Федеральное государственное бюджетное учреждение науки
Санкт-Петербургское отделение
Математического института им. В. А. Стеклова РАН

На правах рукописи

КУЛИКОВ Александр Сергеевич

СХЕМНАЯ СЛОЖНОСТЬ ЯВНО ЗАДАННЫХ БУЛЕВЫХ ФУНКЦИЙ

01.01.06 — математическая логика, алгебра и теория чисел

Диссертация

на соискание учёной степени
доктора физико-математических наук

Санкт-Петербург
2016

Оглавление

Введение

Основные определения и обозначения

Определение 1 (булева функция)**.** *Обозначим через $B_{n,m}$ множество всех* булевых функций $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ *с n входами и m выходами, где $\mathbb{F}_2 = \{0,1\}$ — поле из двух элементов. Через B_n будем обозначать $B_{n,1}$. Через n мы всегда будем обозначать число входных битов рассматриваемой функции. Функция называется* симметрической*, если её значение зависит только от суммы входных битов.*

Замечание 2. *В доказательствах нижних оценок, как правило, под булевой функцией мы будем понимать бесконечную последовательность функций $f_1, f_2, \ldots$, где $f_i \in B_i$.*

Определение 3 (булева схема)**.** Булевой схемой *над базисом $\Omega \subseteq B_2$ называется ациклический ориентированный граф, в котором каждая внутренняя вершина помечена бинарной булевой операцией из множества Ω, входные биты (называемые также переменными) подаются в вершины входящей степени ноль. Вершины также называются* функциональными элементами *или просто* элементами*, а рёбра —* проводами*. Некоторые m вершин также помечены как выходные. Схема естественным образом вычисляет булеву функцию из n в m переменных, где n — это число входных вершин, а m — число выходных.* Размером *схемы $\mathcal{C}$ будем называть количество внутренних вершин в $\mathcal{C}$ (то есть вершин, входящая степень которых больше нуля).*

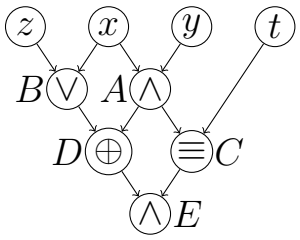
Обозначение: $\texttt{gates}(\mathcal{C})$.

В данной работе мы будем рассматривать два основных базиса:

- полный бинарный базис B_2;
- базис $U_2 = B_2 \setminus \{\oplus, \equiv\}$, состоящий из всех бинарных функций, кроме функции чётности ($\oplus$) и её дополнения ($\equiv$).

По умолчанию под схемой мы будем понимать схему над полным бинарным базисом.

Нетрудно видеть, что схема соответствует естественной и очень просто устроенной программе для вычисления булевой функции из $B_{n,m}$: каждая инструкция в такой программе вычисляет и сохраняет в новой переменной результат бинарной булевой операции, применённой к двум переменным, каждая из которых является либо входным битом, либо результатом одной из предыдущих операций. Размер схемы равен числу инструкций в программе. Ниже показана схема размера пять, вычисляющая булеву функцию от четырёх переменных, и соответствующая ей программа.



$$B = (z \vee x)$$
$$A = (x \wedge y)$$
$$D = (B \oplus A)$$
$$C = (A \equiv t)$$
$$E = (D \wedge C)$$

Определение 4 (схемная сложность булевой функции)**.** Схемной сложностью $\texttt{gates}(f)$ *булевой функции* f *называется размер минимальной схемы, вычисляющей данную функцию.*

Актуальность избранной темы и степень её разработанности

Изучение схемной сложности булевых функций — центральный вопрос современной теоретической информатики. Нетрудно видеть, что любую функцию $f \in B_n$ можно вычислить схемой размера $O(n2^n)$, представив функцию в виде дизъюнктивной нормальной формы. В 1956 г. Д. Мюллером [39] было показано, что $\texttt{gates}(f) = O(2^n/n)$, а в 1958 г. О. Б. Лупанов [65] установил, что

$$\texttt{gates}(f) \leq \left(1 + O\left(\frac{\log n}{n}\right)\right)\frac{2^n}{n}.$$

Симметрические функции могут быть посчитаны схемами гораздо меньшего размера: Е. А. Деменковым и др. в 2010 г. [12] было показано, что схемная сложность любой симметрической функции не превосходит $4.5n + o(n)$.

Уже в 1949 г. К. Шэннон [52] показал, что почти все булевы функции от n переменных требуют схем размера $\Omega(2^n/n)$. Это следует из простых мощностных соображений: число 2^{2^n} различных функций от n переменных растёт быстрее, чем число схем малого размера. Данное доказательство, однако, неконструктивно — оно не даёт примера явно заданной булевой функции высокой схемной сложности. Под явно заданной, как правило, понимается функция $\{f_1, f_2, \dots\}$, для которой $\cup_{i=1}^{\infty} f_i^{-1}(1) \in \mathrm{NP}$.

Доказательство суперполиномиальных нижних оценок на схемную сложность явно заданных булевых функций оказалось очень трудной задачей (отметим, что такая оценка повлекла бы за собой неравенство классов P и NP). К настоящему моменту удалось доказать лишь небольшие линейные нижние оценки. В 1965 г. Б. М. Клосс и В. А. Малышев [64] доказали нижнюю оценку $2n - O(1)$ для функции $\oplus_{1 \leq i < j \leq n} x_i x_j$. В 1974 г. К. Шнорр [48] доказал нижнюю оценку $2n - O(1)$ для широкого класса функций со следующим естественным свойством: для любых двух входных переменных среди четырёх

подфункций, получаемых подстановкой констант данным двум переменным, будет хотя бы три разные. В 1977 г. Л. Стокмайер [53] опубликовал доказательство нижней оценки $2.5n - O(1)$ для многих симметрических функций (в частности, для функции $\mathrm{MOD}^n_{m,r}$, выдающей 1 тогда и только тогда, когда сумма n входных битов сравнима с r по модулю $m \geq 3$). В том же году В. Пол [44] доказал нижнюю оценку $2n - o(n)$ для функции индексации, а также нижнюю оценку $2.5n - o(n)$ для специально построенной функции, комбинирующей несколько функций индексации. Наконец, в 1984 г. Н. Блюм [7] расширил идеи В. Пола и получил доказательство нижней оценки $3n - o(n)$.

Другие модели

Сложность вычислительной задачи зависит от рассматриваемой модели вычислений. К распространённым моделям относятся машины Тьюринга, равнодоступные адресные машины (PAM-машины), булевы схемы. Схемы в полном бинарном базисе — гибкая модель вычислений. Использование k-арного базиса вместо бинарного изменяет сложность лишь в константу раз. Использование фиксированного множества элементов неограниченной арности (например, конъюнкций, дизъюнкций и отрицаний) сохраняет сложность, измеряемую как число проводов в схеме. Нахождение функции, которую трудно вычислить схемами, может рассматриваться как комбинаторная задача (в отличие от нижних оценок для равномерных моделей). Поэтому доказательство суперлинейной нижней оценки на схемную сложность — важный рубеж на пути получения сильных нижних оценок.

Более сильные, чем $3n$, нижние оценки известны для различных ограниченных базисов. Один из наиболее популярных таких базисов U_2 состоит из всех бинарных функций, кроме функции чётности и её дополнения. В 1976 г. П. Шнорр [48] доказал, что сложность функции чётности в таком базисе равна $3n - 3$. У. Цвик в 1991 г. [62] привёл доказательство нижней оцен-

ки $4n - \Theta(1)$ для многих симметрических функций. В 2001 г. О. Лахич и Р. Раз [35] доказали нижнюю оценку $4.5n - o(n)$ на сложность $(n - o(n))$-смешанной функции (функции называется k-смешанной, если для любых её k переменных все 2^k способов подставить им значения дают попарно различные подфункции). К. Ивама и Х. Морицуми в 2002 г. [27] улучшили оценку до $5n - o(n)$. Интересно отметить, что для улучшения нижней оценки $5n - o(n)$ в базисе U_2 нужны новые идеи: в 2011 г. К. Амано и Й. Таруй [2] привели пример $(n - o(n))$-смешанной функции, схемная сложность которой над U_2 не превосходит $5n + o(n)$.

