реализовано с использованием не более $4,5k3^{n+k}$ операций сложения-вычитания в $\mathbf{K}$, а в случае* $\operatorname{char}\mathbf{K} = 2$ *— не более $3,5k3^{n+k}$ операций сложения.*

Доказательство. Реализация ДПФ порядка 3^{k+1}, если воспользоваться леммой 4 с выбором параметров $S = 3$ и $T = 3^k$, сводится к выполнению 3^k ДПФ порядка 3, трех ДПФ порядка 3^k и умножениям на $x^{3^{n-k}jt}$, где $j = 1,\ 2$ и $t = 1, \ldots, 3^k - 1$.

Пусть $m = c3^n + m'$, где $c \in \mathbb{Z}$ и $|m'| < 3^n/2$. Тогда вместо умножения на x^m будем выполнять умножение на $x^{m'}$, перенося умножение на x^{c3^n} внутрь внешнего ДПФ порядка 3. Поскольку входы любого из внешних ДПФ имеют вид $\gamma_{(0),t}$, $x^l\gamma_{(1),t}$, $x^{2l}\gamma_{(2),t}$, где $l = 3^{n-k}t < 3^n$, то при сведении к умножениям на $x^{m'}$ внешнее ДПФ заменяется одним из преобразований леммы 10.

В рассматриваемой группе умножений умножения на каждую из степеней $x^{m'}$, $m' = 3^{n-k}t$, $t = -(3^k-1)/2, \ldots, (3^k-1)/2$, выполняются по два раза, поскольку

$$\{2t \bmod 3^k \mid t = 1, \ldots, 3^k - 1\} = \{1, \ldots, 3^k - 1\}.$$

Учитывая сложность каждого такого умножения как $|m'|$, сложность совокупности умножений оценивается величиной

$$4 \cdot 3^{n-k} \sum_{t=1}^{\frac{3^k-1}{2}} t = 3^{n-k} \frac{3^{2k}-1}{2}$$

операций сложения-вычитания в $\mathbf{K}$.

Таким образом, для сложности $F_n(3^{k+1})$ построенной схемы имеем рекуррентное соотношение

$$F_n(3^{k+1}) \leqslant 3F_n(3^k) + 3^k F_n(3) + 3^{n+k}/2,$$

которое при начальных условиях $F_n(3) = 13 \cdot 3^n$ (или $F_n(3) = 10 \cdot 3^n$ для кольца характеристики 2) разрешается так, как заявлено в утверждении леммы. Лемма доказана.

Лемма 12. *Умножение в кольце* $\mathbf{K}_{3,1}(x)$ *может быть выполнено за* 30 *операций сложения-вычитания и* 27 *операций умножения в* $\mathbf{K}$.

Доказательство. Представим перемножаемые многочлены $A(x), B(x) \in \mathbf{K}_{3,1}(x)$ в виде $A(x) = A_1(x)x^3 + A_0$, $B(x) = B_1(x)x^3 + B_0$, где $\deg A_i, B_i \leqslant 2$. Их произведение вычислим методом Карацубы:

$$AB = A_1B_1x^6 + ((A_1 - A_0)(B_0 - B_1) + A_1B_1 + A_0B_0)x^3 + A_0B_0.$$

Обозначим $C = (A_1 - A_0)(B_0 - B_1) = C_1x^3 + C_0$, $D = A_0B_0 = D_1x^3 + D_0$, $E = A_1B_1 = E_1x^3 + E_0$, где $\deg C_0, D_0, E_0 \leqslant 2$ и $\deg C_1, D_1, E_1 \leqslant 1$. Тогда в кольце $\mathbf{K}_{3,1}(x)$, т.е. по модулю $x^6 + x^3 + 1$, справедливо соотношение

$$\begin{aligned} AB &= Dx^6 + (C + D + E)x^3 + E = (C + D)x^3 + (D - E) = \\ &= ((D_0 - C_1) + C_0 - E_1)x^3 + ((D_0 - C_1) - E_0 - D_1) = G_1x^3 + G_0. \end{aligned}$$

Произведения C, D, E многочленов степени не выше 2 выполним прямолинейным способом за 9 умножений и 4 сложения каждое. Остальные действия выполняются за 18 операций сложения-вычитания: из них шесть используется для вычисления $A_1 - A_0$, $B_0 - B_1$ и 12 — для вычисления линейных комбинаций G_0, G_1. Лемма доказана.

Теорема 7. *Умножение в кольце* $\mathbf{K}_{3,n}(x)$ *может быть выполнено при помощи* $13,5 \cdot 3^n n(\log_2 n + O(1))$ *операций сложения-вычитания, не более* $3^{n+2} \cdot 2^{\lceil \log_2 n \rceil}$ *операций умножения и* $O(3^n)$ *операций скалярного умножения в* $\mathbf{K}$. *В случае кольца характеристики* 2 *аддитивная сложность составляет не более* $10,5 \cdot 3^n n(\log_2 n + O(1))$.

Доказательство. Теорема доказывается аналогично теореме 5. При $n \geqslant 2$ представим кольцо $\mathbf{K}_{3,n}(x)$ как расширение кольца $\mathbf{K}_{3,m}(y)$ (справедлив аналог леммы 8):

$$\mathbf{K}_{3,n}(x) \cong \mathbf{K}_{3,m}(y)[x]/(x^{3^{n-m}} - y)$$

и выберем $m = \lceil n/2 \rceil$. Как и в двоичном случае, умножение многочленов над $\mathbf{K}_{3,m}(y)$ по модулю $x^{3^{n-m}} - y$ будем выполнять как обычное умножение многочленов степени не выше $3^{n-m} - 1$ с последующим приведением по модулю.

В отличие от двоичного случая (ввиду отсутствия ДПФ порядка $2 \cdot 3^{n-m}$) для умножении используется шесть ДПФ порядка $3^{n-m} = 3^{\lfloor n/2 \rfloor}$: три ДПФ используются обычным образом для вычисления произведения по модулю $x^{3^{n-m}} - 1$, а три других — для вычисления произведения по модулю $x^{3^{n-m}} - \alpha^{3^{n-m}}$, где $\alpha = y^{3^{n \bmod 2}}$, которое сводится к подстановке $x = \alpha z$ и вычислению произведения по модулю $z^{3^{n-m}} - 1$. Действительно,

$$\big(f(x) \bmod (x^N - \alpha^N)\big) \mid_{x=\alpha z} = f(\alpha z) \bmod (z^N - 1).$$

Выполнение любого из преобразований $x = \alpha z$ и $z = x/\alpha$ выполняется в кольце $\mathbf{K}_{3,n}(x)$ посредством $O(3^{n-m})$ операций умножения на степени y в кольце $\mathbf{K}_{3,m}(y)$, т.е. всего за $O(3^n)$ аддитивных операций в $\mathbf{K}$, если проводить вычисления с точностью до множителя ± 1.

Восстановление многочлена $f(x) \in \mathbf{K}_{3,m}(y)[x]$ степени не выше $2N - 2$ по остаткам f_1 и f_α от деления соответственно на $x^N - 1$ и $x^N - \alpha^N$ можно выполнить по формуле

$$f(x) = \frac{1}{\alpha^N - 1}\left((x^N - 1)f_\alpha - (x^N - \alpha^N)f_1\right),$$

причем при $N = 3^{n-m}$ в силу $\alpha^{2N} + \alpha^N + 1 = 0$ множитель $(\alpha^N - 1)^{-1}$ равен $-3^{-1}(\alpha^N + 2)$. Ясно, что описанная процедура восстановления многочлена также может быть выполнена за $O(3^{n-m})$ сложений-вычитаний и умножения на степени y в кольце $\mathbf{K}_{3,m}(y)$, т.е. всего за $O(3^n)$ сложений-вычитаний в $\mathbf{K}$.

Приведение многочлена степени, меньшей $2 \cdot 3^{n-m}$, по модулю $x^{3^{n-m}} - y$ также сводится к $O(3^{n-m})$ операций сложения-вычитания и умножения на степени y в кольце $\mathbf{K}_{3,m}(y)$ или к $O(3^n)$ аддитивных операций в $\mathbf{K}$.

Для чисел $\mu_A(n)$ сложений-вычитаний и $\mu_M(n)$ нескалярных умножений в данной схеме при $n \geqslant 2$ получаем рекуррентные соотношения:

$$\mu_A(n) \leqslant 2 \cdot 3^{\lfloor n/2 \rfloor} \mu_A(\lceil n/2 \rceil) + 6F_{\lceil n/2 \rceil}(3^{\lfloor n/2 \rfloor}) + O(3^n),$$

$$\mu_M(n) \leqslant 2 \cdot 3^{\lfloor n/2 \rfloor} \mu_M(\lceil n/2 \rceil),$$

где значение $F_{\lceil n/2 \rceil}$ определяется из леммы 11. Эти соотношения разрешаются так, как заявлено в утверждении теоремы, если при $n = 1$ воспользоваться оценками леммы 12. Теорема доказана.

Заметим, что метод теоремы 5 дает для сложности умножения многочленов суммарной степени не выше $N - 1$, где $N = 2^k$, асимптотическую оценку $3N \log_2 N \log_2 \log_2 N$, а метод теоремы 7 в случае кольца характеристики 2 и $N = 2 \cdot 3^k$ — близкую оценку $3,32N \log_2 N \log_2 \log_2 N$.

Замечание. *Умножение двоичных трехчленов можно выполнить за* 6 *умножений и* 12 *сложений-вычитаний. Поэтому оценки числа умножений в лемме* 12 *и, как следствие, в теореме* 7 *могут быть уменьшены в* $1,5$ *раза ценой некоторого увеличения числа аддитивных операций.*

7. Заключение

Стратегию умножения в случае $2^{-1}, 3^{-1} \notin \mathbf{K}$ указывает метод Кантора—Калтофена [4]. Способом теорем 5 и 7, только заменяя обратные преобразования $\text{ДПФ}_{N,\zeta}^{-1}$ ненормированными преобразованиями $\text{ДПФ}_{N,\zeta^{-1}}$, вычисляются «почти произведения»

$$2^{N_1} fg = 2^{N_1} fg \bmod (x^{2^{n_1}} + 1), \quad 3^{N_2} fg = 3^{N_2} fg \bmod (x^{2 \cdot 3^{n_2}} + x^{3^{n_2}} + 1)$$

при подходящих $n_1, N_i \in \mathbb{N}$, где f, g — перемножаемые многочлены. Окончательно произведение fg можно вычислить как $q2^{N_1} fg + s3^{N_2} fg$, где q, s — коэффициенты Безу, т.е. $q2^{N_1} + s3^{N_2} = 1$.

Впрочем, актуальность разработки быстрых алгоритмов умножения над такими достаточно экзотическими кольцами представляется пока незначительной.

Работа выполнена при финансовой поддержке РФФИ, проекты 08–01–00863 и 08–01–00632–а, программы «Ведущие научные школы», проект НШ–4470.2008.1, и программы фундаментальных исследований Отделения

математических наук РАН «Алгебраические и комбинаторные методы математической кибернетики» (проект «Синтез и сложность управляющих систем»).

Список литературы

1. Гашков С. Б. Замечания о быстром умножении многочленов, преобразовании Фурье и Хартли // *Дискретная математика.* — 2000. — Вып. 12, №3. — С. 124–153.
2. Bernstein D. J. The tangent FFT // *Proc. AAECC. LNCS.* — 2007. — V. 4851. — P. 291–300.
3. Bernstein D. J. Fast multiplication and its applications // *Algorithmic Number Theory, MSRI Publ.* — 2008. — V. 44. — P. 325–384.
4. Cantor D., Kaltofen E. On fast multiplication of polynomials over arbitrary algebras //*Acta Inf.* — 1991. — V. 28, №7. — P. 693–701.
5. Cooley J., Tukew J. An algorithm for the machine calculation of complex Fourier series //*Math. Comp.* — 1965. — V. 19. — P. 297–301.
6. Good I. J. The interaction algorithm and practical Fourier analysis //*J. R. Statist. Soc. B.* — 1958. — V. 20, №2. — 361–372; 1960. — V. 22, №2. — 372–375.
7. Schönhage A. Schnelle multiplikation von polynomen über körpern der charakteristik 2 // *Acta Inf.* — 1977. — V. 7. — P. 395–398.
8. Schönhage A., Strassen V. Schnelle multiplikation großer zahlen // *Computing.* — 1971. — V. 7. — P. 271–282. [Русский перевод: Шёнхаге А., Штрассен В. Быстрое умножение больших чисел // Кибернетический сборник. Вып. 10. — М.: Мир, 1973. — С. 87–98].

АСИМПТОТИЧЕСКИ ХОРОШИЕ КОДЫ С ЛИНЕЙНОЙ СЛОЖНОСТЬЮ КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ

Д. А. Жуков

Московский государственный технический университет им. Н. Э. Баумана,
105005 Москва, 2-я Бауманская ул., д. 5

e-mail: oldbug@mail.ru

1. Определения

Множество $\{0,1\}^n$ всех двоичных наборов длины n называется *булевым кубом* размерности n. Двоичные наборы одинаковой длины можно складывать (покомпонентно по модулю 2) и умножать на скаляры из поля $\mathbb{F}_2$. Относительно этих операций булев куб является линейным векторным пространством. Куб является также и метрическим пространством с *расстоянием Хемминга* между векторами $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n) \in \{0,1\}^n$, определяемым как

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|.$$

Обозначим $\|x\|$ *вес* вектора x; он равен числу единиц в x. Нетрудно убедится, что расстояние $d(x,y)$ между векторами x, y равно числу разрядов в которых они различаются, а также величине $\|x \oplus y\|$. В метрическом пространстве можно определить множество

$$B_r(a) = \{x : d(a,x) \leqslant r\},$$

называемое *шаром* радиуса r с центром a.

Любое линейное подпространство $\mathcal{C}$ в кубе $\{0,1\}^n$ называется *линейным кодом* длины n. Его элементы называются *кодовыми словами*. Основными параметрами линейного кода являются длина, размерность и кодовое расстояние. *Размерностью* k кода $\mathcal{C}$ называется его размерность как линейного векторного простанства над полем $\mathbb{F}_2$, то есть число векторов в каком-либо базисе в $\mathcal{C}$. Очевидно, что если $\dim \mathcal{C} = k$, то $|\mathcal{C}| = 2^k$. Отношение $R(\mathcal{C}) = k/n$ размерности кода к его длине называется *скоростью* кода. *Кодовым расстоянием* линейного кода называется минимальное расстояние между различными кодовыми словами:

$$d(\mathcal{C}) = \min_{x_1 \neq x_2 \in \mathcal{C}} d(x_1, x_2).$$

Легко видеть, что в силу линейности кодовое расстояние также равно минимальному весу ненулевого кодового слова: $d(\mathcal{C}) = \min_{x \in \mathcal{C}, x \neq 0} \|x\|$. Величина $\delta(\mathcal{C}) = d(\mathcal{C})/n$ называется *относительным кодовым расстоянием* кода $\mathcal{C}$. Для сокращения будем называть линейный код длины n и размерности k (n,k)-кодом, или (n,k,d)-кодом, чтобы подчеркнуть, что его кодовое расстояние равно d.

Двоичная матрица $G = G_{k \times n}$ называется *порождающей* матрицей кода $\mathcal{C}$, если её строки образуют базис в $\mathcal{C}$. Матрица H называется *проверочной* матрицей кода $\mathcal{C}$, если она ортогональна всем векторам из $\mathcal{C}$ и только им:

$$\forall\, x \in \mathcal{C} : Hx = 0 \quad \text{и} \quad \forall\, x \notin \mathcal{C} : Hx \neq 0.$$

Вектор Hy называется *синдромом* вектора y. Таким образом, проверочная матрица позволяет достаточно просто определить, является ли вектор кодовым: это так в том и только в том случае, когда у него нулевой синдром. По матрице H можно также найти кодовое расстояние кода, как утверждает следующая теорема.