Несмотря на то, что неизвестно нелинейных нижних оценок для булевых схем в базисе константной арности, более сильные оценки известны для ограниченных классов схем:

- для монотонных схем (А. А. Разборов, 1985 г. [69]),
- для схем константной глубины (Э. Яо, 1985 г.; Й. Хостад, 1986 г. [60, 23]),
- для формул (Б. А. Субботовская, 1961 г. [71]; Э. И. Нечипорук, 1961 г. [66]; В. М. Храпченко, 1971 г. [72]; А. А. Андреев, 1985 г. [63]; Р. Импальяццо и Н. Ниссан, 1993 г. [26], М. Патерсон и У. Цвик, 1993 г. [42], Й. Хостад, 1998 г. [24] А. Таль, 2014 г. [54]).

Данные нижние оценки, однако, не транслируются в нелинейные нижние оценки для неограниченных моделей схем в базисе константной арности. Подробные обзоры результатов для различных моделей можно найти в книгах И. Вегенера [55], Р. Г. Нигматуллина [67], С. Юкны [28].

Связь с алгоритмами для задачи выполнимости схемы

Недавние результаты Р. Вильямса [57] устанавливают интересную связь между доказательством нижних оценок на сложность схем из некоторого

класса и доказательством верхних оценок на время работы алгоритмов, проверяющих выполнимость схем из этого класса. А именно, существование алгоритма, решающего задачу выполнимости существенно быстрее, чем за время 2^n, влечёт за собой экспоненциальную нижнюю оценку на схемную сложность функций из большого сложностного класса (такого как NEXP). С использованием данной связи Вильямсом в 2014 г. [58] были доказаны безусловные нижние оценки для ACC_0 схем (схем константной глубины с элементами неограниченной арности, вычисляющими конъюнкцию, дизъюнкцию, отрицание и произвольные MOD-функции) Э. Бен-Сассон и Э. Виола [6] в 2014 г. показали, что для доказательства конкретной линейной нижней оценки на схемную сложность функции E^{NP} достаточно уменьшить константу в основании экспоненты времени работы алгоритма для задачи 3-выполнимости до подходящего значения (стоит, однако, отметить, что известные на данный момент константы не дают новых нижних оценок).

Техника, использующаяся в доказательстве нижних оценок на схемную сложность, применяется также при разработке эффективных алгоритмов для задачи выполнимости схем и формул (см., например, [68, 70, 46, 50, 32, 9, 8]).

Цели и задачи работы

Основной целью данной работы является как усиление известных нижних и верхних оценок на схемную сложность явно заданных булевых функций, так и развитие методов получения таких оценок.

Структура и результаты настоящей диссертации, используемые методы

Работа поделена на две основные главы: в первой главе доказываются нижние оценки на схемную сложность, во второй — верхние. **Глава 1** начинается с описания метода элиминации элементов. В качестве примера используется доказательство К. Шнорра [47] нижней оценки $2n - \Theta(1)$ на схемную сложность широкого класса функций, удовлетворяющих следующему свойству: для любых двух переменных среди четырёх подфункций, получающихся подстановкой констант этим двум переменным, есть хотя бы три различные. В доказательстве показывается, что в любой схеме, вычисляющей такую функцию, найдётся переменная, из которой выходят хотя бы два провода. При подстановке константы в такую переменную удаляются хотя бы два элемента, после чего требуемая оценка получается по индукции.

В **разделе 1.1** нижняя оценка К. Шнорра усиливается до $\frac{7n}{3} - \Theta(1)$ наложением дополнительного ограничения на степень функции. Само доказательство при этом остаётся почти таким же простым и при этом по-прежнему работает для очень широкого класса функций. Основной идеей доказательства является использование нестандартной меры сложности схем, присваивающей разные веса элементам разного типа. Такие меры использованы впервые.

В **разделе 1.2** приводится очень простое (содержащее всего два случая) доказательство нижней оценки $3n - o(n)$ на схемную сложность аффинных дисперсеров сублинейной размерности. Аффинным дисперсером размерности d называется функция $f \in B_n$, не обращающаяся в константу ни на каком аффинном подпространстве $\mathbb{F}_2^n$ размерности хотя бы d. Такие объекты активно изучаются в последнее время в области извлечения случайности. В частности, сравнительно недавно Э. Бен-Сассоном и С. Коппарти [5] была

приведена явная конструкция аффинного дисперсера сублинейной размерности (то есть $d = o(n)$). Для получения нижних оценок на схемную сложность таких функций важно следующее из свойство: они не обращаются в константу после любых $n - d$ ограничений типа $p(x) = 0$, где p — линейный многочлен (такие ограничения как раз и задают аффинное подпространство). Используя данную конструкцию как чёрный ящик, удаётся очень просто доказать нижнюю оценку $3n - o(n)$. Для этого показывается, что для любой схемы найдётся линейная подстановка, удаляющая из схемы хотя бы три элемента. Таким образом, доказательство в некотором смысле аналогично доказательству К. Шнорра, но вместо константных подстановок используются линейные. При этом гораздо более сложным становится вопрос построения функции, устойчивой относительно таких подстановок. Стоит отметить, что полученная нижняя оценка $3n-o(n)$ с точностью до членов младшего порядка совпадает с нижней оценкой $3n - o(n)$ Н. Блюма [7], представленной им в 1984 г. и являющейся рекордной на протяжении последующих тридцати лет. Представленное в работе доказательство значительно проще (содержит всего два случая вместо нескольких десятков), но проводится для более сложной функции.

В **разделе 1.3** нижняя оценка для аффинных дисперсеров сублинейной размерности усиливается до $\left(3 + \frac{1}{86}\right) n - o(n)$. Основными идеями, позволившими достичь данного улучшения, являются следующие три. Во-первых, вместо схем рассматривается более общая модель — схемы с циклами в линейной части. Это позволяет производить линейные подстановки, оставаясь в необходимом классе схем, и пользоваться более сильным предположением индукции. Во-вторых, используются квадратичные подстановки, которые могут рассматриваться как отложенные линейные подстановки: в некоторых проблемных ситуациях производится подстановка $x_i \leftarrow x_j x_k$; впоследствии обязательно также производится подстановка константы вместо x_j или x_k, что

делает исходную подстановку линейной; в-третьих, используется аккуратно подобранная мера сложности схем, которая зависит от многих параметров схемы (и даже самого процесса элиминации элементов). Полученная оценка улучшает оценку Н. Блюма 1984 г. и является самой сильной из известных на сегодняшний день (для явно заданных булевых функций).

В **разделе 1.4** приводится гораздо более короткое и технически простое доказательство нижней оценки $3.11n$ для так называемых квадратичных дисперсеров. Грубо говоря (все формальные определения приведены в соответствующем разделе), такие функции устойчивы относительно достаточно большого количества подстановок типа $x \leftarrow p$, где p — многочлен степени не более двух. В настоящий момент явных конструкций (то есть конструкций из класса NP) таких функций неизвестно, хотя случайно выбранная функция является квадратичным дисперсером с вероятностью $1 - o(1)$ и известны конструкции с более слабыми параметрами и конструкции для полей большего размера. Основным ингредиентом доказательства является индукция не по числу переменных, как во всех известных доказательствах, а по размеру соответствующего квадратичного многообразия.

В **разделе 1.5** рассматриваются схемы над ограниченным базисом U_2, содержащим все бинарные функции, кроме функции чётности ($\oplus$) и её дополнения ($\equiv$). Приводится доказательство нижней оценки $5n - o(n)$ для линейной функции с $o(n)$ выходами, матрицей которой является проверочная матрица кода Хэмминга. Такая же оценка, но для функции с одним выходом, получена в 2002 г. К. Ивамой и Х. Морицуми [27] и является рекордной известной на сегодняшний день для данного базиса. Приводящееся в диссертации доказательство значительно короче и проще.

В **разделе 1.6** показывается, что методы, использующиеся при доказательстве нижних оценок на размер схем, могут быть также использованы для построения эффективных алгоритмов вычисления числа выполняющих на-

боров схемы, а также для получения нижних оценок на схемную сложность в среднем. Впервые такая связь была явно продемонстрирована Р. Ченем и В. Кабанцом [8] в 2015 г. В диссертации их идеи развиваются и обобщаются: доказывается общая теорема, которая позволяет по разбору случаев сразу сказать, какой из него получается алгоритм для задачи выполнимости схем и какие получаются нижние оценки на схемную сложность в среднем и наихудшем случаях. Далее с помощью данного метода доказывается нижняя оценка $3.24n$ на схемную сложность в среднем случае над базисом U_2 для аффинных дисперсеров сублинейной размерности (что означает, что схемы размера $3.24n$ не могут даже приближённо считать такие функции), а также верхняя оценка вида $(2-\varepsilon)^n$ (где ε — положительная константа) для задачи выполнимости схем размера не более $3.24n$. Обе полученные оценки являются самыми сильными из известных на данный момент.