Теорема 1. *Пусть $\mathcal{C}$ — (n,k)-код с проверочной матрицей H. Тогда для того, чтобы кодовое расстояние кода равнялось $d > 1$, необходимо и достаточно выполнения двух условий*: 1) *любые $(d-1)$ столбцов матрицы H линейно независимы*; 2) *в матрице H найдутся d линейно зависимых столбцов.*

Из линейной алгебры хорошо известно, что проверочная (порождающая) матрица существует для каждого линейного кода, и что, вообще говоря, она не единственна: один и тот же код можно задать разными проверочными (порождающими) матрицами.

Линейные (n, k)-коды используются при передаче данных по каналам связи, вносящим ошибки. Передатчик делит информационную последовательность на блоки длины k и независимо кодирует каждый блок кодовым словом длины n. Наиболее естественный способ кодирования — умножение информационного блока на порождающую матрицу G. При этом информационный вектор является координатами разложения кодового слова по базису, состоящему из строк G. В особенно простом частном случае, когда матрица G имеет вид $(E|B)$, называемый *систематическим* (здесь $E = E_{k\times k}$ — единичная матрица, $B = B_{k\times(n-k)}$ — матрица общего вида), первые k разрядов всякого кодового слова совпадают с разрядами информационного. Они называются *информационными* разрядами кодового слова, остальные $(n-k)$ разрядов — *проверочными*, а само кодирование — *систематическим*. Можно легко убедиться, что всякая порождающая матрица элементарными преобразованиями строк и, возможно, перестановкой столбцов, приводится к систематическому виду, и это преобразование не меняет метрических свойств кода. Поэтому любой линейный код может быть представлен в эквивалентном систематическом виде.

После кодирования информационного сообщения передатчик передаёт кодовое слово в канал связи, который вносит в него ошибки. Предположим, что разряды при передаче по каналу не могут пропадать или невосстановимо искажаться, а могут только инвертироваться, такие разряды называются *ошибочными*. Если x — кодовое слово, а y — принятое, то вектор $e = x \oplus y$ называется *вектором ошибки*. В случае, когда каждый разряд инвертируется с вероятностью p независимо от остальных, канал связи называется *двоичным симметричным каналом* (без памяти) с вероятностью ошибки p, сокращённо ДСК(p).

На приёмном конце возникает задача восстановления кодового слова по принятому искажённому. Такое преобразование называется *декодированием*. Заметим, что если (n, k, d)-код имеет кодовое расстояние $d = 2t + 1$, то он способен исправлять вплоть до t произвольных ошибок. Действительно, если при передаче слова $x \in \mathcal{C}$ произошло $r \leqslant t$ ошибок, то мы получим такое искажённое слово y, что $d(x, y) = r$. Значит, y лежит в шаре радиуса t с центром x. Шары радиуса t с центрами в кодовых словах не пересекаются[1)]. Поэтому по y при условии $d(x, y) \leqslant t$ можно однозначно восстановить кодовое слово x, например, одним из двух очевидных способов: перебором

[1)] Действительно, если x_1 и x_2 — различные кодовые слова, то $d(x_1, x_2) \geqslant 2t + 1$. Если мы предположим, что $B_t(x_1) \cap B_t(x_2) \neq \varnothing$, то получим противоречие с неравенством треугольника, которому должна удовлетворять метрика d: существует $y \in B_t(x_1) \cap B_t(x_2)$ для которого $d(x_1, x_2) \leqslant d(x_1, y) + d(y, x_2) \leqslant 2t$.

всех кодовых слов с поиском среди них ближайшего к y слова, либо перебором всех векторов ошибок e веса не более t с проверкой на принадлежность коду слова $y \oplus e$.

При больших n, k, d оба эти способа декодирования становится невозможны — их сложность экспоненциальна. Но именно такие коды и имеют особый теоретический и прикладной интерес.

Определение. *Бесконечное семейство $\{\mathcal{C}_n\}$, состоящее из (n, k_n, d_n)-кодов, называется асимптотически хорошим при $n \to \infty$, если существуют такие положительные константы R и δ, не зависящие от n, что при всех достаточно больших n*

- $\frac{k_n}{n} \geqslant R > 0$,
- $\frac{d_n}{n} \geqslant \delta > 0$.

Асимптотически хорошие коды требуются для передачи информации по каналам типа ДСК(p), где матожидание числа ошибок при передаче n символов равно pn и растёт линейно с n. Существование таких кодов следует из теоремы Варшамова — Гильберта (см., напр. [4, 8]), которая утверждает, что если выполнено неравенство

$$\sum_{i=0}^{d-2} \binom{n-1}{i} < 2^{n-k}, \tag{1}$$

то существует линейный (n, k, d)-код. Её асимптотическая форма утверждает, что для почти всех линейных (n, k)-кодов при $n \to \infty$ выполняется неравенство

$$R \gtrsim 1 - H(\delta), \tag{2}$$

где $R > 0$ и $\delta > 0$ — скорость и относительное кодовое расстояние, а функция $H(x) = -x \log x - (1 - x) \log(1 - x)$ называется *бинарной энтропией*[2)].

Для уточнения понятия алгоритма в настоящей работе выбрана модель RAM (random access machine), т.е. машина с произвольным доступом к памяти [2]. Такая машина имеет один процессор с неограниченным объёмом памяти, разбитой на занумерованные ячейки-регистры, и работает под управлением программы. Различные разряды кодовых и искажённых слов записываются в отдельные ячейки памяти, но каждая ячейка способна хранить произвольное целое число. Программа для RAM — это набор команд.

[2)] Здесь и далее все логарифмы берутся по основанию 2.

За один шаг работы процессора выполняется одна команда (элементарная операция), при этом процессор может прочесть содержание любой ячейки или записать в неё некоторые данные, выполнить в данной ячейке некоторую арифметическую операцию, выполнить условный переход к команде с номером k по содержанию некоторого регистра, остановиться и т.п.

Мы ограничимся *равномерной* разновидностью этой модели (uniform cost model), в которой все RAM-команды затрачивают одинаковое время на своё исполнение, скажем, завершаются за один такт работы процессора. Однако результаты, которые будут получены нами ниже, легко могут быть перенесены без изменения порядка сложности и на более слабую (и более реалистичную) *логарифмическую* модель (logarithmic cost model), в которой время исполнения команды пропорционально длине участвующих в ней чисел[3)] (все подробности можно найти в [17]).

Сложностью программы будем считать время её работы. В равномерной модели она пропорциональна числу элементарных операций, произведённых процессором от начала до конца работы.

Обозначим $L_e(\mathcal{C})$ сложность кодирования кодом $\mathcal{C}$, т.е. число элементарных операций, которые требуются чтобы получить из информационного слова кодовое. Аналогично, обозначим $L_d(\mathcal{C}, t)$ сложность декодирования одного кодового слова из $\mathcal{C}$, испорченного не более чем в t разрядах, т.е. число операций, требуемых в худшем случае чтобы получить из искажённого слова $y = x \oplus e$ исходное слово $x \in \mathcal{C}$ при условии $\|e\| \leqslant t$. Если t ясно из контекста (как правило оно ограничено половиной кодового расстояния), то будем записывать просто $L_d(\mathcal{C})$.

Справедливо следующее утверждение.

Утверждение 1. *Сложность кодирования линейным кодом с порождающей матрицей G одного информационного слова не превышает по порядку числа единиц в G.*

Таким образом, если $\mathcal{C}$ — (n,k)-код, то $L_e(\mathcal{C}) \leqslant \mathcal{O}(nk)$. Отметим, что если код $\mathcal{C}$ асимптотически хороший, то любое его ненулевое слово имеет вес не менее δn, где δ — некоторая постоянная. Таким образом, любая его порождающая матрица содержит не менее $rn \times \delta n$ единиц, где $r > 0$ — скорость кода, и вышеописанный метод кодирования, основанный на умножении матрицы на вектор, будет иметь сложность $\Omega(n^2)$. Для кодов общего вида более простые способы кодирования неизвестны.

[3)] Например, чтение n-разрядного числа из ячейки памяти, чей адрес записан m разрядами, занимает $n + m$ единиц времени.

Декодирование, вообще говоря, является значительно более трудной операцией, чем кодирование. Так, для произвольного линейного кода декодирование в ближайшее кодовое слово и даже нахождение кодового расстояния является NP-трудной задачей [10]. Про декодирование внутри кодового расстояния это пока неизвестно, но неизвестны и полиномиальные алгоритмы.

2. Представление кода двудольным графом

Пусть $\mathcal{G} = (V, E)$ — двудольный граф с долями L и R, так что $V = L \sqcup R$. Мы будем называть его (A, B)-*регулярным*, если степень каждой вершины из левой доли L равна A, а из правой R — равна B. Очевидно, что в этом случае $A \cdot |L| = B \cdot |R|$. Пример регулярного графа приведён на рис. 1. Если степени вершин из левой и правой доли не превосходят A и B соответственно, то будем называть такой граф (A, B)-*ограниченным*. Ясно, что ограниченный граф не обязательно регулярен.

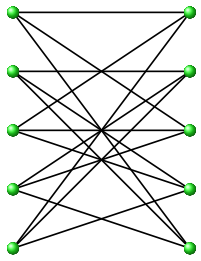

Рис. 1

Обозначим через

$$\Gamma(v) = \{u : (u, v) \in E\}$$

окрестность вершины $v \in V$ и через

$$\Gamma(S) = \bigcup_{v \in S} \Gamma(v)$$

окрестность множества вершин $S \subset V$. Окрестность множества вершин также часто называют его *тенью*. В дальнейшем нам как правило будут встречаться ситуации когда $S \subset L$ или $S \subset R$.

Нам также потребуются множества

$$\Gamma_1(S) = \{u \in \Gamma(S) : |\Gamma(u) \cap S| = 1\},$$

$$\Gamma_{\text{неч}}(S) = \{u \in \Gamma(S) : |\Gamma(u) \cap S| \text{ — нечётно}\},$$

состоящие из вершин тени множества S с единственным соседом в S и с нечётным числом соседей в S соответственно. Очевидно, что

$$\Gamma(S) \supseteq \Gamma_{\text{неч}}(S) \supseteq \Gamma_1(S). \tag{3}$$

По линейному (n, k)-коду $\mathcal{C}$ с проверочной матрицей $H = H_{(n-k)\times n}$ построим двудольный граф $\mathcal{G} = (L \sqcup R, E)$ с матрицей смежности H. В этом графе вершина с номером i из доли L соединяется ребром с вершиной с номером j из доли R в том и только в том случае, когда на пересечении i-го столбца и j-й строки матрицы H стоит единица. Вершины из левой доли L пометим различными символами x_i, $i = 1, \ldots, n$ и будем называть *кодовыми* вершинами. Вершины из правой доли R пометим символами s_j, $j = 1, \ldots, n-k$ и будем называть *проверочными* или *синдромными*. Таким образом, $|L| = n$ и $|R| = n - k$. Отметим, что степень вершины $x_i \in L$ равна числу единиц в i-м столбце проверочной матрицы, а степень вершины $s_j \in R$ равна числу единиц в её j-м столбце. Будем говорить, что граф $\mathcal{G}$ представляет код $\mathcal{C}$, а код $\mathcal{C}$ порождает граф $\mathcal{G}$.

Такой граф называется *графом Таннера* кода $\mathcal{C}$ [19]. Очевидно, что по линейному коду всегда можно построить представляющий его граф Таннера. Обратно, всякий двудольный граф является графом Таннера для некоторого линейного кода.

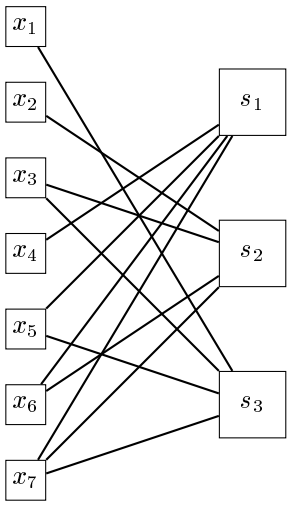

Рис. 2

В качестве примера на рис. 2 приведён граф Таннера для $(7, 4, 3)$-кода Хемминга с проверочной матрицей

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Вектор $(x_1, \ldots, x_7)$ и его синдром (s_1, s_2, s_3) связаны соотношением

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix}.$$

Глядя же на граф слева нетрудно заметить, что значение каждой синдромной вершины равно сумме значений тех и только тех кодовых вершин, которые смежны с ней. Следуя этой идее, свяжем с каждой синдромной вершиной

s_j и в общем случае линейную проверку, соответствующую j-й строке матрицы $H = (h_{ji})$ по формуле

$$s_j = \bigoplus_{l=1}^{n} h_{jl} x_l. \tag{4}$$

Тогда вершины $(s_1, \ldots, s_{n-k})$ образуют синдром вектора $(x_1, \ldots, x_n)$.

По графу Таннера так же как и по проверочной матрице (теорема 1) можно найти кодовое расстояние. Пусть $\mathcal{C}$ — линейный код и $\mathcal{G}$ — его граф Таннера. Рассмотрим какое-нибудь слово $x = (x_1, \ldots, x_n)$ веса w. Мы знаем, что оно является кодовым тогда и только тогда, когда у него нулевой синдром:

$$x \in \mathcal{C} \quad \Leftrightarrow \quad \forall\, s_j \in R: \;\; s_j = \bigoplus_{l=1}^{n} h_{jl} x_l = \bigoplus_{l:x_l=1} h_{jl} = 0.$$

Между словами x и множествами $S \subseteq L$ вершин левой доли графа можно установить биекцию по правилу $S = S(x) = \{x_i \in L : x_i = 1\}$. Сумма $\sum_{l:x_l=1} h_{jl}$ равна мощности множества $\Gamma(s_j) \cap S(x)$; значит синдром $s_j = \bigoplus_{l:x_l=1} h_{jl}$ равен нулю тогда и только тогда, когда у него чётное число соседей в $S(x)$. Поэтому в коде $\mathcal{C}$ есть кодовое слово веса w в том и только в том случае, когда в левой доле графа $\mathcal{G}$ есть множество вершин S мощности w, у всех вершин тени которого чётное число соседей из S:

$$\exists\, x \in \mathcal{C},\ \|x\| = w \quad \Leftrightarrow \quad \exists\, S \subset L,\ |S| = w : \Gamma_{\text{неч}}(S) = \varnothing. \tag{5}$$

Рассмотрим два свойства графа $\mathcal{G}$ (предполагается, что $S \neq \varnothing$):

$$\forall\, S \subset L: \;\; |S| \leqslant w \Rightarrow \Gamma_{\text{неч}}(S) \neq \varnothing, \tag{6}$$

$$\forall\, S \subset L: \;\; |S| \leqslant w \Rightarrow \Gamma_1(S) \neq \varnothing. \tag{7}$$

Из (3) следует, что если граф обладает свойством (7), то он обладает и свойством (6). Тогда из (5) получаем следующий достаточный признак большого кодового расстояния.

Утверждение 2. *Если в графе Таннера кода $\mathcal{C}$ выполнено свойство* (7) *для некоторого w, то кодовое расстояние кода $\mathcal{C}$ строго больше w.*

Множество графов, обладающих свойством (6), до сих пор не описано. Тем не менее довольно давно известен достаточно широкий класс графов, для которого выполнено (7).

Определение. *Двудольный граф* $\mathcal{G} = (L \sqcup R, E)$, $n = |L|$, *называется* (γ, δ)-*расширителем*[4)], *если*

$$\forall\, S \subset L:\ |S| \leqslant \delta n \quad \Rightarrow \quad |\Gamma(S)| \geqslant \gamma \cdot |S|. \tag{8}$$

Число γ *называется коэффициентом расширения.*

Проще говоря, расширитель — это такой граф, в котором каждое достаточно малое множество вершин обладает достаточно большой тенью. Очевидно, что если степень вершин левой доли расширителя не превосходит A, то для его коэффициента расширения справедливо неравенство $\gamma \leqslant A$. Примером расширителя может служить двудольный граф на рис. 1. Действительно, нетрудно убедиться, что

$$|\Gamma(S)| = \begin{cases} 3, & \text{если } |S| = 1, \\ \geqslant 4, & \text{если } |S| = 2, \\ 5, & \text{если } |S| \geqslant 3, \end{cases}$$

если множество S целиком лежит в одной его доле. Следовательно, свойство (8) выполняется для него, например, при $\delta = \frac{2}{5}$, $\gamma = 2$ или при $\delta = \frac{3}{5}$, $\gamma = \frac{5}{3}$.