В перечисленных выше разделах показывается, что метод элиминации элементов может быть использован для доказательства более сильных оценок, если у нас в распоряжении имеется функция, устойчивая относительно достаточно сильных подстановок. Например, аффинные дисперсеры позволяют доказать нижнюю оценку $\left(3+\frac{1}{86}\right)n - o(n)$, квадратичные — оценку $3.11n$. Естественно задаться вопросом: можно ли доказать нелинейные нижние оценки для функций, устойчивых относительно подстановок типа $p = 0$, где p — произвольный многочлен степени, скажем, 10 или даже $\log n$? (Отметим в скобках, что на настоящий момент у нас нет явных конструкций даже для функций, где многочлену p разрешается иметь степень всего лишь два.) **Раздел 1.7** посвящён отрицательному ответу на данный вопрос. Показывается, что относительно любых, сколь угодно сильных подстановок, есть схемы, из которых удаляется только константное число элементов. Таким образом, для доказательства нелинейных нижних оценок на схемную сложность будет недостаточно просто построить более сильные дисперсеры.

В **главе 2** доказываются новые верхние оценки для симметрических булевых функций. В **разделе 2.1** описывается оценки, доказанные при помощи компьютерной программы поиска оптимальных схем для функций от малого числа переменных. Для поиска таких схем используются программы, эффективно решающие задачу выполнимости формул в конъюнктивной нормальной форме (так называемые SAT-солверы). А именно, по данной таблице истинности функции $f \in B_{n,m}$ и числу r строится формула в КНФ, которая выполнима тогда и только тогда, когда для f существует схема из r элементов. Помимо сведения приводятся различные эвристики, которые на практике позволяют сократить или ускорить перебор.

Есть несколько причин интересоваться точной схемной сложностью функций от малого числа переменных. Во-первых, для некоторых функций эффективные схемы строятся из блоков константного размера. Уменьшение размера такого блока автоматически даёт более сильную оценку для такой функции в общем случае. Во-вторых, было бы очень полезно иметь энциклопедию оптимальных схем для функций от малого количества переменных. Скажем, знание оптимальных схем для задачи умножения булевых матриц размера $n \times n$ для, например, $n = 2, 3, \ldots, 10$ потенциально могло бы помочь нам понять, как устроен эффективный алгоритм для этой задачи. К сожалению, современные компьютеры и программы не позволяют найти оптимальный размер схем для этой задачи даже при $n = 3$. Д. Кнут [29] недавно реализовал свой вариант сведения и нашёл точную схемную сложность всех функций от четырёх переменных, а также некоторых функций от пяти переменных.

Таким образом автоматически были построены оптимальные схемы для некоторых функций от не более чем пяти переменных. Также был найден блок, с помощью которого функцию MOD_n^3 можно посчитать схемой размера $3n$. Это новая оценка, которая является лучшей из известных. Такие же эксперименты недавно были проведены Д. Кнутом [29], который, в частно-

сти, выдвинул гипотезу, что сложность функции $\mathrm{MOD}_n^{3,r}$ в точности равна $3n - 5 - [(n + r) \equiv 0 \pmod 3]$. Данная гипотеза верна при малых значениях n.

В **разделе 2.2** изучается вопрос одновременного вычисления нескольких симметрических функций. Известно, что любую симметрическую функцию $f \in B_n$ можно посчитать схемой размера $5n + o(n)$ (это следует из того, что сумму трёх битов можно посчитать схемой размера 5, известной как Full Adder) и даже схемой размера $4.5n + o(n)$, как показано Е. Деменковым и др. [14]. В то же время из мощностных соображений получается, что для почти всех наборов из n симметрических функций из B_n требуются схемы размера $\Omega(n^{2-o(1)})$. В настоящее время мы не знаем ни одного такого набора, требующего даже схем суперлинейного размера. Есть три естественных подкласса симметрических функций:

- $\mathrm{EX}_n^k \in B_n$ равна 1 тогда и только тогда, когда сумма входных n битов равна k;
- $\mathrm{THR}_n^k \in B_n$ равна 1 тогда и только тогда, когда сумма входных n битов хотя бы k;
- $\mathrm{MOD}_n^{m,r} \in B_n$ равна 1 тогда и только тогда, когда сумма входных n битов сравнима с r по модулю m.

Известно, что все функции из первого и второго класса (то есть для всех $k = 1, 2, \ldots, n$) можно вычислить схемой размера $O(n)$. Естественно задаться вопросом, верно ли это и для третьего класса. В **разделе 2.2** строятся схемы размера $O(n \log n)$ для вычисления всех MOD-функций одновременно. Таким образом, для нахождения наборов из n симметрических функций схемной сложности $\Omega(n^{1+\varepsilon})$ нужно брать более сложно устроенные симметрические функции.

Положения, выносимые на защиту

1. Доказательство нижней оценки $7n/3 - O(1)$ на схемную сложность широкого класса функций, представляемых многочленами степени n.

2. Доказательство нижней оценки $3n - o(n)$ на схемную сложность аффинных дисперсеров сублинейной размерности.

3. Доказательство нижней оценки $(3 + \frac{1}{86})n - o(n)$ на схемную сложность аффинных дисперсеров сублинейной размерности.

4. Доказательство нижней оценки $3.11n$ на схемную сложность при условии существования явно заданных квадратичных дисперсеров.

5. Доказательство нижней оценки $5n - o(n)$ в базисе U_2 на схемную сложность линейной функции с $o(n)$ выходами.

6. Доказательство нижней оценки $3.24n$ на схемную сложность в среднем случае над базисом U_2 для дисперсера сублинейной размерности относительно проекций.

7. Алгоритм, решающий задачу выполнимости схем в базисе U_2 за время $(2 - \varepsilon)^n$ для схем размера не более $3.24n$ (где $\varepsilon > 0$ — константа).

8. Показано, что для получения нелинейных нижних оценок на схемную сложность будет недостаточно просто привести явные конструкции дисперсеров относительно более сильных подстановок.

9. Доказательство верхней оценки $3n$ на схемную сложность функции MOD_n^3.

10. Доказательство верхней оценки $O(n \log\log n)$ на схемную сложность одновременного вычисления функций MOD_n^m для всех $m = 1, \ldots, n$.

Научная новизна

Все полученные оценки на размер схем являются новыми и ранее не известными. Нижняя оценка $3n - o(n)$ на схемную сложность аффинных дисперсеров сублинейной размерности совпадает (с точностью до членов младшего порядка) с оценкой, доказанной Н. Блюмом в 1984 г. (для другой функции), но доказывается существенно проще. Нижняя оценка $(3 + \frac{1}{86})n - o(n)$ является самой сильной из известных для схем над полным бинарным базисом. Нижняя оценка $5n - o(n)$ является рекордной для схем над базисом U_2 для функций с $o(n)$ выходами. Нижняя оценка $3.24n$ на схемную сложность в среднем является самой сильной из известных. Верхняя оценка $3n$ является самой сильной известной оценкой на схемную сложность функции MOD_n^3. Построенный алгоритм для задачи выполнимости булевых схем является самым быстрым из известных.

Теоретическая и практическая значимость работы

Диссертация имеет теоретический характер. Полученные новые нижние оценки на размер схем могут быть использованы для дальнейшего изучения сложности булевых функций. В то же время полученные верхние оценки могут быть применены при практической реализации микросхем. Некоторые из полученных в диссертации результатов уже включены в содержание специальных курсов и учебников по теоретической информатике [28, 59].

Апробация работы

Основные результаты обсуждались на следующих конференциях и семинарах:

1. Международная студенческая школа Fall School of Logic and Complexity in Prague (Чехия, 2009).

2. Международный семинар Estonian Theory Days (Эстония, 2009).

3. Российский семинар “Логика и теоретическая информатика” (Россия, 2009).

4. Международный семинар Franco-Russian workshop on Algorithms, complexity and applications (Россия, 2010).

5. Международная конференция Computability in Europe (Португалия, 2010).

6. Международный семинар Exact Complexity of NP-hard Problems (Германия, 2010).

7. Семинар Университета Киото (Япония, 2010).

8. Международный семинар Estonian Theory Days (Эстония, 2011).

9. Международная конференция International Symposium on Mathematical Foundations of Computer Science (Польша, 2011).

10. Международная конференция Computability in Europe (Англия, 2012).

11. Международная конференция International Computer Science Symposium in Russia (Россия, 2012).

12. Международный семинар SAT Interactions (Германия, 2012).

13. Семинар Математического института Чешской академии наук (Прага, 2013).

14. Международный семинар Optimal algorithms and proofs (Германия, 2014).

15. Семинар Университета Калифорнии в Сан-Диего (США, 2015).

16. Семинар Курантовского института математических наук (США, 2015).

17. Международный семинар Connections Between Algorithm Design and Complexity Theory (США, 2015).

18. Семинар Уральского федерального университета (Россия, 2015).

19. Международный семинар Problems in Theoretical Computer Science (Россия, 2015).

20. Международная конференция Annual Innovations in Theoretical Computer Science (США, 2016).

21. Международный семинар Low-Depth Complexity Workshop (Россия, 2016).

22. Международная конференция International Symposium on Mathematical Foundations of Computer Science (Польша, 2016).

23. Международная конференция Annual IEEE Symposium on Foundations of Computer Science (США, 2016).

Степень достоверности

Результаты исследований отражены в 11 работах, опубликованных в изданиях, индексируемых международными базами данных (MathSciNet, Scopus): [31, 12, 30, 13, 15, 33, 14, 19, 21, 22, 20].

Глава 1

Нижние оценки

Дополнительные определения и обозначения

Типы операций и элементов

Определение 5 (операция элемента)**.** *Бинарную булеву функции, вычисляющуюся в элементе схемы, мы будем называть* операцией*, чтобы отличать её от функции от всех входных n битов, которая вычисляется в данном элементе.*

Определение 6 (типы операций)**.** *Все возможные 16 бинарных операций $b(x, y)$ обычно классифицируются следующим образом:*

- *две* константные *или* тривиальные операции*: 0, 1;*
- *четыре* вырожденные *или* проходные операции*: x, $\overline{x}$, y, $\overline{y}$;*
- *восемь* операций типа $\wedge$*: $(x \oplus a) \wedge (y \oplus b) \oplus c$ при $a, b, c \in \mathbb{F}_2$;*
- *две* операции типа $\oplus$*: $x \oplus y \oplus c$ при $c \in \mathbb{F}_2$.*

Данное определение естественным образом распространяется и на элементы: будем говорить, что элемент имеет некоторый тип, если в нём вычисляет-

ся операция такого типа. Таким образом, схемы над базисом U_2 не содержат элементов типа $\oplus$.

Определение 7 (аффинная схема)**.** Аффинной схемой *будем называть схему, которая не содержит элементов типа* $\wedge$*. Каждый элемент такой схемы вычисляет аффинную функцию от входных переменных.*

Определение 8 (минимальный элемент)**.** *В любой непустой схеме найдётся элемент, зависящий только от входных элементов (поскольку граф схемы является ациклическим). Каждый такой элемент будем называть* минимальным*.*

Определение 9 (k-элемент)**.** k-элементом *называется элемент исходящей степени ровно* k*, а* k^+-элементом *называется элемент, из которого выходит хотя бы* k *проводов. На некоторых рисунках работы мы будем подписывать исходящую степень элемента над ним. Обозначать исходящую степень мы будем через* outdeg*.*

Определение 10 (подстановка)**.** *Пусть переменными функции* $f \in B_n$ *являются* $x_1, x_2, \ldots, x_n$*, а функция* $\rho \in B_{n-1}$ *зависит от тех же переменных, кроме* x_i*. Под* $f|_{x_i \leftarrow \rho} \in B_{n-1}$ *мы будем понимать функцию, определяемую следующим образом:*

$$f|_{x_i \leftarrow \rho}(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = \\ = f(x_1, x_2, \ldots, x_{i-1}, \rho(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n), x_{i+1}, \ldots, x_n).$$

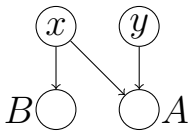
Метод элиминации элементов

Единственным известным на данный момент методом получения нижних оценок на размер произвольных схем (без ограничений на глубину или исходящую степень) является метод элиминации элементов. Чтобы проиллю-

стрировать его, мы приводим доказательство нижней оценки $2n-O(1)$, представленной К. Шнорром в 1974 г. [47]. Функция $\mathrm{MOD}_n^{3,r} \in B_n$ выдаёт единицу тогда и только тогда, когда сумма (над целыми числами) входных n битов сравнима с r по модулю 3. Формальное определение функции $\mathrm{MOD}_n^{m,r} \in B_n$ таково:

$$\mathrm{MOD}_n^{m,r}(x_1, \ldots, x_n) = 1 \Leftrightarrow \sum_{i=1}^{n} x_i \equiv r \pmod{m}.$$

Мы покажем, что для вычисления $\mathrm{MOD}_n^{3,r}$ понадобятся схемы размера хотя бы $2n-6$, индукцией по n. База $n \leq 3$ выполняется. Для перехода рассмотрим оптимальную схему $\mathcal{C}$, вычисляющую $\mathrm{MOD}_n^{3,r}$, и её минимальный элемент A. Такой есть, поскольку при $n \geq 4$ функция $\mathrm{MOD}_n^{3,r}$ не является константой. Пусть x и y суть входы A. Простым, но ключевым замечанием является следующее: хотя бы один из x и y входит в хотя бы ещё один элемент. Действительно, если из x и y провода идут только в A, тогда вся схема зависит от x и y только через A. Это, в частности, означает, что при подстановке констант входам x и y четырьмя разными способами ($(x, y) = (0,0), (0,1), (1,0), (1,1)$) получается не более двух различных подфункций, в то время как должно получиться хотя бы три: $\mathrm{MOD}_{n-2}^{3,0}$, $\mathrm{MOD}_{n-2}^{3,1}$, и $\mathrm{MOD}_{n-2}^{3,2}$ (они попарно различны при $n \geq 4$). Предположим, что хотя бы ещё один провод выходит из x и обозначим через B соответствующий элемент.



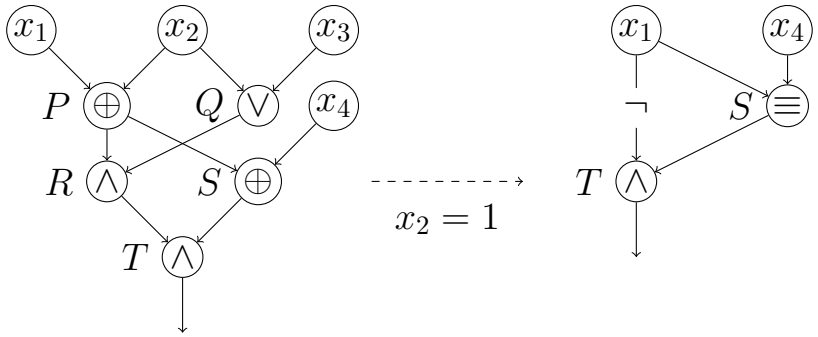
Подставим в x константу 0. Это удаляет хотя бы два элемента из схемы (A и B): если один из входов элемента вычисляет константу, то сам элемент вычисляет либо константу, либо унарную функцию от другого входа. Полученная схема вычисляет $\mathrm{MOD}_{n-1}^{3,r}$, и нижняя оценка получается по индукции. Наилучшей известной на данный момент нижней оценкой для $\mathrm{MOD}_n^{3,r}$ является оценка $2.5n - O(1)$, доказанная Л. Стокмайером [53] в 1977 г. Далее

в данной работе мы также докажем верхнюю оценку $3n + O(1)$, которая на данный момент является самой сильной из известных.

Отметим, что в оригинальной работе К. Шнорра оценка доказывается не для функции MOD_n^3, а для широкого класса функций, имеющих хотя бы три различные подфункции относительно любых двух переменных.

Упрощение схемы при подстановках

Опишем процесс удаления элементов чуть подробнее. Ниже приведён пример подстановки и упрощения схемы.