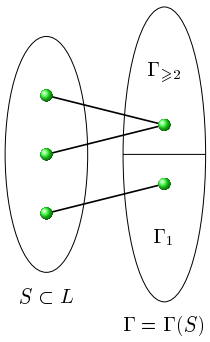

Рис. 3

Далее будем для сокращения обозначать через $\mathcal{G}^{\gamma,\delta}_{A,B,n}$ двудольный (γ, δ)-расширитель, в левой и правой долях которого n и $n/2$ вершин соответственно, причём степени вершин левой и правой доли ограничены константами A и B. Отложим на время вопрос о существовании графа $\mathcal{G}^{\gamma,\delta}_{A,B,n}$ и покажем сначала, что среди расширителей есть графы со свойством (7).

Пусть $\mathcal{G} = \mathcal{G}^{\gamma,\delta}_{A,B,n}$ — двудольный расширитель, и пусть $S \subset L$ — достаточно малое множество вершин из левой доли, так что $|S| \leqslant \delta n$, где $n = |L|$. Рассмотрим множество

$$\Gamma_{\geqslant 2} = \Big\{ v \in \Gamma(S) : |\Gamma(v) \cap S| \geqslant 2 \Big\}$$

тех вершин его тени, которые имеют не менее двух соседей в S (см. рис. 3). Оценим число l рёбер, ведущих из множества S в его тень $\Gamma(S)$, двумя

[4)] или экспандером от англ. expander.

разными способами. С одной стороны, так как из всякой вершины $v \in S$ выходит не более A рёбер, то $l \leqslant A \cdot |S|$. С другой стороны, в каждую вершину из $\Gamma_1 = \Gamma_1(S)$ входит ровно одно ребро, а в каждую вершину из $\Gamma_{\geqslant 2}$ — не менее двух рёбер. Поэтому $l \geqslant 1 \cdot |\Gamma_1| + 2 \cdot |\Gamma_{\geqslant 2}|$. Отсюда получаем неравенство

$$|\Gamma_1| + 2|\Gamma_{\geqslant 2}| \leqslant A \cdot |S|.$$

Кроме того, согласно свойству расширения множества S,

$$|\Gamma_1| + |\Gamma_{\geqslant 2}| = |\Gamma| \geqslant \gamma \cdot |S|.$$

Выражая $|\Gamma_1|$ из этих двух неравенств получаем

$$2|\Gamma_1| \geqslant 2\gamma \cdot |S| - 2|\Gamma_{\geqslant 2}| \geqslant 2\gamma \cdot |S| - A \cdot |S| + |\Gamma_1|,$$

откуда

$$|\Gamma_1| \geqslant (2\gamma - A) \cdot |S|. \tag{9}$$

Оценим также мощность множества $\Gamma_{\geqslant 2}$ сверху. В силу того, что $|\Gamma_1| \leqslant A|S| - 2|\Gamma_{\geqslant 2}|$, получаем $\gamma \cdot |S| \leqslant |\Gamma_1| + |\Gamma_{\geqslant 2}| \leqslant A|S| - |\Gamma_{\geqslant 2}|$, откуда

$$|\Gamma_2| \leqslant (A - \gamma) \cdot |S|. \tag{10}$$

Таким образом, мы доказали следующее простое, но крайне полезное утверждение.

Утверждение 3. *Пусть $|L| = n$, $|R| = \frac{n}{2}$, граф $\mathcal{G} = (L \sqcup R, E)$ является (γ, δ)-расширителем и степень вершин его левой доли не превосходит A. Тогда для тени всякого непустого множества $S \subset L$, такого что $|S| \leqslant \delta n$, справедливо неравенство*

$$|\Gamma_1(S)| \geqslant (2\gamma - A) \cdot |S|.$$

Заметим, что при $\gamma > \frac{A}{2}$ отсюда следует, что $|\Gamma_1(S)| > 0$ при всяком таком $S \neq \varnothing$, что $|S| \leqslant \delta n$. Следовательно, $\Gamma_1(S) \neq \varnothing$, и поэтому граф $\mathcal{G}$ обладает свойством (7). Таким образом, из утверждений 2 и 3 получаем следующую теорему.

Теорема 2. *Пусть $\mathcal{G}$ — двудольный (γ, δ)-расширитель, степень вершин левой доли которого не превосходит A, причём $\gamma > \frac{A}{2}$. Тогда линейный $(n, \frac{n}{2}, d)$-код $\mathcal{C}$, для которого граф $\mathcal{G}$ является графом Таннера, имеет кодовое расстояние d не меньшее, чем δn.*

Такой код, построенный с помощью графа-расширителя, называется *экспандерным кодом* (expander code) или кодом Сипсера — Спайлмана [17]. Будем обозначать его $\mathcal{C}_n^{\gamma,\delta}$. Это наш первый пример асимптотически хорошего кода. Отметим, что пока нам не потребовалась ограниченность степеней вершин правой доли расширителя $\mathcal{G}$ константой B.

3. Существование расширителей

При $A, B = \text{const}$ число рёбер в графе $\mathcal{G}_{A,B,n}^{\gamma,\delta}$ не более чем в константу раз превышает число вершин. Такие графы являются экстремальными — в них мало рёбер, и тем не менее даже у достаточно значительных множеств вершин большие тени. В этом разделе мы докажем обещанную теорему о существовании таких расширителей, причём с произвольно большими коэффициентами расширения.

Теорема 3. *Пусть ε — произвольное положительное число. Тогда для каждого достаточно большого чётного $n \geqslant n_0$ существует двудольный (γ, δ)-расширитель $\mathcal{G} = (L \sqcup R, E)$, такой, что:*

- $|L| = n$, $|R| = \frac{n}{2}$;
- *степень каждой вершины из левой доли не превосходит некоторой константы A;*
- *степень каждой вершины из правой доли не превосходит некоторой константы B;*
- $\delta = \delta(\varepsilon, A) > 0$ *— некоторая постоянная, не зависящая от n;*
- $\gamma \geqslant (1 - \varepsilon)A$.

Доказательство. Рассмотрим следующую вероятностную конструкцию графа $\mathcal{G}$. Возьмём множество $L' = \{1, 2, \ldots, An\}$, состоящее из An целых чисел и применим к нему случайную перестановку α из S_{An}. Константу A выберем позднее.

Перестановку α можно представить двудольным графом $\mathcal{G}'$. Его левой долей является множество L', правая доля R' равномощна левой и её вершины также занумерованы числами от 1 до An. Вершины $x \in L'$ и $y \in R'$ соединяются ребром, когда $y = \alpha x$. Таким образом, степень любой вершины графа $\mathcal{G}' = (L' \sqcup R', E')$ равна единице и $\mathcal{G}'$ является совершенным паросочетанием.

Объединим элементы из L' в кластеры X_i из A вершин в порядке следования их номеров: в первый кластер X_1 войдут элементы с номерами $1, 2, \ldots, A$, во второй — с номерами $A+1, \ldots, 2A$, и т.д. Аналогично поступим со множеством R', также разбив его на кластеры Y_j по B вершин в каждом. Рассмотрим множества L и R, состоящие из элементов-кластеров:

$$L = \{X_1, \ldots, X_n\}, \qquad R = \{Y_1, \ldots, Y_{An/B}\}.$$

Положим $B = 2A$, тогда $|R| = \frac{n}{2}$. Кластеры X_i и Y_j, где $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant n/2$ будем считать вершинами нового двудольного графа $\mathcal{G} = (L \sqcup R, E)$. Вершины X_i и Y_j соединим ребром в графе $\mathcal{G}$ в том и только в том случае, когда найдутся вершины $x \in X_i$ и $y \in Y_j$, соединённые ребром в исходном графе $\mathcal{G}'$:

$$\exists\, x \in X_i \quad \exists\, y \in Y_j \ : \quad y = \alpha x.$$

В полученном двудольном графе $\mathcal{G}$ степень любой вершины из доли L по построению не превосходит A, а степень любой вершины из доли R — $2A$. Можно считать, что граф $\mathcal{G}$ получен из случайного паросочетания $\mathcal{G}'$ «стягиванием» вершин с последующим удалением возможных кратных рёбер. Покажем, что с высокой вероятностью граф $\mathcal{G}$ является искомым расширителем.

Пусть $S \subset L$ и $s = |S| \leqslant \delta n$. Множеству S соответствует множество вершин $S' \subset L'$ паросочетания, входящих в кластеры из S. Из вершин множества S' в графе $\mathcal{G}'$ выходит ровно As рёбер. Зафиксируем какую-нибудь нумерацию этих рёбер $e_1, e_2, \ldots, e_{As}$, например в порядке следования их левых концов в L'. Возьмём сначала первое ребро e_1 и рассмотрим его правый конец, затем второе e_2, и т.д. Правые концы всех рёбер входят в какие-то кластеры. Будем говорить, что вершина $v \in R'$ является *помеченной* по отношению к ребру e_i если хотя бы одно из предыдущих рёбер $e_1, \ldots, e_{i-1}$ инцидентно какой-либо вершине из её B-кластера. Назовём вершину просто *помеченной*, если она помечена относительно хотя бы одного ребра. Таким образом, любое множество S индуцирует разбиение доли R' на помеченные и непомеченные вершины.

Рассмотрим случайную величину ξ_i, $1 \leqslant i \leqslant As$:

$$\xi_i = \begin{cases} 1, & \text{если ребро } e_i \text{ ведёт в помеченную относительно } e_i \text{ вершину,} \\ 0, & \text{иначе.} \end{cases}$$

По построению число непомеченных вершин в $\Gamma(S')$ равно числу кластеров в $\Gamma(S)$, а число помеченных вершин в $\Gamma(S')$ равно сумме $\sum_{i=1}^{As} \xi_i$. Кроме

того, число соединяющих S и $\Gamma(S)$ рёбер не превосходит числа рёбер, соединяющих S' и $\Gamma(S')$ в графе $\mathcal{G}'$, поэтому

$$|\Gamma(S)| \leqslant As - \sum_{i=1}^{As} \xi_i.$$

Оценим величину $\sum_{i=1}^{As} \xi_i$. Для этого заметим, что существует

$$An - i + 1 \geqslant An - As \geqslant An(1-\delta)$$

способов выбора правого конца i-го ребра, и что занятых правых концов рёбер не более чем $As \cdot B$ при всяком i. Следовательно,

$$P[\xi_i = 1 | \xi_1, \ldots, \xi_{i-1}] \leqslant \frac{As \cdot B}{An(1-\delta)} = \frac{s}{n} \cdot \frac{B}{1-\delta}.$$

Обозначим $p = \frac{s}{n} \cdot \frac{B}{1-\delta}$ и воспользуемся следующей известной из теории вероятностей оценкой суммы произвольных, не обязательно независимых случайных величин, аналогичной неравенству Ацумы [1].

Утверждение 4. *Пусть $\xi_i \in \{0,1\}$, $i = 1, \ldots, n$ — последовательность случайных величин, причём*

$$P[\xi_i = 1 | \xi_1, \xi_2, \ldots, \xi_{i-1}] \leqslant p$$

при $i = 1, \ldots, n$. Тогда для всех $x > p$ выполнено

$$P\left[\sum_{i=1}^{n} \xi_i \geqslant xn\right] \leqslant \left(\frac{ep}{x}\right)^{nx}.$$

В нашем случае применение этого утверждения даёт неравенство

$$P\left[\sum_{i=1}^{As} \xi_i \geqslant x \cdot As\right] \leqslant \left(\frac{ep}{x}\right)^{x \cdot As}$$

для произвольного $x \in \left(\frac{s}{n} \cdot \frac{B}{1-\delta}, 1\right)$, откуда

$$P\Big[\ |\Gamma(S)| = As - \sum \xi_i \leqslant As - xAs\ \Big] \leqslant \left(\frac{ep}{x}\right)^{x \cdot As}.$$

Отметим, что с помощью подходящего выбора $\frac{s}{n}$ величина x может быть сделана сколь угодно малой.

Таким образом, вероятность того, что конкретное множество S мощности s «плохо расширяется» не превосходит величины

$$P\Big[\ |\Gamma(S)| \leqslant As(1-x)\ \Big] \leqslant \left(\frac{s}{n}\cdot\frac{eB}{x(1-\delta)}\right)^{x\cdot As}.$$

Пусть

$$\eta_S = \begin{cases} 1, & \text{если } |\Gamma(S)| \leqslant A\cdot|S|\cdot(1-x), \\ 0, & \text{иначе.} \end{cases}$$

Случайная величина $\sum_S \eta_S$, где сумма берётся по всем множествам $S \subset L$ таким что $|S| \leqslant \delta n$, принимает положительное значение в том и только в том случае, когда коэффициент расширения γ графа $\mathcal{G}$ не превосходит $A(1-x)$. Число различных подмножеств S мощности s равно $\binom{n}{s}$. Более того,

$$\binom{n}{s} < \sum_{i=0}^{s}\binom{n}{i} \leqslant \left(\frac{en}{s}\right)^s. \tag{11}$$

Поэтому вероятность того, что $\mathcal{G}$ — не (γ,δ)-расширитель при $\gamma = A(1-x)$ не превосходит

$$\begin{aligned} P\left[\sum_S \eta_S\right] &\leqslant \sum_S P\left[\eta_S = 1\right] \leqslant \\ &\leqslant \sum_{s=1}^{\delta n}\binom{n}{s}\left(\frac{s}{n}\cdot\frac{eB}{x(1-\delta)}\right)^{x\cdot As} \leqslant \\ &\leqslant \sum_{s=1}^{\delta n}\left(\frac{en}{s}\right)^s\left(\frac{s}{n}\cdot\frac{eB}{x(1-\delta)}\right)^{x\cdot As} = \\ &= \sum_{s=1}^{\delta n}\left(e\cdot\left(\frac{eB}{x(1-\delta)}\right)^{xA}\cdot\left(\frac{s}{n}\right)^{xA-1}\right)^s \leqslant \\ &\leqslant \sum_{s=1}^{\delta n}\left(e\cdot\left(\frac{2eA}{x(1-\delta)}\right)^{xA}\cdot\delta^{xA-1}\right)^s. \end{aligned}$$

Последнее выражение оценивается сверху суммой $\sum_{s=1}^{\infty}(c\delta)^s$ в случае, когда, например, $xA = 2$, где

$$c = e \cdot \left(\frac{200}{99} \cdot \frac{eA}{x}\right)^{xA}$$

при условии, что $\delta < 1/100$. Выбирая параметр δ достаточно малым можно добиться того, что и сумма ряда $\sum_{s=1}^{\infty}(c\delta)^s$ будет сколь угодно мала. Так, если $x = \frac{1}{8}$ и $A = 16$, то при

$$\delta < \frac{1}{5c} = \frac{1}{5e}\Big(\frac{200}{99} \cdot 8e \cdot 16\Big)^{-2} < 1.48 \cdot 10^{-7}$$

вероятность того, что $\mathcal{G}$ — не расширитель не превосходит $\sum_{s=1}^{\infty} 5^{-s} = \frac{1}{4}$. Теорема доказана. □

4. Алгоритм инвертирования бита

В разделе 2 мы убедились, что построенные по графам-расширителям коды асимптотически хороши. Здесь мы покажем, что они обладают к тому же достаточно эффективным алгоритмом декодирования.