Входу x_2 подставляется значение 1. Тогда Q вычисляет константу 1, поэтому P и R вычисляют $x_1 \oplus 1$. Эти три элемента 3 могут быть удалены из схемы. После этого S вычисляет $(x_1 \oplus 1) \oplus x_4$, то есть $x_1 \equiv x_4$, в то время как T вычисляет $(x_1 \oplus 1)S$. Знак отрицания на проводе из x_1 в T отражает тот факт, что функция, вычисляющаяся в T — не просто xy (как на рисунке), а $(x \oplus 1)y$.

Ниже мы резюмируем несколько простых, но важных фактов, проиллюстрированных данным примером.

- Подстановка $x_2 = 1$ тривиализирует элемент Q (то есть делает его константой), поэтому не только сам Q удаляется, но и все его потомки. В то же время P не тривиализируется, а становится проходным (зависит существенно лишь от одного из своих входов). Это показывает разницу

между элементами типа $\oplus$ и $\wedge$ и объясняет, почему для схем над базисом U_2 удаётся доказывать более сильные нижние оценки.

- При упрощении схемы после подстановки может понадобиться изменить функции, вычисляющиеся в элементах (при этом тип функции никогда не меняется).

- Полученная схема не зависит ни от x_2, ни от x_3, хотя был подставлен только вход x_2.

Определение 11 (упрощение схемы)**.** Упрощением схемы *будем называть процесс удаления тривиальных и проходных элементов. Схему, в которой таких элементов нет, будем называть* упрощённой *или* нормализованной*.*

Меры сложности схем

В работе мы будем использовать различные меры сложности схем для получения нижних оценок на размер схем. Перед тем, как формально определить понятие меры сложности схем, рассмотрим следующие числовые характеристики схем:

1. `gates` — число внутренних элементов;

2. `inputs` — число входных элементов;

3. $\texttt{inputs}_1$ — число входных элементов исходящей степени ровно один;

4. `ands` — число элементов типа $\wedge$;

5. `xors` — число элементов типа $\oplus$.

Определение 12 (мера сложности схем)**.** *Функция μ, присваивающая каждой схеме неотрицательное вещественное число, называется* мерой сложности схем*, если для любой схемы $\mathcal{C}$ упрощение схемы не увеличивает $\mu(\mathcal{C})$.*

Лемма 13. *Следующие функции являются мерами сложности схем:*

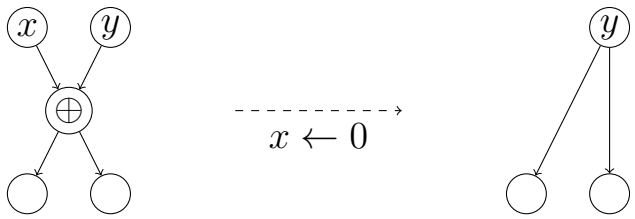
1. $\texttt{gates}$;

2. $\texttt{gates} + \alpha\texttt{inputs}$, *где* $\alpha > 0$ — *константа;*

3. $\alpha\texttt{xors} + \beta\texttt{ands}$, *где* $\alpha, \beta > 0$ — *константы;*

4. $\texttt{gates} + \alpha\texttt{inputs} - \sigma\texttt{inputs}_1$, *где* $\alpha > 0, \min\{1, \alpha\} \geq \sigma > 0$ — *константы.*

Доказательство. Истинность первых двух утверждений следует непосредственно из определения. Третья функция является мерой, поскольку при удалении проходного элемента в процессе упрощения схемы могут поменяться операции его потомков, но не может поменяться их тип. Чтобы убедиться, что четвёртая функция является мерой, заметим, что она всегда неотрицательна, поскольку $\texttt{inputs}_1 \leq \texttt{inputs}$ и $\sigma \leq \alpha$. В процессе упрощения схемы может удалиться проходной элемент, что может привести к увеличению исходящей степени входного элемента:



Мера при этом не увеличивается, поскольку $\sigma \leq 1$. □

Определение меры сложности схем естественным образом распространяется на функции: если μ — мера сложности схем, а f — булева функция, то $\mu(f)$ — это минимальное значение $\mu(\mathcal{C})$ среди всех схем $\mathcal{C}$, вычисляющих f.

1.1 Нижняя оценка $7n/3 - O(1)$ для функций высокой степени

1.1.1 Мультипликативная сложность и полиномы над $\mathbb{F}_2$

Определение 14 (мультипликативная сложность)**.** Мультипликативной сложностью $\texttt{mult}(f)$ *функции* f *называется минимальное число* $\wedge$*-элементов в схеме, вычисляющей* f*.*

Для каждой функции $f \in B_n$ существует единственный вычисляющий её мультилинейный многочлен от n переменных над $\mathbb{F}_2$, то есть сумма (по модулю 2) мономов (конъюнкций) переменных. Такой многочлен известен как многочлен Жегалкина. Построить его можно так: возьмём сумму всех "литеральных" мономов, то есть конъюнкций литералов (где литерал — это переменная или её отрицание), соответствующих элементам $f^{-1}(1)$. Например, для функции голосования из B_3 получим такой многочлен:

$$x_1x_2x_3 \oplus (1 \oplus x_1)x_2x_3 \oplus x_1(1 \oplus x_2)x_3 \oplus x_1x_2(1 \oplus x_3) = x_1x_2 \oplus x_2x_3 \oplus x_1x_3\,.$$

Хорошо известно, что такое представление единственно. Действительно, у каждой функции из B_n такое представление есть, а всего различных мультилнейных многочленов над $\mathbb{F}_2$ столько же, сколько и функций в B_n (то есть 2^{2^n}). Будем обозначать этот многочлен через $\chi(f)$. Важной характеристикой функции f является степень многочлена $\chi(f)$, которую мы обозначать через $\deg(f)$. Интуитивно ясно, что если в схеме мало $\wedge$-элементов, то она не сможет вычислить функцию высокой степени (это верно, потому что мы работаем с полем размера два и мультилинейными многочленами; для вычисления же, например, x^n достаточно около $\log n$ умножений). Формально это показано К. Шнорром в 1988 г.

Лемма 15 ([49]). *Любая схема, вычисляющая функцию $f \in B_n$, содержит хотя бы* $\deg(f)-1$ *элементов типа* $\wedge$. *Другими словами,* $\texttt{mult}(f) \geq \deg(f)-1$.

Данная лемма даёт нижнюю оценку на мультипликативную сложность. В частности, если степень функции равна n, то её мультипликативная сложность не меньше $n-1$. Интересно отметить, что это самая сильная из известных на данный момент нижних оценок. Не доказана даже нижняя оценка n, не говоря уже об оценках вида $(1+\varepsilon)n$ для $\varepsilon > 0$, хотя неконструктивно несложно доказать экспоненциальную нижнюю оценку.

1.1.2 Доказательство нижней оценки

В данном разделе мы доказываем нижнюю оценку $\frac{7n}{3} - \Theta(1)$. Доказательство такое же простое, как и оригинальное доказательство К. Шнорра нижней оценки $2n - \Theta(1)$.

Определение 16. *Для константы k и $n \geq k$ обозначим через S_n^k класс всех функций $f \in B_n$, удовлетворяющих следующим свойствам:*

1. *при подстановке констант любым двум переменным получаются хотя бы три различные подфункции f;*
2. *для любой переменной x_i и константы $c \in \mathbb{F}_2$, $f|_{x_i \leftarrow c} \in S_{n-1}^k$.*

Естественной функцией, лежащем в данном классе, является $\mathrm{MOD}_n^{m,r}$. Нетрудно видеть, что для любых $m \geq 3$ и r верно $\mathrm{MOD}_n^{m,r} \in S_n^{m+2}$. Действительно,

$$\begin{aligned} \mathrm{MOD}_n^{m,r}|_{x_i \leftarrow 0, x_j \leftarrow 0} = \mathrm{MOD}_{n-2}^{m,r},\ \mathrm{MOD}_n^{m,r}|_{x_i \leftarrow 1, x_j \leftarrow 1} &= \mathrm{MOD}_{n-2}^{m,r-2}, \\ \mathrm{MOD}_n^{m,r}|_{x_i \leftarrow 0, x_j \leftarrow 1} = \mathrm{MOD}_n^{m,r}|_{x_i \leftarrow 1, x_j \leftarrow 0} &= \mathrm{MOD}_{n-2}^{m,r-1}. \end{aligned}$$

Для $m \geq 3$ и $n \geq m+2$, $\mathrm{MOD}_{n-2}^{m,r}$, $\mathrm{MOD}_{n-2}^{m,r-1}$, $\mathrm{MOD}_{n-2}^{m,r-2}$ не являются константами и попарно различны (отметим, что для $m = 2$ это неверно, поскольку $\mathrm{MOD}_{n-2}^{2,r-2} = \mathrm{MOD}_{n-2}^{2,r}$). Более того, $\mathrm{MOD}_{n}^{m,r}|_{x_i \leftarrow c} = \mathrm{MOD}_{n-1}^{m,r-c}$.

Для доказательства заявленной нижней оценки мы будем использовать следующую меру сложности схем:

$$\mu(\mathcal{C}) = 3\mathtt{xors}(\mathcal{C}) + 2\mathtt{ands}(\mathcal{C})\,,$$

где $\mathtt{xors}(\mathcal{C})$ и $\mathtt{ands}(\mathcal{C})$ обозначают, соответственно, число $\oplus$-элементов и $\wedge$-элементов в схеме $\mathcal{C}$.

Лемма 17. *Для любой схемы $\mathcal{C}$, вычисляющей функцию $f \in S_n^k$,*

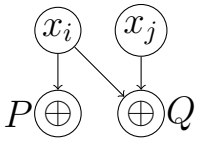
$$\mu(\mathcal{C}) \geq 6(n-k-1)\,.$$

Доказательство. Доказательство проведём индукцией по n. Базовый случай $n \leq k+1$ выполняется. Для доказательства перехода предположим, что $n > k+1$, и возьмём оптимальную относительно μ схему $\mathcal{C}$, вычисляющую f. Ниже мы покажем, что обязательно найдётся переменная, которой можно присвоить константу, так чтобы мера μ уменьшилась хотя бы на 6. Поскольку полученная функция принадлежит классу S_{n-1}^k, необходимая оценка следует по предположению индукции. Отметим, что получающаяся функция не может быть константой (в противном случае у неё не было бы трёх различных подфункций относительно любых двух переменных). Это позволяет заключить, что если какой-то элемент схемы становится константой при подстановке, то этот элемент не может быть выходным и, следовательно, удалится ещё хотя бы один его потомок.

Все константные и вырожденные элементы можно удалить из $\mathcal{C}$, не увеличив при этом $\mu(\mathcal{C})$. Рассмотрим минимальный элемент Q схемы $\mathcal{C}$ и его входы x_i и x_j. Данные переменные различны, поскольку Q не является вы-

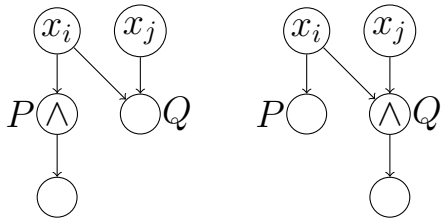
рожденным. Поскольку функция f имеет хотя бы три различные подфункции относительно переменных x_i и x_j, из хотя бы одной из этих переменных должен выходить ещё хотя бы один провод. Пусть провод идёт из x_i в элемент $P \neq Q$.

Случай 1. И P, и Q суть $\oplus$-элементы (см. рис. ниже; на рисунке показаны только типы элементов, а не конкретные операции, вычисляющиеся в них).

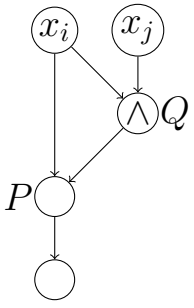


Подставим константу вместо x_i — оба элемента удалятся и μ уменьшится хотя бы 6.

Случай 2. Один из P и Q является $\wedge$-элементом.



Подставим в x_i константу, тривиализирующую этот элемент. Это удалит P и Q, а также хотя бы одного из их потомков. Отметим, что P сам может оказаться потомком Q.



В этой ситуации после подстановки константы в x_i тривиализируется не только элемент Q, но и элемент P, поскольку оба его входа стали константами. Значит, удалятся и все потомки P. Таким образом, в любом случае удаляются хотя бы три элемента, что уменьшает μ хотя бы на 6.

□

Теорема 18 ([30]). *Пусть* $f \in S_n^k$ *и* $\deg(f) = n$. *Тогда*

$$\mathtt{gates}(f) \geq \frac{7n}{3} - 2k - 3\,.$$

Доказательство. Рассмотрим схему $\mathcal{C}$ для f. Число элементов в $\mathcal{C}$ не меньше $\mathtt{xors}(\mathcal{C}) + \mathtt{ands}(\mathcal{C})$. Сложив следующие два неравенства, первое из которых следует из леммы 17, а второе — из леммы 15, получаем необходимое неравенство:

$$\begin{aligned} 3\mathtt{xors}(\mathcal{C}) + 2\mathtt{ands}(\mathcal{C}) &\geq 6n - 6(k+1)\,, \\ \mathtt{ands}(\mathcal{C}) &\geq n - 1\,. \end{aligned}$$

□

Данная теорема даёт нижнюю оценку $\frac{7n}{3} - O(1)$ для функций из S_n^k степени n, но её можно использовать, на самом деле, даже для ещё более широкого класса функций. Допустим, что $f \in S_n^k$ и $\deg(f) < n$. Возьмём *произвольную* функцию $g \in B_{n-1}$ степени $n-1$ (число таких функций есть $2^{2^{n-1}-1}$, поскольку всё, что требуется, это присутствие монома $x_1 x_2 \ldots x_{n-1}$ в $\chi(g)$) и определим функцию $h \in B_n$ следующим образом:

$$h(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \oplus x_n g(x_1, \ldots, x_{n-1})\,.$$

Рассмотрим схему $\mathcal{C}$ для h. Поскольку $\deg(h) = n$, $\mathtt{ands}(\mathcal{C}) \geq n - 1$. Заметим

также, что $h|_{x_n \leftarrow 0} = f|_{x_n \leftarrow 0} \in S^k_{n-1}$. Следовательно,

$$\mu(\mathcal{C}) \geq \mu(\mathcal{C}|_{x_n \leftarrow 0}) \geq 6(n-1) - 6(k+1),$$

что даёт нижнюю оценку $\frac{7n}{3} - c(k)$ на размер $\mathcal{C}$.

1.2 Нижняя оценка $3n - o(n)$ для аффинных дисперсеров сублинейной размерности

Как было продемонстрировано в предыдущих разделах, критическим случаем в доказательствах нижних оценок, основанных на методе элиминации элементов, является ситуация, когда рассматриваемая переменная входит в два $\oplus$-элемента. В такой ситуации мы не можем подставить константу данной переменной так, чтобы сделать один из этих элементов константой. Чтобы найти хорошую подстановку и в таком проблемном случае, мы будем использовать линейные подстановки в данном разделе. Для этого, в свою очередь, нам понадобится функция, устойчивая относительно таких подстановок.

1.2.1 Аффинные дисперсеры

Определение 19 (аффинный дисперсер)**.** *Функция $f \in B_n$ называется* аффинным дисперсером *размерности d, если она не обращается в константу ни на каком аффинном подпространстве $\mathbb{F}_2^n$ размерности хотя бы d.*

Понятие дисперсера является обобщением понятия экстрактора — функции, которая получает на вход последовательность битов в соответствии с некоторым распределением и выдаёт бит, распределение которого статистически близко к равномерному. В отличие от экстракторов, дисперсерам требуется выдавать просто неконстантный бит. Чтобы задать распределение, обычно задают класс источников $\mathcal{F}$, где каждое $X \in \mathcal{F}$ — это распределение

на $\mathbb{F}_2^n$. Поскольку дисперсеры выдают просто неконстантный бит, мы отождествляем X с его носителем на $\mathbb{F}_2^n$. Функция $f \in B_n$ называется дисперсером для класса источников $\mathcal{F}$, если $|f(X)| = 2$ для любого $X \in \mathcal{F}$. Поскольку существуют источники (даже с почти полной энтропией), из которых невозможно извлечь даже один неконстантный бит, изучаются различные частные случаи источников: см. обзор Р. Шалтиела [51]. В данной работе мы будем использовать аффинные источники и их обобщение — источники для полиномиальных многообразий. Аффинные дисперсеры изучаются очень активно в последнее время. В частности, были построены явные конструкции аффинных дисперсеров для размерности $d = o(n)$: [5, 61, 36, 51, 4]. Дисперсеры для полиномиальных многообразий в полях большого размера изучались в З. Двиром [17], дисперсеры для $\mathbb{F}_2$ изучались Г. Коэном и А. Талем [11].