Рассмотрим экспандерный код $\mathcal{C}$ длины n из теоремы 2. Пусть $\mathcal{G} = \mathcal{G}^{\gamma,\delta}_{A,B,n}$ — его граф Таннера, причём $A, B = \mathcal{O}(1)$. Пусть $y = (y_1, \ldots, y_n)$ — произвольный двоичный вектор длины n. Поместим его разряды в кодовые вершины графа $\mathcal{G}$. Тогда согласно (4) значениями проверочных вершин графа $\mathcal{G}$ будут компоненты синдрома Hy, где H — проверочная матрица кода $\mathcal{C}$.

Раскрасим проверочные вершины чёрным и белым цветом так, что каждая вершина с нулевым синдромом белая, а с ненулевым — чёрная. Обозначим

$$R_\circ = \{s_m \in R : s_m = 0\},$$
$$R_\bullet = \{s_m \in R : s_m = 1\}$$

множества белых и чёрных вершин в $\mathcal{G}$. Таким образом, любой двоичный вектор y длины n задаёт некоторую раскраску правой доли графа Таннера и её разбиение на два подмножества $R_\circ(y)$ и $R_\bullet(y)$.

Предположим, что $x = (x_1, \ldots, x_n)$ — некоторое кодовое слово из $\mathcal{C}$, а слово $y = (y_1, \ldots, y_n)$ отличается от x не более чем в $\frac{\delta n}{2}$ разрядах. Рассмотрим раскраску доли R, индуцированную словом y. Очевидно, что слово y — кодовое тогда и только тогда, когда $R_\bullet(y) = \varnothing$.

Пусть $S \subset L$ — произвольное множество кодовых вершин. Обозначим

$$\Gamma_\circ(S) = \Gamma(S) \cap R_\circ, \quad \Gamma_\bullet(S) = \Gamma(S) \cap R_\bullet$$

множества белых и чёрных вершин из его тени. Назовём кодовую вершину $y_j \in L$ *плохой*, если $|\Gamma_\circ(y_j)| < |\Gamma_\bullet(y_j)|$, то есть если у неё большинство соседей чёрного цвета. В противном случае, даже если чёрных и белых соседей у y_j поровну, она *хорошая*. Пример плохой вершины приведён на рис. 4.

Интуиция подсказывает, что если у кодовой вершины много чёрных соседей-синдромов, то она скорее всего ошибочная. На этом наблюдении основан следующий метод декодирования, восходящий к работе Галлагера [5].

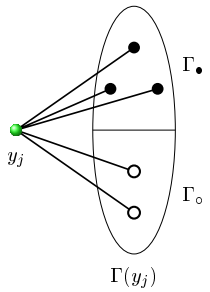

Рис. 4

Алгоритм A.

1. Вычислить в графе $\mathcal{G}$ значения всех проверочных вершин s_m из R.
2. Пока множество плохих вершин непусто, выбрать из него какую-нибудь плохую вершину $y_j \in L$, заменить её значение на противоположное $y_j \to \overline{y}_j$, и пересчитать значения проверочных вершин.

Будем называть этот алгоритм *алгоритмом инвертирования бита* или *алгоритмом Галлагера*, его первую часть — *фазой инициализации*, а вторую — *фазой декодирования*.

Докажем, что если $\mathcal{G} = \mathcal{G}^{\gamma,\delta}_{A,B,n}$ — расширитель с достаточно большим коэффициентом расширения, то этот алгоритм правильно декодирует любую комбинацию из не более чем $\frac{\delta n}{2}$ ошибок. Вначале покажем, что он всегда заканчивает свою работу.

Утверждение 5. *Алгоритм инвертирования бита заканчивает свою работу при любых начальных значениях кодовых вершин. Более того, число шагов его фазы декодирования не превосходит количества ненулевых синдромов, вычисленных на фазе инициализации.*

Доказательство. Значение каждой проверочной вершины равно сумме значений смежных с ней кодовых вершин. Поэтому сложность инициализации вершин доли R не превосходит числа рёбер в $\mathcal{G}$, равного An, то есть линейна.

Рассмотрим произвольный шаг фазы декодирования. Пусть y_j — плохая вершина, выбранная на этом шаге, $y'_j = \overline{y}_j$ — её значение после инвертирования, и пусть $\Gamma_\bullet = \Gamma_\bullet(y_j)$, $\Gamma_\circ = \Gamma_\circ(y_j)$ — чёрные и белые вершины из её окрестности непосредственно до рассматриваемого шага алгоритма, а $\Gamma'_\bullet$ и $\Gamma'_\circ$ — сразу после (см. рис. 5).

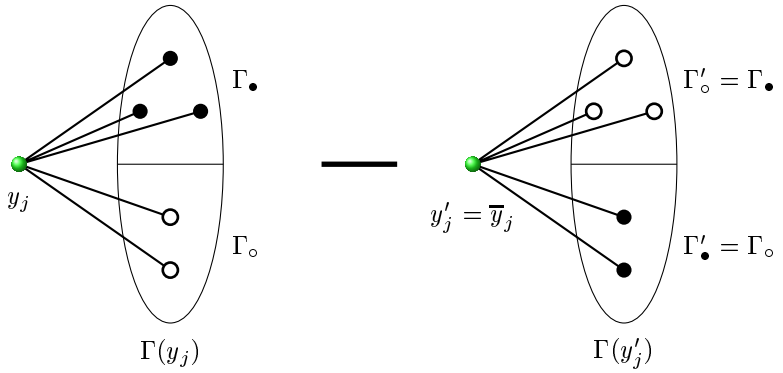

Рис. 5

Соответствующий выбранной вершине кодовый символ y_j входит в те и только в те проверочные суммы, которые отвечают синдромным вершинам из тени $\Gamma(y_j)$. Значит, после её инвертирования инвертируются все такие проверочные суммы и только они. Поэтому все вершины из $\Gamma(y_j)$ поменяют свой цвет на противоположный, а цвета остальных вершин, не смежных y_j, не изменятся.

Значения остальных кодовых вершин по построению остаются прежними. Так как $|\Gamma_\bullet| > |\Gamma_\circ|$, то $|\Gamma'_\bullet| < |\Gamma'_\circ|$ и выбранная вершина становится хорошей. Более того,

$$|R'_\bullet| = |R_\bullet| + |\Gamma'_\bullet| - |\Gamma'_\circ| < |R_\bullet|.$$

Таким образом, число ненулевых синдромов в правой доле уменьшается на каждом шаге. Точнее, если вначале их было m штук, то до своей остановки алгоритм сделает не более m шагов. □

Теперь убедимся, что наш алгоритм корректен.

Утверждение 6. *При условии $\gamma > \frac{3A}{4}$ алгоритм инвертирования бита корректно декодирует все слова, находящиеся от кодовых на расстоянии не более $\frac{\delta n}{2}$, то есть если применить его к любому такому слову y,*

что $d(x, y) \leqslant \frac{\delta n}{2}$*, где* x *— некоторое кодовое слово, то результат его работы совпадёт со словом* x*.*

Доказательство. Пусть $z = (z_1, \ldots, z_n)$ — вектор значений вершин левой доли графа $\mathcal{G}$ на очередном шаге алгоритма, то есть текущее приближение к x. Обозначим $S = \{z_i : z_i \neq x_i\}$ множество ошибочных разрядов, в которых различаются z и x. Тогда $\Gamma_\bullet(S) = R_\bullet$ и $\Gamma_1(S) \subseteq \Gamma_\bullet(S)$, то есть все вершины из $\Gamma_1(S)$ чёрные.

Предположим, что $|S| \leqslant \delta n$ (на первом шаге это выполнено по условию), тогда множество S должно хорошо расширяться, то есть $|\Gamma(S)| \geqslant \gamma \cdot |S|$. При $\gamma > \frac{3A}{4}$ из утверждения 3 получаем

$$|\Gamma_1(S)| \geqslant (2\gamma - A) \cdot |S| > \frac{A}{2}|S|. \tag{12}$$

Так как $|\Gamma(S)| \leqslant A \cdot |S|$, то $|R_\bullet| \geqslant |\Gamma_1(S)| > \frac{1}{2}|\Gamma(S)|$. Поэтому больше половины всех рёбер ведёт из множества S в чёрные вершины. Значит, в S найдётся плохая вершина и алгоритм не закончит работу до тех пор, пока не окажется, что $S = \varnothing$ или что к S неприменима оценка (12). Последнее может случиться только если коэффициент расширения множества S окажется меньше γ. Но граф $\mathcal{G}$ выбран так, что все множества мощности не более δn расширяются с коэффициентом не меньше γ, поэтому $|S| > \delta n$.

Итак, мы показали, что если алгоритм остановился, то это возможно только в двух случаях: либо $S = \varnothing$, либо $|S| > \delta n$. Покажем, что второго случая не бывает. Докажем даже более сильную оценку: на любом шаге справедливо неравенство

$$|S| < \delta n. \tag{13}$$

Для этого оценим число чёрных вершин на произвольном шаге.

С одной стороны, в начале работы алгоритма по условию $|S| = d(x, y) \leqslant \frac{\delta n}{2}$, поэтому на первом шаге (13) выполнено, а число чёрных вершин в R не превосходит $|\Gamma(S)| \leqslant A \cdot |S| \leqslant A \cdot \frac{\delta n}{2}$. Кроме того, выше мы убедились (утверждение 5), что число чёрных вершин с каждым шагом монотонно убывает. Следовательно, $|R_\bullet| \leqslant A \cdot \frac{\delta n}{2}$ на любом шаге.

С другой стороны, $|R_\bullet| \geqslant |\Gamma_1(S)| > \frac{A}{2}|S|$ при условии $|S| \leqslant \delta n$. Сравнивая нижнюю и верхнюю оценки $|R_\bullet|$, получаем неравенство $\frac{A}{2}|S| < A\frac{\delta n}{2}$, равносильное неравенству (13). Итак, для любого шага доказано:

$$\text{если } |S| \leqslant \delta n, \text{ то } |S| < \delta n. \tag{14}$$

Отметим, что это справедливо не только для множества S целиком, но и для любого его подмножества.

Предположим, что на некотором шаге $|S| \geqslant \delta n$. Без ограничения общности можно считать, что δn целое число. Рассмотрим какое-нибудь подмножество $S' \subseteq S$, такое, что $|S'| = \delta n$. Получаем противоречие, так как согласно (14) должно быть $|S'| < \delta n$.

Таким образом $|S| < \delta n$ на любом шаге, шагов конечное число, и значит по окончании работы алгоритма $S = \varnothing$. Утверждение доказано. □

Вход: Граф $\mathcal{G} = (L \sqcup R, E)$, вершины L содержат слово y.

Инициализация:

1. Перебрав список синдромов R, вычислить их;
2. $\overline{L} = L_0 = \ldots = L_A = \varnothing$;
3. для каждого $y \in L$ вычислить $m = |\Gamma_\bullet(y)|$;
   поместить y в L_m, и, если $m > \frac{1}{2}\deg y$, то и в $\overline{L}$.

Декодирование: до тех пор, пока $\overline{L} \neq \varnothing$:

4. найти наибольшее m, т.ч. $L_m \neq \varnothing$;
5. выбрать первую вершину y из L_m и инвертировать её,
   удалить y из $\overline{L}$ и перенести из L_m в $L_{\deg y - m}$;
6. для каждой вершины $s \in \Gamma(y)$ инвертировать её цвет;
7. для каждой вершины $y' \in \Gamma(\Gamma(y))$:
   - пересчитать $m' = |\Gamma_\bullet(y')|$;
   - переместить y' в $L_{m'}$, и, возможно, в $\overline{L}$.

Выход: если $L = L_0$, выдать L; иначе ошибка декодирования.

Рис. 6

В заключение дадим пример реализации алгоритма в модели RAM. Схема его программы приведена на рис. 6. Будем предполагать, что граф $\mathcal{G}$

представлен двумя связанными списками указателей на смежные вершины его долей: списком кодовых вершин L и списком синдромов R.

Поясним работу программы. На фазе инициализации программа вычисляет значения всех синдромов и организует вспомогательные множества $\overline{L}$ и $L_0, \ldots, L_A$. Все эти множества также как L и R удобно хранить в памяти в виде списков или очередей. Множество $\overline{L} \subset L$ содержит плохие вершины (точнее, указатели на них). Список

$$L_m = \{y \in L : |\Gamma_\bullet(y)| = m\}$$

состоит из тех кодовых вершин, у которых ровно m чёрных соседей. Так как степень кодовой вершины ограничена константой A, то каждая вершина из L войдёт ровно в одно из множеств L_m, $m = 0, 1, \ldots, A$. После фазы инициализации все синдромы из R разбиты на две части $R_\circ, R_\bullet$ и заполнены списки $\overline{L}$ и L_m.

На фазе декодирования программа в цикле находит «самую плохую» вершину y с максимальным числом чёрных синдромов и инвертирует её вместе со смежными синдромами. При этом для её корректной работы нужно переместить некоторые вершины из одних вспомогательных списков в другие, что и происходит в строках `5` и `7`. Программа завершает свою работу, когда список плохих вершин пуст. Если при этом в списке R остались чёрные синдромы, т.е. на последнем шаге $m \neq 0$, то произошла ошибка декодирования; в противном случае кодовые вершины содержат кодовое слово.

Утверждение 7. *Реализация алгоритма Галлагера в модели RAM имеет линейную сложность.*

Доказательство. Корректность программы, представленной на рис. 6, следует из утверждений 5 и 6. Оценим требуемые ею ресурсы — память и время. Так как $A = \mathcal{O}(1)$ и каждая вершина входит ровно в один из дополнительных списков $L_0, \ldots, L_A$, то объём используемой программой памяти линеен по n.

Инициализация всех списков (строка `3`) очевидно происходит за линейное время. Для просмотра текущего элемента любого списка и нахождения всех смежных ему вершин достаточно $\max(A, B) = \mathcal{O}(1)$ операций. Число шагов основного цикла (строки `4-7`) по утверждению 5 не превосходит $|R^0_\bullet| \leqslant |R| = \mathcal{O}(n)$. Поиск максимума в строке `4` выполняется за время $\mathcal{O}(A) = \mathcal{O}(1)$ и не зависит от n. Для строк `5,6` справедливо то же самое. Самой сложной частью одного шага цикла является строка `7`: раз

мы изменили синдромы s из тени вершины y, то это действие влияет на все кодовые вершины y', смежные s, и требует их коррекции. Тем не менее, число таких вершин y', равное $|\Gamma(\Gamma(y))|$, не превосходит $AB = \mathcal{O}(1)$, так как каждый синдром s смежен с не более чем B кодовыми вершинами[5)]. Значит, их просмотр и изменение статуса (для каждой вершины y' нужно заново пересчитать число чёрных соседей, всего $\mathcal{O}(A)$ операций) занимает константное время. Таким образом, число операций на каждом шаге 4-7 не зависит от n, и общее число операций, произведённых программой, равно $\mathcal{O}(n)$. □

Теперь мы можем сформулировать основной результат раздела. Он следует из утверждений 1, 7 и теоремы 2.