1.2.2 Доказательство нижней оценки

В данном разделе доказывается следующая нижняя оценка.

Теорема 20 ([13])**.** *Пусть $f \in B_n$ — аффинный дисперсер размерности d. Тогда для любой схемы $\mathcal{C}$, вычисляющей f, верно*

$$\texttt{gates}(\mathcal{C}) \geq 3n - 4d\,.$$

В частности, число элементов в схеме, вычисляющей аффинный дисперсер сублинейной размерности, не меньше $3n - o(n)$.

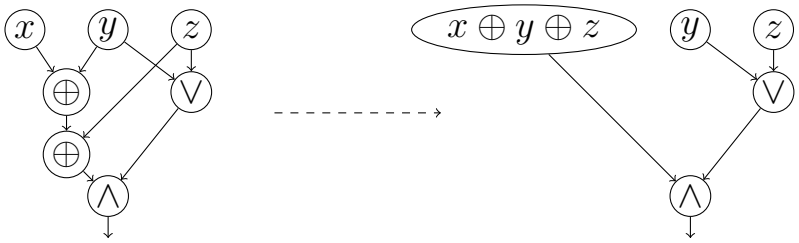
Для доказательства нижней оценки будет удобно использовать так называемые XOR-схемы.

Определение 21 (XOR-схема)**.** XOR-схемой *называется схема над базисом B_2, в которой в качестве входов разрешено использовать не только переменные, но и произвольные линейные функции от входных переменных.*

Рассмотрим следующую меру сложности XOR-схемы $\mathcal{C}$:

$$\mu(\mathcal{C}) = \texttt{gates}(\mathcal{C}) + \texttt{inputs}(\mathcal{C}),$$

где, как обычно, `gates` и `inputs` обозначают, соответственно, число внутренних и входных элементов. Нетрудно видеть, что если минимальный элемент XOR-схемы $\mathcal{C}$ имеет тип $\oplus$, то его можно заменить на входной элемент (новый или старый). При этом мера μ не увеличится.



Теорема 20 является непосредственным следствием следующей леммы (при $D = \mathbb{F}_2^n$).

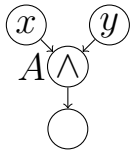
Лемма 22. *Пусть $f \in B_n$ — афинный дисперсер размерности d, $S \subseteq \mathbb{F}_2^n$ — аффинное подпространство $\mathbb{F}_2^n$ размерности D, а $\mathcal{C}$ — схема, вычисляющая f на S (то есть для любого $x \in S$, $\mathcal{C}(x) = f(x)$). Тогда*

$$\mu(\mathcal{C}) \geq 4 \cdot (D - d - 1)\,.$$

Доказательство. Доказательство проведём индукцией по D. База $D \leq d+1$ выполнена. Для перехода рассмотрим схему $\mathcal{C}$ с минимальным значением μ. Пусть A — минимальный элемент, входами которого являются линейные функции x и y (такой элемент точно есть, поскольку f на S не может считать линейную функцию, так как $D > d + 1$). Если A имеет тип $\oplus$, то его можно заменить на вход (не увеличив при этом μ), поэтому допустим, что A вычисляет произведение, то есть $(x \oplus c_1)(y \oplus c_2) \oplus c$, где $c_1, c_2, c \in \mathbb{F}_2$. Мы сделаем подстановку $x \leftarrow c_1$ или $y \leftarrow c_2$. (Делать такие подстановки будем

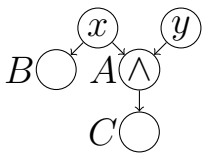
точно так же, как и для обычных схем: удалим все элементы, хотя бы один из входных проводов которых начал вычислять константную функцию.) Это даёт аффинное подпространство $\mathbb{F}_2^n$ размерности хотя бы $D-1$ (если бы размерность упала до нуля, это означало бы, что x или y вычисляют константу на S, что противоречило бы оптимальности схемы). Ниже мы покажем, что при подстановке μ уменьшится хотя бы на 4. Для этого мы рассмотрим два случая.

Случай 1. Исходящая степень x и y равна 1 (а исходящая степень y может быть 1 или больше).



Подставим $x \leftarrow c_1$. Это тривиализирует A до c, поэтому всего его потомки тоже удаляются. У A обязан быть хотя бы один потомок, поскольку если бы A оказался выходным элементом, то вся схема стала бы константой на аффинном подпространстве размерности хотя бы d. Таким образом, при подстановке удаляются два элемента, а также пропадает зависимость от двух входов (x и y), поэтому μ уменьшается хотя бы на 4.

Случай 2. Исходящая степень, скажем, x хотя бы два.



Пусть B — другой потомок x и пусть C потомок A. Подставим $x \leftarrow c_1$. Это удалит вход x и элементы A, B и C. Если $B = C$, то C также становится константой (поскольку оба его входа вычисляют константы после подстановки), поэтому удалятся и его потомки. Таким образом, в

этом случае удаляется хотя бы один вход и хотя бы три элемента, поэтому μ снова уменьшается хотя бы на 4.

□

В заключение раздела интересно отметить, что константу 4 в доказанном неравенстве $\texttt{gates}(\mathcal{C}) + \texttt{inputs}(\mathcal{C}) \geq 4 \cdot (n - d - 1)$ улучшить не получится: функция скалярного произведения, определяемая как

$$\mathrm{IP}(x_1, y_1, x_2, y_2, \ldots, x_{n/2}, y_{n/2}) = x_1 y_1 \oplus x_2 y_2 \oplus \cdots \oplus x_{n/2} y_{n/2}\,,$$

является аффинным дисперсером размерности $n/2 + 1$ (см., например, [10, теорема A.1]) и имеет схемную сложность $n - 1$.

1.3 Нижняя оценка $(3 + \frac{1}{86})n - o(n)$ для аффинных дисперсеров сублинейной размерности

Основным результатом данного раздела является усиление оценки из предыдущего раздела до $\left(3 + \frac{1}{86}\right) n - o(n)$.

Теорема 23 ([19])**.** *Пусть $f \in B_n$ — аффинный дисперсер сублинейной размерности. Тогда*

$$\texttt{gates}(f) \geq \left(3 + \frac{1}{86}\right) n - o(n)\,.$$

На сегодняшний день это самая сильная известная нижняя оценка на схемную сложность функции из NP. Её доказательство также является наиболее технически трудным в данной работе.

1.3.1 Схемы с циклами

В данном разделе нам понадобится другое обобщение схем. Это необходимо для того, чтобы оставаться в нужном классе схем в процессе элиминации

элементов.

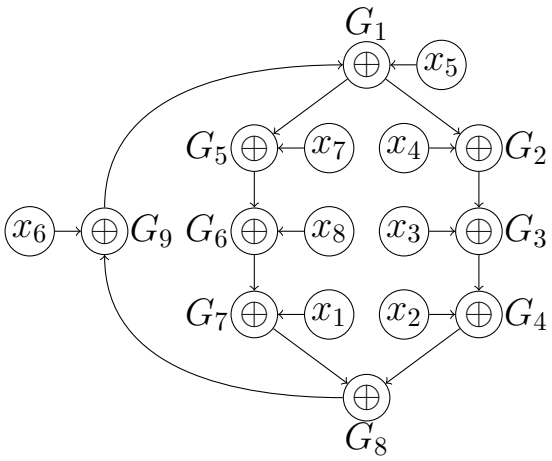
Определение 24 (схема с циклами)**.** Схемой с циклами *называется ориентированный граф, необязательно являющийся ациклическим, в котором все вершины имеют входящую степень* 0 *или* 2*. Размер схемы, входные и внутренние элементы определяются так же, как и для обычных схем.* Аффинной схемой с циклами *называется схема с циклами, в которой каждый элемент вычисляет аффинную операцию. Как правило, все элементы такой схемы вычисляют операции* $\oplus$ *и* $\equiv$*, но по техническим причинам мы допускаем также проходные и тривиальные элементы. Мы будем рассматривать схемы с несколькими выходами, поэтому, в отличие от обычных схем, вычисляющих булевы предикаты, у аффинной схемы с циклами в качестве выходных может быть помечено несколько элементов и они не обязаны иметь исходящую степень, равную нулю.*

Аффинная схема с циклами соответствует системе уравнений над $\mathbb{F}_2$. Переменные этой системы — значения элементов. Операция внутреннего элемента задаёт уравнение на входные и выходное значение элемента. Входные элементы учитываются в столбце свободных членов (поскольку нас будут интересовать решения системы уравнений в ситуации, когда входным элементам присвоены константные значения). Таким образом, мы имеем отдельную систему уравнений для каждого набора значений входных элементов, но у всех этих систем общая матрица. Если в элемент G ведут провода из F и H и этот элемент вычисляет операцию $\odot$, мы записываем уравнение $G \oplus (F \odot H) = 0$. Например, если G вычисляет $F \oplus x \oplus 1$, где x — входной элемент, тогда соответствующим уравнением будет $G \oplus F = x \oplus 1$; в данном случае элементы G и F соответствуют двум единицам в матрице, а $x \oplus 1$ — это правая часть уравнения. Для аффинной схемы с циклами соответствующая система уравнений имеет квадратную матрицу.