Теорема 4 (М. Сипсер, Д. Спайлман). *Пусть $\mathcal{G} = \mathcal{G}^{\gamma,\delta}_{A,B,n}$ — двудольный расширитель, такой что $\gamma > \frac{3A}{4}$ и $A, B = \mathcal{O}(1)$. Пусть $\mathcal{C} = \mathcal{C}^{\gamma,\delta}_n$ — линейный код, для которого граф $\mathcal{G}$ является графом Таннера. Тогда скорость кода $\mathcal{C}$ не меньше $\frac{1}{2}$, а относительное кодовое расстояние не меньше δ. Более того, код $\mathcal{C}$ обладает линейным алгоритмом декодирования и квадратичным алгоритмом кодирования, т.е. $L_e(\mathcal{C}) = \mathcal{O}(n^2)$ и $L_d(\mathcal{C}) = \mathcal{O}(n)$.*

5. Анализ работы алгоритма при неверных проверках

Вернёмся к алгоритму декодирования экспандерного кода (стр. 39) и поставим вопрос, как он будет работать в случае когда испорчены не только кодовые, но и проверочные вершины. Действительно, пусть, как и ранее, $\mathcal{G} = (L \sqcup R, E)$ — двудольный (A, B)-ограниченный (γ, δ)-расширитель и $|L| = n$. Вершины левой доли содержат искажённое слово y, отличающееся от некоторого кодового слова x не более чем в $\delta n/2$ ошибочных разрядах. Проверочные вершины (правая доля) соответствуют компонентам синдрома принятого слова y. Будем как и прежде считать, что они раскрашены белым и чёрным цветом в зависимости от того, равен соответствующий синдромный разряд нулю или единице. Предположим в отличие от ранее рассмотренного случая, что в графе $\mathcal{G}$ в некоторых синдромных вершинах могут, как и в кодовых, произойти ошибки: испорченная проверочная вершина графа $\mathcal{G}$ всегда принимает значение, противоположное истинному.

[5)] Именно в этом месте и только здесь нам требуется ограниченность степени B вершин правой доли графа $\mathcal{G}$.

Такая вершина даёт неверные ответы на запросы на протяжении всей работы алгоритма — если соответствующая компонента синдрома на самом деле равна нулю, то есть цвет вершины должен быть белым, то она чёрная, и наоборот.

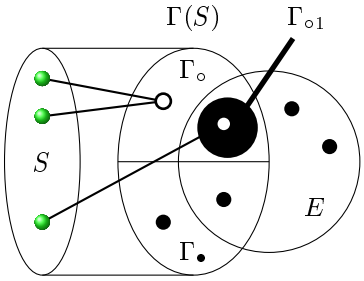

Рис. 7

Запустим алгоритм на слове y. Ясно, что если число испорченных проверочных вершин велико, то он не сможет декодировать все ошибки в принятом слове. Постараемся выяснить, при каких условиях ему наверняка удастся их значительно уменьшить. Итак, пусть $S \subset L$ и $E \subset R$ — множества ошибок-инверсий в кодовых и проверочных вершинах соответственно, и пусть $e = |E|$. Пусть, как и ранее, $\Gamma_\circ = \Gamma_\circ(S)$ и $\Gamma_\bullet = \Gamma_\bullet(S)$ — множества белых и чёрных проверочных вершин из тени $\Gamma = \Gamma(S)$. Предположим, что $\Gamma(S) \cap E \neq \varnothing$. Рассмотрим множество $\Gamma_{\circ 1} = \Gamma_\circ \cap \Gamma_1$, состоящее из белых вершин, смежных ровно с одной вершиной из S. Так как они белого цвета, а должны быть чёрными, то очевидно, что они ошибочные, то есть $\Gamma_{\circ 1} \subseteq E$ и $|\Gamma_{\circ 1}| \leqslant e$. Подсчитаем число рёбер, соединяющих S с его тенью $\Gamma(S)$, оценив их число сверху и снизу. Любая вершина из множества $\Gamma_\circ \setminus \Gamma_{\circ 1}$ соединена с S по крайней мере двумя рёбрами, и поэтому

$$\begin{aligned} A \cdot |S| &\geqslant |\Gamma| + |\Gamma_\circ \setminus \Gamma_{\circ 1}| \geqslant |\Gamma| + |\Gamma_\circ| - e = \\ &= 2|\Gamma| - |\Gamma_\bullet| - e \geqslant 2\gamma \cdot |S| - |\Gamma_\bullet| - e. \end{aligned} \tag{15}$$

Множество S является множеством разрядов, в которых различаются принятое и исходное кодовое слово. Посмотрим, как меняется S в процессе работы алгоритма. Если он окончил работу, то всякая вершина из S имеет не более $\frac{A}{2}$ чёрных соседей, то есть $|\Gamma_\bullet| \leqslant \frac{A}{2}|S|$, иначе можно сделать ещё один шаг. Из неравенства (15) тогда получаем

$$A \cdot |S| \geqslant 2\gamma \cdot |S| - \frac{A}{2}|S| - e,$$

откуда

$$|S| \leqslant \frac{e}{2\gamma - \frac{3A}{2}}.$$

Потребуем, чтобы по окончании работы алгоритма всегда выполнялось неравенство $|S| \leqslant e/2$. Это будет так при условии

$$\frac{e}{2\gamma - \frac{3A}{2}} \leqslant \frac{e}{2},$$

для чего необходимо $\gamma \geqslant 1 + \frac{3A}{4}$. Мы видим, что для эффективной работы нашего алгоритма в новых условиях от расширителя требуется значительно больший коэффициент расширения, чем при работе с безошибочными проверками. Отметим, что при малых значениях A, а именно при $A \leqslant 4$, таких графов не существует вовсе. Тем не менее, например при $A \geqslant 8$ для наших целей достаточно взять $\gamma \geqslant \frac{7A}{8}$, и существование таких расширителей гарантировано теоремой 3.

Выясним теперь, при каких дополнительных условиях работа алгоритма с испорченными проверками корректна и он остановится при $|S| \leqslant e/2$. Во-первых, алгоритм всегда окончит свою работу, даже независимо от величины $|E|$. Это становится очевидным как только мы заметим (см. утверждение 5), что с каждым шагом число чёрных вершин в R убывает, а значит число шагов конечно и не превосходит $|R| = n/2$. Во-вторых, отсюда следует, что сложность алгоритма не более чем линейна по n, так как число операций на одном шаге не зависит от n.

Пусть S^i — множество ошибочных вершин из L на i-ом шаге, соответствующих разрядам в которых различаются кодовое слово и вычисленное алгоритмом его текущее приближение. Нам осталось показать, что множество S^i достаточно мало при всех i для того чтобы «хорошо расширяться» с коэффициентом γ и чтобы к нему можно было применить всё сказанное выше. Мы сделаем это аналогично тому, как доказали утверждение 6.

Действительно, пусть неравенства $|S^0| \leqslant \frac{\delta n}{2}$ и $e = |E| \leqslant \frac{\delta n}{2}$ выполнены к первому шагу алгоритма. Тогда в начале его работы число чёрных проверочных вершин не превосходит $A \cdot |S| + |E| \leqslant \frac{\delta n}{2}(A+1)$. Обозначим через $R^i_\bullet$ множество вершин чёрного цвета в R на i-ом шаге. Очевидно, что $|R^i_\bullet| \geqslant |\Gamma_1(S^i)| - |\Gamma_{\circ 1}(S^i)|$. В процессе работы множество E не меняется, а $|R^i_\bullet|$ монотонно убывает. Кроме того, в силу свойства (9)

$$|R^i_\bullet| \geqslant |\Gamma_1(S^i)| - e \geqslant (2\gamma - A) \cdot |S^i| - e$$

при условии $|S^i| \leqslant \delta n$, откуда

$$(2\gamma - A) \cdot |S^i| \leqslant |R^i_\bullet| + e \leqslant \frac{\delta n}{2}(A+1) + e \leqslant \frac{\delta n}{2}(A+2),$$

то есть

$$|S^i| \leqslant \delta n \cdot \frac{A+2}{4\gamma - 2A}. \tag{16}$$

Заметим, что на каждом шаге величина $|S^i|$ меняется на 1. Поэтому если мы потребуем, чтобы из неравенства (16) следовало

$$|S^i| \leqslant \delta n - 1, \tag{17}$$

то к следующему шагу $|S^{i+1}| \leqslant |S^i| + 1 \leqslant \delta n$ и мощность множества S^{i+1} не переходит через границу «хорошего расширения». Следовательно, и на $(i+1)$-ом шаге все оценки остаются в силе, в частности $|S^{i+1}| \leqslant \delta n - 1$. Значит, чтобы применить индукцию по i, которая завершит доказательство корректности, нам достаточно выполнения неравенства

$$\delta n \frac{A+2}{4\gamma - 2A} < \delta n - 1.$$

Оно равносильно неравенству

$$\gamma > \frac{1}{2} \cdot \frac{\delta n}{\delta n - 1} + A\left(\frac{1}{2} + \frac{1}{4} \cdot \frac{\delta n}{\delta n - 1}\right). \tag{18}$$

Мы рассматриваем достаточно большие кодовые графы, так чтобы отношение $\frac{\delta n}{\delta n - 1}$ было невелико. Например, при $\frac{\delta n}{\delta n - 1} < \frac{6}{5}$ и $A \geqslant 8$ неравенство (18) является более слабым, чем требование $\gamma \geqslant 1 + \frac{3A}{4}$. Следовательно, при этих условиях выполнено (17) и алгоритм работает корректно.

Таким образом, мы доказали следующее утверждение.

Утверждение 8. *Пусть $\mathcal{C}$ — экспандерный код длины n и размерности не менее $n/2$, построенный по двудольному расширителю $\mathcal{G} = \mathcal{G}^{\gamma,\delta}_{A,B,n}$, причём $n \geqslant n_0 = \frac{6}{\delta}$, $A \geqslant 8$ и $\gamma \geqslant \frac{7A}{8}$. Пусть $x \in \mathcal{C}$ — кодовое слово и y — произвольное слово, отличающееся от x не более чем в $\frac{\delta n}{2}$ разрядах. Пусть на вход алгоритма инвертирования бита подаются граф $\mathcal{G}$ и слово y, причём $e \leqslant \frac{\delta n}{2}$ проверочных вершин в $\mathcal{G}$ испорчены, так что их значение всегда противоположно истинному. Тогда алгоритм закончит работу за не более чем линейное по n число шагов, затратив $\mathcal{O}(n)$ элементарных операций, и найденное им слово будет отличаться от слова x не более чем в $e/2$ разрядах.*

6. Код, редуцирующий ошибки

Раньше мы считали, что кодовым словом является такое слово $x = (x_1, \ldots, x_n)$, что при $x_i \in L$ выполнено $s_m = 0$ для всех $s_m \in R$. Теперь рассмотрим другой способ построения кода с помощью графа-расширителя: будем считать кодовыми все вершины графа, а не только вершины левой доли.

Пусть $\mathcal{G}$ — двудольный расширитель с левой долей из k вершин[6)] и правой долей из $k/2$ вершин. Его коэффициенты γ и δ уточним позднее. Рассмотрим двоичный линейный код $\mathcal{R}_k$, заданный порождающей матрицей вида

$$G = G_{k \times \frac{3}{2}k} = (E|B)\,,$$

где $E = E_{k \times k}$ — единичная матрица, $B = B_{k \times \frac{k}{2}}$ — матрица смежности графа $\mathcal{G}$, то есть матрица, в которой на пересечении i-й строки и j-го столбца стоит единица, когда i-я вершина левой доли соединена ребром в графе $\mathcal{G}$ с j-й вершиной правой доли, и стоит ноль в противном случае. Кодовые слова кода $\mathcal{R}_k$ имеют вид

$$(x_1, \ldots, x_k, s_1, \ldots, s_{k/2}),$$

где вектор $x = (x_1, \ldots, x_k)$ произволен, а набор $s = (s_1, \ldots, s_{k/2})$ является значением синдрома вектора x в прежнем смысле. Таким образом, первые k разрядов кодового слова — информационные, последние $k/2$ разрядов — проверочные, и код $\mathcal{R}_k$ является систематическим.

Выясним, какова корректирующая способность кода $\mathcal{R}_k$. Его размерность равна k, длина равна $\frac{3}{2}k$. Если степень вершин из левой и правой долей графа $\mathcal{G}$ не превышает соответственно A и B, то в каждой строке матрицы G не более $(A+1)$ единиц, и не более B единиц в каждом из последних столбцов. Значит, матрица G является разреженной и порождаемый ею линейный код имеет кодовое расстояние $\mathcal{O}(1)$, не зависящее от k. Таким образом код $\mathcal{R}_k$ не является асимптотически хорошим, так как плохо исправляет ошибки. Однако он хорошо их редуцирует.

Утверждение 9. *Пусть $\boldsymbol{x} = (x, s) \in \mathcal{R}_k$ — кодовое слово, соответствующее информационному вектору x. Пусть получено слово $\boldsymbol{y}$, отличное от слова $\boldsymbol{x}$ не более чем в $\frac{\delta k}{2}$ разрядах, среди которых e проверочных. Тогда слово $\boldsymbol{z}$, полученное на выходе алгоритма инвертирования бита применённого к $\boldsymbol{y}$, отличается от $\boldsymbol{x}$ не более чем в $\frac{e}{2}$ информационных разрядах. Более того, число шагов алгоритма не более чем линейно.*

[6)] Мы специально поменяли обозначения чтобы подчеркнуть, что то, что в прежнем коде было длиной, в новом коде стало размерностью.

Доказательство. Прежде всего поймём, как можно применить алгоритм декодирования кода Сипсера — Спайлмана к коду $\mathcal{R}_k$. Пусть векторы y', y'' составлены из первых k и, соответственно, последних $k/2$ разрядов слова $\boldsymbol{y}$. Рассмотрим граф $\mathcal{G}$, на основе которого построен код $\mathcal{R}_k$. Заменим фазу инициализации алгоритма инвертирования бита, на которой проверочным вершинам правой доли $\mathcal{G}$ первый раз приписываются некоторые значения, следующей процедурой: разряды векторов y' и y'' помещаются в вершины левой и правой доли $\mathcal{G}$. После этого алгоритм продолжает свою работу как обычно.

Нетрудно видеть, что работа алгоритма именно при таких начальных условиях была разобрана в предыдущем разделе. Теперь искомый результат следует из утверждения 8. Заметим, что утверждение 8 позволяет немного усилить формулировку утверждения 9: оно будет справедливо и в том случае, когда слово $\boldsymbol{y}$ отличается от $\boldsymbol{x}$ не более чем в $\frac{\delta k}{2}$ информационных и не более чем в e проверочных разрядах, где $e \leqslant \frac{\delta k}{2}$. □

Наконец отметим, что сложность кодирования кодом $\mathcal{R}_k$, то есть число элементарных битовых операций, требующихся для получения кодового вектора из информационного, равна числу рёбер в графе $\mathcal{G}$, то есть равна $\mathcal{O}(k)$. Это следует также и из утверждения 1.

Утверждение 10. $L_e(\mathcal{R}_k) = \mathcal{O}(k)$.

Будем называть код $\mathcal{R}_k$ кодом, *редуцирующим ошибки*.

7. Код Спайлмана

Отметим, что если $e = 0$ в утверждении 9, то есть число ошибок в слове $\boldsymbol{y}$ не превышает $\frac{\delta k}{2}$ и все они сосредоточены в информационной части, то алгоритм декодирования кода $\mathcal{R}_k$ исправит все такие ошибки. Отсюда возникает идея: надо защитить проверочные символы от ошибок, тогда информационные будут декодированы безошибочно. Это можно сделать, применив следующую рекурсивную конструкцию.

Сначала изложим неформально общую схему. Пусть $k = 2^a b$, где a, b — некоторые натуральные числа. Обозначим через $\mathcal{S}_k$ код, все кодовые слова которого имеют вид, приведённый на рис. 8.

Каждое кодовое слово кода $\mathcal{S}_k$ состоит из четырёх частей M_k, A_k, B_k и D_k, длины которых равны k, $k/2$, $3k/2$ и k соответственно. Разряды множества M_k выбираются произвольно. Они кодируются вышеописанным кодом $\mathcal{R}_k$, и полученные $k/2$ его проверочных разрядов обозначаются A_k. Далее по индукции разряды A_k кодируются кодом $\mathcal{S}_{k/2}$, проверочные символы

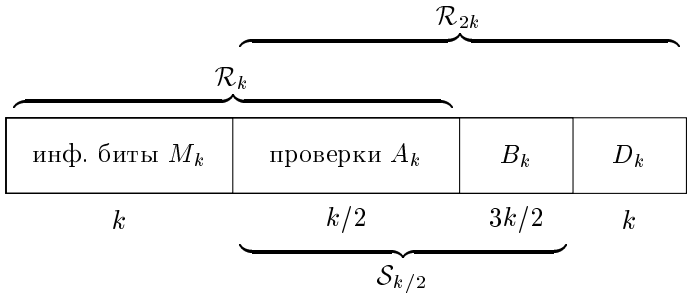

Рис. 8

которого обозначены B_k. Наконец, ко всему слову $(A_k, B_k) \in \mathcal{S}_{k/2}$ длины $k/2 + 3k/2 = 2k$ применяется кодер кода $\mathcal{R}_{2k}$, и вычисленные им проверочные символы обозначаются D_k. На рисунке отмечены соответствующие кодовые слова (M_k, A_k), (A_k, B_k) и (A_k, B_k, D_k) кодов $\mathcal{R}_k$, $\mathcal{S}_{k/2}$ и $\mathcal{R}_{2k}$, которые являются подсловами кодового слова из $\mathcal{S}_k$.

Таким образом, M_k — информационные разряды кода $\mathcal{S}_k$, (A_k, B_k, D_k) — проверочные, длина кода $\mathcal{S}_k$ равна $4k$ и скорость равна $\frac{1}{4}$. Код $\mathcal{S}_k$ был впервые предложен в работе [18]. Назовём его *кодом Спайлмана*.

Рассмотрим вложенную последовательность кодов, использованных при построении кода $\mathcal{S}_k$:

$$\mathcal{S}_k, \quad \mathcal{S}_{k/2}, \quad \mathcal{S}_{k/2^2}, \quad \mathcal{S}_{k/2^3}, \quad \ldots \quad \mathcal{S}_{2b}, \quad \mathcal{S}_b. \tag{19}$$

На некотором шаге с номером a она оборвётся, и $\mathcal{S}_b$ будет «затравочным» кодом для этой последовательности. Уточним его параметры, но прежде чем выбрать подходящие a, b и $\mathcal{S}_b$ выясним, с какой сложностью происходит кодирование и декодирование такой каскадной конструкции.

Заметим, что $B_k = (A_{k/2}, B_{k/2}, D_{k/2})$ и поэтому кодовое слово x кода $\mathcal{S}_k$ имеет вид

$$x = (M_k, A_k, A_{k/2}, A_{k/4}, \ldots, A_{2b}, C_b, D_{2b}, \ldots, D_{k/2}, D_k),$$

где C_b — некоторое слово из кода $\mathcal{S}_b$.

Сложность кодирования. По построению, кодирование осуществляется рекурсивно в три этапа:

$$M_k \xrightarrow{\mathcal{R}_k} (M_k, A_k) \xrightarrow{\mathcal{S}_{k/2}} (M_k, A_k, B_k) \xrightarrow{\mathcal{R}_{2k}} (M_k, A_k, B_k, D_k).$$

Следовательно, для сложности кодирования кодом Спайлмана справедливо равенство

$$L_e(\mathcal{S}_k) = L_e(\mathcal{R}_k) + L_e(\mathcal{S}_{k/2}) + L_e(\mathcal{R}_{2k}).$$

В утверждении 10 предыдущего раздела было показано, что $L_e(\mathcal{R}_k) \leqslant c_e k$ для некоторой константы c_e. Отсюда получаем рекуррентное неравенство

$$L_e(\mathcal{S}_k) \leqslant L_e(\mathcal{S}_{k/2}) + 3c_e k,$$

решением которого будет

$$L_e(\mathcal{S}_k) \leqslant L_e(\mathcal{S}_b) + 3c_e \sum_{i=1}^{a} \frac{k}{2^i} < L_e(\mathcal{S}_b) + 6c_e k = L_e(\mathcal{S}_b) + \mathcal{O}(k). \tag{20}$$

Таким образом, выбором b можно добиться линейной сложности кодера.

Сложность декодирования. Декодирование осуществляется в обратном порядке. Пусть передавалось слово $x = (M_k, A_k, B_k, D_k) \in \mathcal{S}_k$, а получено некоторое слово $y = (M'_k, A'_k, B'_k, D'_k)$. Вначале разряды (A'_k, B'_k, D'_k) поступают в декодер кода $\mathcal{R}_{2k}$. Он исправляет некоторые ошибки и выдаёт некоторый вектор (A''_k, B''_k), который, возможно, отличается от (A_k, B_k). Затем применяется шаг индукции: вектор (A''_k, B''_k) подаётся на вход декодера кода $\mathcal{S}_{k/2}$. На его выходе мы получаем некоторый вектор A'''_k. Завершается декодирование слова y применением декодера кода $\mathcal{R}_k$, дающего нам вектор M''_k:

$$(M'_k, A'_k, B'_k, D'_k) \xrightarrow{\mathcal{R}_{2k}^{-1}} (M'_k, A''_k, B''_k) \xrightarrow{\mathcal{S}_{k/2}^{-1}} (M'_k, A'''_k) \xrightarrow{\mathcal{R}_k^{-1}} M''_k. \tag{21}$$

Таким образом, декодер кода $\mathcal{S}_k$ состоит из декодеров кодов $\mathcal{R}_{2k}$, $\mathcal{S}_{k/2}$ и $\mathcal{R}_k$, и поэтому

$$L_d(\mathcal{S}_k) = L_d(\mathcal{R}_{2k}) + L_d(\mathcal{S}_{k/2}) + L_d(\mathcal{R}_k).$$

Как мы видели выше (утверждение 9), $L_d(\mathcal{R}_k) \leqslant c_d k$ для некоторой константы c_d. Отсюда аналогично (20) получаем

$$L_d(\mathcal{S}_k) \leqslant L_d(\mathcal{S}_b) + 6c_d k = L_d(\mathcal{S}_b) + \mathcal{O}(k). \tag{22}$$

Значит, при условии $L_d(\mathcal{S}_b) = \mathcal{O}(k)$ декодер также как и кодер имеет линейную сложность. В итоге мы доказали следующее

Утверждение 11. *Код Спайлмана имеет линейную сложность кодирования и декодирования.*

Корректность. Теперь настало время убедиться, что код $\mathcal{S}_k$ исправляет некоторую положительную долю ошибок, а значит асимптотически хорош. Докажем, что если число ошибок не превосходит $\frac{\delta k}{2}$, то вектор M_k'', полученный на выходе каскадного декодера (21), совпадает с исходным информационным вектором M_k.

Утверждение 12. *Если код $\mathcal{S}_i$ исправляет $\frac{\delta i}{2}$ ошибок, то код $\mathcal{S}_{2i}$ исправляет δi ошибок.*

Доказательство. Пусть принятое слово $y = (M'_{2i}, A'_{2i}, B'_{2i}, D'_{2i})$ отличается от кодового слова $x = (M_{2i}, A_{2i}, B_{2i}, D_{2i})$ кода $\mathcal{S}_{2i}$ не более чем в δi разрядах. Так как

$$(A''_{2i}, B''_{2i}) = \mathcal{R}_{4i}^{-1}(A'_{2i}, B'_{2i}, D'_{2i})$$

и расстояние между вектором $(A_{2i}, B_{2i}, D_{2i}) \in \mathcal{R}_{4i}$ и вектором $(A'_{2i}, B'_{2i}, D'_{2i})$ не превосходит δi, то, согласно утверждению 9, полученный на выходе декодера кода $\mathcal{R}_{4i}$ вектор (A''_{2i}, B''_{2i}) отличается от вектора $(A_{2i}, B_{2i}) \in \mathcal{S}_i$ не более чем в $\frac{\delta i}{2}$ разрядах.

Тогда, согласно нашему предположению, декодер кода $\mathcal{S}_i$ должен исправить все ошибки в слове (A''_{2i}, B''_{2i}), поэтому

$$A_{2i} = \mathcal{S}_i^{-1}(A''_{2i}, B''_{2i}).$$

Наконец, так как проверочные разряды A_{2i} не содержат ошибок и к тому же $d(M'_{2i}, M_{2i}) \leqslant \delta i$, то результатом декодера $\mathcal{R}_{2i}$, применённого к (M'_{2i}, A_{2i}), будет исходный информационный вектор: $M_{2i} = \mathcal{R}_{2i}^{-1}(M'_{2i}, A_{2i})$. □

Для применения индукции нам осталось подобрать код $\mathcal{S}_b$. Мы доказали, что $\mathcal{S}_k$ исправляет любые $\frac{\delta k}{2}$ ошибок, если $\mathcal{S}_{k/2}$ исправит любые $\frac{\delta k}{2^2}$ ошибок. В свою очередь, код $\mathcal{S}_{k/2}$ исправит $\frac{\delta k}{2^2}$ ошибок, если $\mathcal{S}_{k/2^2}$ исправит $\frac{\delta k}{2^3}$ ошибок, и т.д. Значит, наша конструкция будет корректно работать, если код $\mathcal{S}_b$ исправит произвольную комбинацию из $\frac{\delta b}{2}$ ошибок.

На вопрос о существовании кода длиной $4b$, скорости $1/4$, с кодовым расстоянием $d \geqslant \delta b + 1$ можно ответить, применив неравенство Варшамова — Гильберта (1): искомый код $\mathcal{S}_b$ существует, если

$$\sum_{i=0}^{d-2} \binom{4b-1}{i} < 2^{3b}.$$

Это неравенство будет выполнено, если $\sum_{i=0}^{\delta b} \binom{4b}{i} < 2^{3b}$, что в свою очередь будет верно (здесь мы воспользуемся (11)), если будет справедливо

неравенство $\left(\frac{e\cdot 4b}{\delta b}\right)^{\delta b} < 2^{3b}$. Последнее при $b > 1$ равносильно неравенству

$$\left(\frac{e\cdot 4}{\delta}\right)^{\delta} < 8,$$

которое выполняется при всех $\delta < \frac{1}{2}$.

Таким образом, при достаточно малом значении δ на роль кода $\mathcal{S}_b$ нам подходит практически любой лежащий на границе Варшамова — Гильберта код, лишь бы сложность его декодирования не превышала $\mathcal{O}(k)$. Кроме того, вдобавок ещё требуется существование расширителя с указанными в утверждении 8 параметрами, для того чтобы построить код $\mathcal{R}_{2b}$, а затем и всю последовательность (19). Это условие будет выполнено, например, при

$$b = \left\lceil \frac{6}{\delta} \right\rceil = \mathcal{O}(1). \tag{23}$$

Другой способ построения кода Спайлмана состоит в выборе в качестве начального кода $\mathcal{S}_b$ для последовательности (19) подходящего кода Сипсера — Спайлмана $\mathcal{C}_b = \mathcal{C}_b^{\gamma,\delta}$ из раздела 2. Согласно теореме 4 он исправляет требуемое количество ошибок (не менее $\frac{\delta b}{2}$), причём $L_e(\mathcal{C}_b) = \mathcal{O}(b^2)$ и $L_d(\mathcal{C}_b) = \mathcal{O}(b)$. Чтобы скомпенсировать его квадратичную сложность кодирования, выберем $b = \mathcal{O}(\sqrt{k})$, и тогда итоговый код Спайлмана $\mathcal{S}_k$ будет иметь линейную по k сложность кодирования и декодирования согласно оценкам (20) и (22). Как и прежде, существование требуемого для кода $\mathcal{C}_b$ расширителя с необходимыми параметрами гарантирует теорема 3.

Мы решили, таким образом, нашу главную задачу — построили хороший код с простыми кодером и декодером. Это код $\mathcal{S}_k$.

Теорема 5 (Д. Спайлман). *Существуют асимптотически хорошие коды с линейной сложностью кодирования/декодирования.*

Список литературы

1. *Алон Н., Спенсер Дж.* Вероятностный метод. — М.: БИНОМ. Лаборатория знаний, 2007.
2. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979.
3. *Бассалыго Л. А., Пинскер М. С.* О сложности оптимальной неблокирующей коммутационной схемы без перестроения // Проблемы передачи информации. — 1973. — Т. 9, вып. 1. — С. 84–87.

4. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. — М.: Мир, 1986.

5. *Галлагер Р. Г.* Коды с малой плотностью проверок на четность. — М.: Мир, 1966.

6. *Гашков С. Б.* Графы-расширители и их применения в теории кодирования // Сб. "Математическое Просвещение". — 2009. — Вып. 13. — С. 104–126.

7. *Зяблов В. В., Пинскер М. С.* Оценка сложности исправления ошибок низкоплотностными кодами Галлагера // Проблемы передачи информации. — 1975. — Т. 11, вып. 1. — С. 23–36.

8. *Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А.* Теория кодов, исправляющих ошибки. — М.: Связь, 1979.

9. *Маргулис Г. А.* Явные конструкции расширителей // Проблемы передачи информации. — 1973. — Т. 9, вып. 4. — С. 71–80.

10. *Berlekamp E., McEliece R., van Tilborg H.* On the inherent intractability of certain coding problems // IEEE Transactions on Information Theory. — 1978. — Vol. 24, № 3. — P. 384–386.

11. *Capalbo M. R., Reingold O., Vadhan S. P., Wigderson A.* Randomness conductors and constant-degree lossless expanders // Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC). — 2002. — P. 659-668.

12. *Guruswami V., Indyk P.* Linear time encodable/decodable codes with near optimal rate // IEEE Transactions on Information Theory. — 2005. — Vol. 51, № 10. — P. 3393–3400.

13. *Hoory S., Linial N., Wigderson A.* Expander graphs and their applications // Bulletin of the AMS. — 2006. — Vol. 43, № 4. — P. 439–561.

14. *Kahale N.* On the second eigenvalue and linear expansion of regular graphs // Proc. 33rd Annual Symposium on Foundations of Computer Science (SFCS). — 1992. — P. 296–303.

15. *Pinkser M. S.* On the complexity of a concentrator // 7th International Teletraffic Conference. — 1973. — P. 318/1–318/4.

16. *Roth R., Skachek V.* Improved nearly-MDS expander codes // IEEE Transactions on Information Theory. — 2006. — Vol. 52, № 7. — P. 3186–3197.

17. *Sipser M., Spielman D.* Expander codes // IEEE Transactions on Information Theory. — 1996. — Vol. 42, № 6. — P. 1710–1722.

18. *Spielman D.* Linear-time encodable and decodable error-correcting codes // IEEE Transactions on Information Theory. — 1996. — Vol. 42, № 6. — P. 1723–1732.

19. *Tanner R. M.* A recursive approach to low complexity codes // IEEE Transactions on Information Theory. — 1981. — Vol. 27, № 5. — P. 533–547.

20. *Zemor G.* On expander codes // IEEE Transactions on Information Theory. — 2001. — Vol. 47, № 2. — P. 835–837.

СОВЕРШЕННОЕ ЛИНЕЙНОЕ ХЕШИРОВАНИЕ В БУЛЕВОМ КУБЕ

А. В. Чашкин

Московский государственный университет им. М. В. Ломоносова, механико-математический факультет, 119992 Москва, Ленинские горы

e-mail: chashkin@inbox.ru

Пусть D — произвольная область в n-мерном булевом кубе $\{0,1\}^n$. Линейным хешированием области D называется линейное отображение этой области в булев куб меньшей размерности. Линейное хеширование называется совершенным, если задающее его отображение является инъективным на D. Совершенное линейное хеширование естественным образом возникает в теории кодирования, теории сложности, при построении словарей, в задачах сжатия и поиска информации. Видимо первым явным примером совершенного линейного хеширования в дискретной математике стал построенный Хеммингом код, исправляющий однократные ошибки. Проверочная матрица $(2^n-1,n)$-кода Хемминга является матрицей инъективного на (2^n-1)-мерном единичном шаре линейного оператора, который отображает этот шар в n-мерное булево пространство. Как правило, при использовании совершенного линейного хеширования требуется сделать как можно меньшими ранг и сложность соответствующего линейного оператора. Об этих двух сторонах (о рангах и сложности) инъективных операторов и будет рассказано ниже.

1. Основное свойство

Большинство утверждений об инъективных на заданной области линейных операторах основаны на следующем простом факте: *линейный оператор f инъективен на области D тогда и только тогда, когда его ядро не*

пересекается с множеством $D^\star$ попарных сумм элементов области D. Действительно, если линейный оператор f инъективно действует на области D, т. е. $f(\boldsymbol{x}_i) \neq f(\boldsymbol{x}_j)$ для любых $\boldsymbol{x}_i$ и $\boldsymbol{x}_j$ из D, то

$$f(\boldsymbol{x}_i \oplus \boldsymbol{x}_j) = f(\boldsymbol{x}_i) \oplus f(\boldsymbol{x}_j) \neq \mathbf{0}. \tag{24}$$

Следовательно, $\boldsymbol{x}_i \oplus \boldsymbol{x}_j \notin \ker f$. Поэтому из (24) следует, что множество $D^\star$ и ядро оператора f не пересекаются. Легко видеть, что верно и обратное: если множество $D^\star$ и подпространство $\mathbb{H} \subseteq \{0,1\}^n$ не пересекаются, то $\mathbb{H}$ является ядром линейного оператора, отображающего несовпадающие наборы области D в несовпадающие наборы ее образа. Рассмотрим подпространство $\mathbb{H}$, не имеющее общих наборов с $D^\star$, и линейный оператор f, ядром которого является $\mathbb{H}$. Пусть $\boldsymbol{x}_i$ и $\boldsymbol{x}_j$ — произвольные наборы из D. Так как $\boldsymbol{x}_i \oplus \boldsymbol{x}_j \notin \mathbb{H} = \ker f$, то

$$f(\boldsymbol{x}_i) \oplus f(\boldsymbol{x}_j) = f(\boldsymbol{x}_i \oplus \boldsymbol{x}_j) \neq \mathbf{0},$$

т. е. образы наборов $\boldsymbol{x}_i$ и $\boldsymbol{x}_j$ различны.

2. Верхняя оценка ранга

Имеет место следующая верхняя оценка на ранг инъективного на произвольном множестве линейного оператора.

Теорема 1. *Для любой области $D \subseteq \{0,1\}^n$, состоящей не более чем из $\sqrt{2^n}$ наборов, найдется инъективный на этой области линейный (m,n)-оператор, для числа компонент которого справедливо неравенство*

$$m \leqslant \lfloor 2 \log_2 |D| \rfloor - 1.$$

Теорема 1 является простым следствием доказываемой далее теоремы 2.

Теорема 2. *Пусть для множества $D^\star$ попарных сумм элементов области $D \subseteq \{0,1\}^n$ справедливо неравенство*

$$2^{m+1} > |D^\star| + 1.$$

Тогда существует инъективный на области D линейный (m,n)-оператор.

Доказательство теоремы 2 основано на последовательном применении ее частного случая — доказываемой ниже леммы.

Лемма 1. *Пусть для множества $D^\star$ попарных сумм элементов области $D \subseteq \{0,1\}^n$ справедливо неравенство*

$$2^n > |D^\star| + 1.$$

Тогда существует инъективный на области D линейный $(n-1,n)$-оператор.

Доказательство. Для построения требуемого линейного оператора достаточно найти в $\{0,1\}^n$ подпространство $\mathbb{H}$, которое не пересекается с множеством $D^\star$ и размерность которого равна единице. Существование такого пространства легко следует из условий леммы. Так как $2^n > |D^\star| + 1$, то среди элементов $\{0,1\}^n$ найдется ненулевой набор $\boldsymbol{y}$, не принадлежащий $D^\star$, который вместе с нулевым набором будет образовывать требуемое одномерное подпространство. Лемма доказана.

Доказательство теоремы 2. Воспользуемся леммой 1. Из этой леммы следует существование такого линейного $(n-1,n)$-оператора f_1, что $f_1(\boldsymbol{x}) \neq f_1(\boldsymbol{y})$ для любых неравных наборов $\boldsymbol{x}$ и $\boldsymbol{y}$ из D. Далее для множества D будем использовать обозначение D_0. Через D_1 обозначим образ области D_0 при действии f_1. Легко видеть, что мощность множества $D_1^\star$, состоящего из попарных сумм различных элементов множества D_1, не превосходит мощности множества $D_0^\star$. Действительно, если это не так, то в D_0 должны присутствовать такие наборы $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ и $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, что $\boldsymbol{x}_1 \oplus \boldsymbol{x}_2 = \boldsymbol{y}_1 \oplus \boldsymbol{y}_2$ и $f_1(\boldsymbol{x}_1) \oplus f_1(\boldsymbol{x}_2) \neq f_1(\boldsymbol{y}_1) \oplus f_1(\boldsymbol{y}_2)$. Однако очевидно, что только одно из этих соотношений может быть справедливым.

Если $2^{n-1} > |D_0^\star|$, то $2^{n-1} > |D_1^\star|$, и поэтому можно снова воспользоваться леммой 1, применив ее к новому множеству D_1. Из этой леммы следует существование линейного $(n-2,n-1)$-оператора f_2 такого, что $f_2(\boldsymbol{x}) \neq f_2(\boldsymbol{y})$ для любых неравных наборов $\boldsymbol{x}$ и $\boldsymbol{y}$ из D_1. Положим $D_2 = f_1(D_1)$. Как и в предыдущем случае, легко видеть, что $|D_2^\star| \leqslant |D_1^\star|$. Заметим, что композиция $f_2 \circ f_1$ операторов f_2 и f_1 будет инъективным на D линейным $(n-2,n)$-оператором.

Предположим, что описанную процедуру выполнили в общей сложности $k-1$ раз и для каждого целого i от единицы до $k-1$ получили инъективный на области D_{i-1} линейный $(n-i+1,n-i)$-оператор f_i и лежащее в $\{0,1\}^{n-i}$ множество D_i такие, что $D_i = f_i(D_{i-1})$, $|D_i^\star| \leqslant |D_{i-1}^\star|$, а композиция $f_{k-1} \circ \cdots \circ f_1$ является инъективным на области D линейным $(n-k+1,n)$-оператором.

Если $2^{n-k+1} > |D_{k-1}^\star|+1$, то лемму 1 можно применить еще раз. Так как по предположению линейный $(n-k+1,n)$-оператор $f_{k-1} \circ \cdots \circ f_1$ отображает

разные наборы области D в разные наборы ее образа D_{k-1}, а линейный $(n-k, n-k+1)$-оператор f_k действует на D_{k-1} инъективно, то легко видеть, что композиция $f_k \circ (f_{k-1} \circ \cdots \circ f_1)$, полученных в результате применения леммы 1 операторов f_i, будет инъективным на области D линейным $(n-k, n)$-оператором.

Наконец заметим, что при $k \leqslant n-m$ из условий теоремы и сделанного предположения следуют неравенства

$$2^{n-k+1} \geqslant 2^{m+1} > |D_0^\star| \geqslant |D_{k-2}^\star| \geqslant |D_{k-1}^\star|.$$

Поэтому очевидно, что леммой 1 можно воспользоваться в общей сложности не менее $n-m$ раз, а получившийся в результате линейный (m, n)-оператор $f_{n-m} \circ \cdots \circ f_1$ будет инъективным на области D. Теорема 2 доказана.

Доказательство теоремы 1. Так как

$$2^{(\lfloor 2\log_2 |D| \rfloor - 1)+1} > \frac{|D|^2}{2} \geqslant \frac{|D|(|D|-1)}{2} + 1 \geqslant |D^*| + 1,$$

то при некотором $m \leqslant \lfloor 2\log_2 |D| \rfloor - 1$ в силу теоремы 2 найдется инъективный на D линейный (m, n)-оператор. Теорема 1 доказана.

3. Нижние оценки ранга

Теперь покажем, что для любого целого m, не превосходящего $n/2$, в n-мерном булевом кубе найдется область D_m, состоящая из $2^{m+1}-1$ наборов, и такая, что число компонент любого инъективного на этой области линейного оператора не меньше $2m$.

Пусть $m \leqslant n/2$ и $\mathbf{e}_1, \ldots, \mathbf{e}_{2m}$ — первые $2m$ базисных векторов стандартного базиса E_n. Положим

$$D_m = \langle \mathbf{e}_1, \ldots, \mathbf{e}_m \rangle \cup \langle \mathbf{e}_{m+1}, \ldots, \mathbf{e}_{2m} \rangle.$$

Легко видеть, что D_m состоит из $2^{m+1}-1$ различных наборов, а множество $D_m^\star$ попарных сумм наборов из D_m вместе с нулевым набором образуют подпространство размерности $2m$ в $\{0,1\}^n$. Поэтому очевидно, что размерность любого подпространства, не пересекающегося с $D_m^\star$, не меньше чем $n-2m$. Следовательно, ранг любого инъективного на области D_m линейного оператора не превосходит $2m$.

Так как $\lfloor 2\log_2 |D_m| \rfloor - 1 = 2m$, то теорема 1 гарантирует существование инъективного на D_m линейного $(2m, n)$-оператора. Таким образом в общем случае неравенство теоремы 1 является точным и усилить его нельзя.

Далее покажем, что неравенство теоремы 1 является асимптотически точным для почти всех областей из $D_{n,N}$ при условии, что $\frac{\log_2 N}{\log_2 n}$ неограниченно возрастает при $n \to \infty$. Сделаем это следующим образом. Сначала для произвольной области D из $\{0,1\}^n$ введем функцию μ, определив ее равенством

$$\mu(D) = \min \operatorname{rank} f,$$

в котором минимум берется по всем инъективным на области D линейным операторам. Затем величину $\mu(D)$ оценим снизу для почти всех областей из $D_{n,N}$.

Теорема 3. *Пусть $n \leqslant N \leqslant 2\sqrt{n^2 2^n}$. Тогда при $n \to \infty$ для почти всех $D \in D_{n,N}$*

$$\mu(D) \geqslant 2\log_2 N - 2\log_2 n - 2.$$

Доказательство. Пусть f — произвольный линейный (m,n)-оператор. Через $M(f,N)$ обозначим число областей из $D_{n,N}$, на которых оператор f является инъективным. Легко видеть, что для каждой такой области D никакие два набора из D не принадлежат одному и тому же смежному классу пространства $\{0,1\}^n$ по ядру оператора f. Поэтому для любого оператора ранга $k \leqslant m$

$$M(f,N) = \binom{2^k}{N} 2^{(n-k)N} \leqslant \binom{2^m}{N} 2^{(n-m)N}. \tag{25}$$

Теперь предположим, что при некоторой постоянной δ не менее чем для $\delta\binom{2^n}{N}$ областей из $D_{n,N}$ среди линейных (m,n)-операторов найдутся инъективные на этих областях операторы. Так как число различных линейных (m,n)-операторов равно 2^{mn}, то в среднем каждый линейный (m,n)-оператор является инъективным не менее чем для $\delta\binom{2^n}{d}2^{-mn}$ областей мощности N. Следовательно, найдется оператор, который будет инъективным по крайней мере для

$$P = \delta\binom{2^n}{N}2^{-mn}$$

различных областей. С другой стороны необходимо, чтобы величина P не превосходила $M(f,N)$. Поэтому из (25) и последнего неравенства

$$\delta\binom{2^n}{N}2^{-mn} \leqslant \binom{2^m}{N}2^{(n-m)N}.$$

Откуда после несложных преобразований получаем

$$\binom{2^n}{N} \Big/ \binom{2^m}{N} \leqslant \frac{1}{\delta} 2^{mn} 2^{(n-m)N}. \tag{26}$$

Легко видеть, что

$$\binom{2^n}{N}\Big/\binom{2^m}{N} = \frac{2^n(2^n-1)\dots(2^n-N+1)}{2^m(2^m-1)\dots(2^m-N+1)} \tag{27}$$

Оценим снизу натуральный логарифм правой части последнего равенства. Так как функция $\ln\frac{a-x}{b-x}$ выпукла вниз при $a > b > 0$ и $x \in [0,b)$, то

$$\ln\frac{2^n(2^n-1)\dots(2^n-N+1)}{2^m(2^m-1)\dots(2^m-N+1)} \geqslant N\ln\frac{2^n-(N-1)/2}{2^m-(N-1)/2} \geqslant$$
$$\geqslant N\ln 2^{n-m} + N\ln\frac{1-(N-1)/2\cdot 2^n}{1-(N-1)/2\cdot 2^m}. \tag{28}$$

Теперь оценим последнее слагаемое в правой части (28). Для этого используем справедливое при $1 > y \geqslant x \geqslant 0$ неравенство $\frac{1-x}{1-y} \geqslant 1-x+y$ и справедливое при $x \in [0,1)$ неравенство $\ln(1+x) \geqslant \frac{x}{2}$. Так как $m < n$ и $N < 2^m$, то

$$N\ln\frac{1-(N-1)/2\cdot 2^n}{1-(N-1)/2\cdot 2^m} \geqslant N\ln\Big(1+\frac{N-1}{2}\Big(\frac{1}{2^m}-\frac{1}{2^n}\Big)\Big) \geqslant \frac{N(N-1)}{4\cdot 2^m}. \tag{29}$$

Из (26)–(29) следует, что

$$\ln\Big(\frac{1}{\delta}2^{mn}2^{(n-m)N}\Big) \geqslant N\ln 2^{n-m} + \frac{N(N-1)}{4\cdot 2^m},$$

или, после очевидных преобразований,

$$2^m \geqslant \frac{N(N-1)}{4(nm\ln 2 - \ln\delta)}.$$

Логарифмируя последнее неравенство по основанию 2, видим, что при любой постоянной δ, начиная с некоторого n имеет место неравенство

$$m \geqslant 2\log_2 N - 2\log_2 n - 2.$$

Теорема доказана.

4. Сложность инъективных линейных операторов

Как обычно, сложность реализации булевой функции f схемами в базисе из всех двухместных функций будем обозначать через $L(f)$.

Теорема 4. *Для любой постоянной $\varepsilon > 0$ и любой области $D \subseteq \{0,1\}^n$, состоящей не более чем из $\sqrt{2^n}$ наборов, найдется инъективный на этой области линейный (m,n)-оператор f, для числа компонент которого справедливо неравенство*

$$m \leqslant (2+\varepsilon)\log_2 |D|,$$

и сложность которого есть $\mathcal{O}(n)$.

Утверждение теоремы является простым следствием доказываемых ниже лемм 5 и 6. При этом линейный оператор f будет композицией операторов из этих лемм.

Будем говорить, что i-я строка двоичной матрицы M покрывает ее j-й столбец, если в M на пересечении i-й строки и j-го столбца стоит единица.

Лемма 2. *Найдется такая постоянная $\delta > 0$, что для любого достаточно большого n существует двоичная матрица $M_{2n,n}$ из $2n$ строк и n столбцов, в каждой строке которой находится ровно 7 единиц и в которой ($\star$) при любом k, не превосходящем $2n\delta$, любые k строк покрывают более чем $4k$ столбцов.*

Доказательство. Пусть R — множество всех матриц, состоящих из $2n$ строк и n столбцов, во всех строках которых находится ровно 7 единиц. Оценим величину N, равную отношению числа тех матриц из R, которые не обладают свойством ($\star$), к числу всех матриц из R. Нетрудно видеть, что k строк, в которых единицы сосредоточены на пересечении не более чем с $4k$ столбцами, можно выбрать $\binom{2n}{k}$ способами, а соответствующие им столбцы — $\binom{n}{4k}$ способами, единицы в выбранных строках можно расставить не более чем $\binom{4k}{7}^k$ способами, в оставшихся строках это можно сделать $\binom{n}{7}^{2n-k}$ способами. Поэтому

$$\begin{aligned}
N &\leqslant \sum_{k=1}^{2n\delta} \binom{2n}{k}\binom{n}{4k}\binom{4k}{7}^k \binom{n}{7}^{2n-k}\binom{n}{7}^{-2n} = \\
&= \sum_{k=1}^{2n\delta} \binom{2n}{k}\binom{n}{4k}\binom{4k}{7}^k \binom{n}{7}^{-k} \leqslant \\
&\leqslant \sum_{k=1}^{2n\delta} \Big(\frac{3\cdot 2n}{k}\Big)^k \Big(\frac{3\cdot n}{4k}\Big)^{4k} \Big(\frac{4k(4k-1)\ldots(4k-6)}{n(n-1)\ldots(n-6)}\Big)^k \leqslant \\
&\leqslant \sum_{k=1}^{2n\delta} \Big(\frac{3\cdot 2n}{k}\Big)^k \Big(\frac{3n}{4k}\Big)^{4k} \Big(\frac{4k}{n}\Big)^{7k} = \sum_{k=1}^{2n\delta} 3^{5k}2^{7k}\Big(\frac{k}{n}\Big)^{2k} < \sum_{k=1}^{2n\delta} \left(3^5 2^7 \delta^2\right)^k.
\end{aligned}$$

Нетрудно видеть, что при при выполнении неравенства $3^5 2^7 \delta^2 \leqslant 2^{-1}$ (которое, очевидно, справедливо при $\delta < 2^{-8}$) отношение N будет меньше единицы, и, следовательно, найдется матрица, удовлетворяющая условиям леммы. Лемма доказана.

Лемма 3. *Найдется такая постоянная $\delta > 0$, что для любого достаточно большого n существует двоичная матрица $M_{2n,n}$ из $2n$ строк и n столбцов, в каждой строке которой находится ровно 7 единиц и в которой при любом k, не превосходящем $2n\delta$, в каждой подматрице, образованной k строками, найдется более k столбцов содержащих ровно один единичный элемент.*

Доказательство. Пусть матрица M удовлетворяет условию ($\star$) из леммы 2. В этой матрице произвольным образом выберем k строк и составим из них подматрицу M' матрицы M. Пусть R_1 обозначает число столбцов, покрываемых ровно одной из выбранных строк, а $R_{\geqslant 2}$ — число столбцов, покрываемых более чем одной такой строкой. Другими словами, R_1 равно числу столбцов подматрицы M' содержащих ровно по одному единичному элементу, а $R_{\geqslant 2}$ равно числу столбцов с более чем одним единичным элементом. В силу леммы 2 величины R_1 и $R_{\geqslant 2}$ удовлетворяют следующим неравенствам

$$R_1 + R_{\geqslant 2} > 4k,$$
$$R_1 + 2R_{\geqslant 2} \leqslant 7k.$$

Исключая из этих неравенств $R_{\geqslant 2}$, имеем

$$R_1 > 4k - R_{\geqslant 2} \geqslant 4k - \frac{1}{2}(7k - R_1) = \frac{1}{2}k + \frac{1}{2}R_1,$$

т. е. $R_1 > k$. Лемма доказана.

Лемма 4. *Пусть $M_{2n,n}$ — матрица из леммы 3. Тогда для любого двоичного набора v веса k, где $k \leqslant 2n\delta$, произведение $\boldsymbol{v} \cdot M_{2n,n}$ содержит более k единичных элементов.*

Доказательство. Пусть в наборе $\boldsymbol{v}$ компоненты $\boldsymbol{v}_{i_1}, \ldots, v_{i_k}$ ненулевые. В матрице $M_{2n,n}$ рассмотрим подматрицу M, образованную строками с номерами $i_1, \ldots, i_k$. В силу леммы 3 в этой подматрице найдутся столбцы с номерами $j_1, \ldots, j_s$ где $s > k$, каждый из которых содержит ровно один единичный элемент. Легко видеть, что для любого j_i из $\{j_1, \ldots, j_s\}$ скалярное

произведение $\boldsymbol{v}$ и j_i-го столба матрицы $M_{2n,n}$ равно единице. Следовательно, произведение $\boldsymbol{v} \cdot M_{2n,n}$ содержит $s > k$ единичных элементов. Лемма доказана.

Лемма 5. *Существует такая постоянная $0 < \gamma < 1$, что для любого достаточно большого n найдется такое линейное отображение $\mathcal{G}_{n,4n}$ из $\{0,1\}^n$ в $\{0,1\}^{4n}$, что $\|\mathcal{G}_{n,4n}(\boldsymbol{v})\| \geqslant \gamma n$ для любого ненулевого вектора $\boldsymbol{v}$ и $L(\mathcal{G}_{n,4n}) = \mathcal{O}(n)$.*

Доказательство. Лемму докажем индукцией по двоичному логарифму n. В основание индукции положим отображение $\mathcal{G}_{m,4m}(\boldsymbol{v}) = (\boldsymbol{v}, \boldsymbol{v}, \boldsymbol{v}, \boldsymbol{v})$, где m — минимально возможное, при котором существует матрица $M_{4m,2m}$. Очевидно, что в этом случае $\gamma = 4/m$ и $L(\mathcal{G}_{m,4m}) = 0$.

Теперь допустим, что удовлетворяющее условиям леммы линейное отображение $\mathcal{G}_{n,4n}$ с матрицей $G_{n,4n}$ существует при некотором $n \geqslant m$. Используя это отображение построим отображение $\mathcal{G}_{2n,8n}$. Пусть $\boldsymbol{v}$ — вектор длины $2n$, $\boldsymbol{v}' = \boldsymbol{v} \cdot M_{2n,n}$ — вектор длины n, $\boldsymbol{w} = \boldsymbol{v}' \cdot G_{n,4n}$ — вектор длины $4n$, $\boldsymbol{u} = \boldsymbol{w} \cdot M_{4n,2n}$ — вектор длины $2n$. Тогда $\mathcal{G}_{2n,8n}(\boldsymbol{v}) = (\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{u})$. Покажем, что неравенство

$$\|\mathcal{G}_{2n,8n}(\boldsymbol{v})\| \geqslant 2n\gamma \tag{30}$$

справедливо для любого ненулевого вектора $\boldsymbol{v}$ длины $2n$ и $\gamma = \min(4/m, \delta)$, где δ — постоянная из леммы 4.

Если $\|\boldsymbol{v}\| \geqslant 2n\gamma$, то, очевидно, имеет место и неравенство (30). Если вес ненулевого вектора $\boldsymbol{v}$ меньше чем $2n\gamma$, то в силу леммы 4 вес вектора $\boldsymbol{v}'$ больше нуля, и в силу предположения индукции $\|\boldsymbol{w}\| \geqslant n\delta$. Если при этом справедливо более сильное неравенство $\|\boldsymbol{w}\| \geqslant 2n\gamma$, то (30) также справедливо. Если же $n\gamma < \|\boldsymbol{w}\| < 2n\gamma$, то в силу леммы 4 вес вектора $\boldsymbol{u}$ больше веса вектора $\boldsymbol{w}$ и поэтому $\|\mathcal{G}_{2n,8n}(\boldsymbol{v})\| > \|\boldsymbol{w}\| + \|\boldsymbol{u}\| > 2n\gamma$.

Покажем, что $L(\mathcal{G}_{n,4n}) = \mathcal{O}(n)$. Допустим, что $L(\mathcal{G}_{n,4n}) \leqslant 42n$ при $n > m$. Так как сложность умножения матрицы на вектор не превосходит числа единичных элементов матрицы, то

$$\begin{aligned} L(\mathcal{G}_{2n,8n}) &\leqslant L(M_{2n,n}) + L(\mathcal{G}_{n,4n}) + L(M_{4n,2n}) \leqslant \\ &\leqslant 14n + 42n + 28n = 42 \cdot 2n. \end{aligned}$$

Лемма доказана.

Лемма 6. *Пусть целые t и R удовлетворяют неравенствам $R \geqslant 256$ и $2 \leqslant t \leqslant \log_2(\frac{1}{2}\log_2 R)$. Положим $m = \lceil \log_2 R \cdot (1 + 4/2^t) \rceil$. Тогда для любых наборов $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_R$ из $\{0,1\}^n$, вес каждого из которых не меньше*

$d = \delta n$, где δ — константа, найдется такой линейный оператор $\mathcal{L}$ из $\{0,1\}^n$ в $\{0,1\}^m$, что $L(\mathcal{L}) = \mathcal{O}(n)$ и ни один из наборов $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_R$ не отображается этим оператором в нулевой набор.

Доказательство. Пусть $M(n,m)$ — множество булевых матриц с m строками и n столбцами, p — постоянная из $(0, \frac{1}{2})$. В $M(n,m)$ случайным образом выберем матрицу M, полагая, что элементы M_{ij} этой матрицы выбираются независимо с вероятностями $\mathbf{Pr}(M_{ij} = 1) = p$ и $\mathbf{Pr}(M_{ij} = 0) = 1-p$. Пусть $\boldsymbol{a}$ — двоичный набор длины n и веса d. Найдем вероятность того, что линейный оператор $\mathcal{M}$ с выбранной матрицей M отображает набор $\boldsymbol{a}$ в нулевой набор. Нетрудно видеть, что для i-й компоненты $\mathcal{M}_i$ оператора $\mathcal{M}$ справедливы равенства

$$\mathbf{Pr}(\mathcal{M}_i(\boldsymbol{a}) = 0) = \sum_{k=0}^{\lfloor d/2 \rfloor} \binom{d}{2k} p^{2k} (1-p)^{d-2k},$$

$$\mathbf{Pr}(\mathcal{M}_i(\boldsymbol{a}) = 1) = \sum_{k=0}^{\lfloor d/2 \rfloor} \binom{d}{2k} p^{2k+1} (1-p)^{d-2k-1}.$$

Поэтому, учитывая равенство

$$\sum_{k=0}^{d} \binom{d}{k} (-p)^k (1-p)^{d-k} = ((1-p)-p)^d,$$

находим, что

$$\mathbf{Pr}(\mathcal{M}_i(\boldsymbol{a}) = 0) + \mathbf{Pr}(\mathcal{M}_i(\boldsymbol{a}) = 1) = 1,$$
$$\mathbf{Pr}(\mathcal{M}_i(\boldsymbol{a}) = 0) - \mathbf{Pr}(\mathcal{M}_i(\boldsymbol{a}) = 1) = (1-2p)^d.$$

Таким образом,

$$\mathbf{Pr}(\mathcal{M}_i(\boldsymbol{a}) = 0) = \frac{1}{2}\big(1 + (1-2p)^d\big),$$

и, следовательно, для искомой вероятности справедливо равенство

$$\mathbf{Pr}(\mathcal{M}(\boldsymbol{a}) = 0) = \Big(\frac{1}{2}\Big(1 + (1-2p)^d\Big)\Big)^m.$$

Тогда,

$$\mathbf{Pr}(\mathcal{M}(\boldsymbol{a}) = 0) = \frac{1}{2^m}\Big(1 + (1-2p)^{\frac{1}{2p}\cdot 2pd}\Big)^m \leqslant \frac{1}{2^m}\Big(1 + \frac{1}{2^{2pd}}\Big)^m \leqslant$$

$$\leqslant \frac{1}{2^m}\Bigl(1+\frac{1}{2^{2pd}}\Bigr)^{2^{2pd}\cdot m/2^{2pd}} \leqslant \frac{1}{2^m}\cdot 4^{m/2^{2pd}} = 2^{-m(1-2/2^{2pd})}.$$

Нетрудно видеть, что полученная оценка вероятности $\mathbf{Pr}(\mathcal{M}(\boldsymbol{a})=0)$ убывает с ростом d.

Положим $p = t/2d$. В этом случае

$$\mathbf{Pr}(\mathcal{M}(\boldsymbol{a})=0) \leqslant \frac{1}{2^m}\cdot 4^{m/2^{2pd}} = 2^{-m(1-2/2^t)}.$$

Поэтому для любых $\boldsymbol{a}_1,\ldots,\boldsymbol{a}_R$ из A справедливо неравенство

$$\mathbf{Pr}(\mathcal{M}(\boldsymbol{a}_1)=0\vee\cdots\vee\mathcal{M}(\boldsymbol{a}_R)=0) \leqslant R\cdot 2^{-m(1-2/2^t)}.$$

Положим $m = \lceil \log_2 R\cdot(1+4/2^t)\rceil$. Тогда при $t\geqslant 3$

$$m(1-2/2^t) \geqslant \log_2 R\cdot(1+4/2^t)(1-2/2^t) \geqslant \log_2 R\cdot(1+1/2^t).$$

Подставляя полученную оценку в предыдущее неравенство, имеем

$$\mathbf{Pr}(\mathcal{M}(\boldsymbol{a}_1)=0\vee\cdots\vee\mathcal{M}(\boldsymbol{a}_R)=0) \leqslant R\cdot R^{-1-2^{-t}} = R^{-2^{-t}},$$

откуда после несложных преобразований находим, что

$$\mathbf{Pr}(\mathcal{M}(\boldsymbol{a}_1)=0\vee\cdots\vee\mathcal{M}(\boldsymbol{a}_R)=0) \leqslant \frac{1}{4} \tag{31}$$

при $3\leqslant t\leqslant \log_2(\frac{1}{2}\log_2 R)$.

Теперь оценим $||M||$ — число единичных элементов в матрице M. Это число является случайной величиной ξ, математическое ожидание $\mathbf{M}\xi$ и дисперсия $\mathbf{D}\xi$ которой равны, соответственно, $mnp = tmn/2d$ и $nmp(1-p)\leqslant mnt/2d$. Положим $s = 2\sqrt{\mathbf{D}\xi}$. Тогда из неравенства Чебышева следует, что

$$\mathbf{Pr}(|\xi-\mathbf{M}\xi|\geqslant s) \leqslant \frac{\mathbf{D}\xi}{s^2} = \frac{1}{4}. \tag{32}$$

Таким образом, вероятность того, что число ненулевых элементов матрицы M больше $tmn/2d+\sqrt{2tmn/d} < 2tmn/d$ не превосходит $1/4$.

Объединяя неравенства (31) и (32), видим, что

$$\begin{aligned}\mathbf{Pr}(\mathcal{M}(\boldsymbol{a}_1)\neq 0\,\&\,\ldots\,\&\mathcal{M}(\boldsymbol{a}_R)\neq 0\,\&\,||M||<tm/\delta) &= \\ = 1-\mathbf{Pr}(\mathcal{M}(\boldsymbol{a}_1)=0\vee\cdots\vee\mathcal{M}(\boldsymbol{a}_R)=0\vee||M||\geqslant tm/\delta) &\geqslant\end{aligned}$$

$$\geqslant 1 - \mathbf{Pr}(\mathcal{M}(\boldsymbol{a}_1) = 0 \vee \cdots \vee \mathcal{M}(\boldsymbol{a}_R) = 0) - \mathbf{Pr}(||M|| \geqslant tm/\delta) \geqslant \frac{1}{2},$$

где $m = \lceil \log_2 R \cdot (1 + 4/2^t) \rceil$ и $3 \leqslant t \leqslant \log_2(\frac{1}{2}\log_2 R)$.

Следовательно, для любых $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_R$ найдется линейный оператор из $\{0,1\}^n$ в $\to \{0,1\}^m$, который не отображает ни один из этих наборов в нулевой набор и в матрице которого находится не более tm/δ единиц. Так как $m \leqslant n$, а δ и t — константы, то очевидно, что сложность оператора $\mathcal{M}$ есть $\mathcal{O}(n)$. Лемма доказана.

Список литературы

1. Кричевский Р. Е. Сжатие и поиск информации. — М.: Радио и связь, 1989.
2. S.I. Gelfand, R.L. Dobrushin, and M.S. Pinsker. On the complexity of coding. In Second International Symposium on Information Theory, pages 177–184, Akademiai Kiado, Budapest, 1973.
3. Peter Bro Miltersen. Error Correcting Codes, Perfect Hashing Circuits, and Deterministic Dynamic Dictionaries. June 1997. http://www.brics.dk/RS/97/17/BRICS-RS-97-17.pdf
4. Madhu Sudan. Essential Coding Theory. http://theory.lcs.mit.edu/~madhu/FT02/

СОДЕРЖАНИЕ