Определение 25 (честная аффинная схема с циклами)**.** *Мы будем называть аффинную схему с циклами* честной*, если её матрица имеет полный ранг.*

Из этого следует, что для любого набора входных значений найдётся *единственный* набор значений всех элементов системы, соответствующей честной схеме.. Таким образом, как и обычная аффинная схема, такая схема вычисляет в каждом элементе некоторую функцию от входов (нетрудно видеть, что эта функция будет аффинной).

Пример циклической аффинной схемы приведён ниже.



Данной схеме соответствует такая система уравнений:

$$
\begin{array}{ll}
G_1 = G_9 \oplus x_5 & \\
G_2 = G_1 \oplus x_4 & \\
G_3 = G_2 \oplus x_3 & \\
G_4 = G_3 \oplus x_2 & \\
G_5 = G_1 \oplus x_7 & \\
G_6 = G_5 \oplus x_8 & \\
G_7 = G_6 \oplus x_1 & \\
G_8 = G_4 \oplus G_7 & \\
G_9 = G_8 \oplus x_6 &
\end{array}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}
\times
\begin{bmatrix}
G_1 \\ G_2 \\ G_3 \\ G_4 \\ G_5 \\ G_6 \\ G_7 \\ G_8 \\ G_9
\end{bmatrix}
=
\begin{bmatrix}
x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_7 \\ x_8 \\ x_1 \\ 0 \\ x_6
\end{bmatrix}
$$

Решив данную систему, получаем такие аффинные функции, вычисляющиеся в элементах данной схемы:

$$G_1 = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_8$$
$$G_2 = x_1 \oplus x_2 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_8$$
$$G_3 = x_1 \oplus x_2 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_8$$
$$G_4 = x_1 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_8$$
$$G_5 = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_8$$
$$G_6 = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6$$
$$G_7 = x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6$$
$$G_8 = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_8$$
$$G_9 = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_7 \oplus x_8$$

1.3.1.1 Связь между циклическими и ациклическими аффинными схемами

Нетрудно показать, что для функций с несколькими выходами циклические аффинные схемы представляют собой более сильную вычислительную модель, чем обычные аффинные схемы. Например, девять аффинных функций, вычисляющихся циклической схемой, показанной выше, не могут быть вычислены ациклической схемой размера девять. Чтобы показать это, предположим противное. Поскольку в схеме девять элементов и она должна вычислить девять различных аффинных функций, каждый элемент должен вычислять некоторый выход. Рассмотрим минимальный элемент схемы (он есть, поскольку схема ациклична). Этот элемент вычисляет сумму двух переменных, поэтому не может вычислять ни одну из выходных функций.

С другой стороны, минимальный размер аффинной схемы, как циклической, так и ациклической, вычисляющей сумму k переменных, равен $k - 1$.

1.3.1.2 Полусхемы

Для доказательства нижней оценки нам понадобится понятие *полусхемы*, обобщающее как обычные схемы, так и циклические аффинные схемы.

Определение 26 (полусхема)**.** Полусхема — *это композиция аффинной циклической схемы и обычной схемы. Её элементы разделяются на два множества — X и C. Элементы из X образуют циклическую аффинную схему. Элементы из C образуют обычную схему, входами которой являются элементы из X. В схеме нет проводов из C в X. Полусхема называется* честной*, если X является честной.*

В оставшейся части раздела 1.3 под схемой мы будем понимать честную полусхему.

1.3.2 Преобразования циклических схем

На протяжении всего раздела мы будем работать с честными полусхемами. Поскольку это новое понятие, в следующем подразделе мы аккуратно описываем возможные преобразования таких схем.

1.3.2.1 Базовые преобразования

В данном разделе мы рассматриваем различные типа подстановок. Самая базовая подстановка — это подстановка константы вместо переменной.

Утверждение 27. *Пусть $\mathcal{C}$ — схема со входами $x_1, \ldots, x_n$ и пусть $c \in \mathbb{F}_2$. Для любого элемента G, в который идёт провод из x_1 заменим операцию $g(x_1, t)$, вычисляющуюся в G, на операцию $g'(x_1, t) = g(c, t)$ (таким образом, новая операция не зависит от x_1). Это преобразует схему $\mathcal{C}$ в схему $\mathcal{C}'$ (в частности, она по-прежнему является честной полусхемой) такого же размера, той же топологии и вычисляющую то же самое: для каждого*

элемента H, вычисляющего функцию $h(x_1, \ldots, x_n)$ в $\mathcal{C}$, соответствующий элемент в $\mathcal{C}'$ вычисляет функцию $h(c, x_2, \ldots, x_n)$.

Будем называть это преобразование *подстановкой константы*. Более сложный тип подстановки — это подстановка переменной x функции, вычисляющейся в элементе схемы G. В таком случае каждый провод из x мы заменяем на провод из G. Такое преобразование назовём *подстановкой функции*.

Утверждение 28. *Пусть $\mathcal{C}$ — схема со входами $x_1, \ldots, x_n$ и пусть $g(x_2, \ldots, x_n)$ — функция, вычисляющаяся в элементе G. Рассмотрим конструкцию $\mathcal{C}'$, полученную подстановкой g в x_1 (такого же размера). Тогда если G не достижим из x_1 по ориентированному пути в $\mathcal{C}$, то $\mathcal{C}'$ — честная полусхема, и для каждого элемента H, вычисляющего функцию $h(x_1, \ldots, x_n)$ в $\mathcal{C}$, за исключением x_1, соответствующий элемент в схеме $\mathcal{C}'$ вычисляет функцию $h(g(x_2, \ldots, x_n), x_2, \ldots, x_n)$.*

Доказательство. В утверждении требуется, чтобы G не был достижим из x_1 (чтобы не появилось новых циклов) и чтобы g не зависел от x_1. Функции, вычисляющиеся в элементах, являются решением соответствующей системы уравнений. Преобразование заменяет каждое уравнение $H = F \odot x_1$ на уравнение $H = F \odot G$ (и уравнение $H' = x_1 \odot x_1$ на уравнение $H' = G \odot G$).

Чтобы доказать, что $\mathcal{C}'$ является честной полусхемой, мы покажем, что для любого набора входных значений есть единственный набор значений элементов $\mathcal{C}'$, совместный со входными значениями. Зафиксируем значения $x_2, \ldots, x_n$. Допустим, что решение старой системы не удовлетворяет какому-то новому уравнению. Тогда $x_1 = g(x_2, \ldots, x_n)$ нарушает соответствующее старое уравнение — противоречие. И обратно: пусть есть два решения новой системы. Они должны удовлетворять старой системе (где $x_1 = g(x_2, \ldots, x_n)$), но у старой системы есть ровно одно решение. □

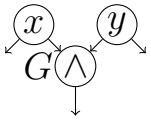
В дальнейшем мы будем также использовать подстановки, которые не удовлетворяют условию это предложения — подстановки, создающие новые циклы. Они будут описаны в подразделе 1.3.2.3.

1.3.2.2 Нормализация и специальные элементы

При анализе случаев нам будет удобно предполагать, что текущая схема нормализована, то есть не содержит очевидных неэффективностей (таких как константные или проходные элементы). В данном подразделе мы описываем несколько правил нормализации. Для каждого правила мы аккуратно покажем, что при применении этого правила схема остаётся честной и вычисляет ту же функцию. Мы также будем следить за количеством так называемых *специальных* элементов в схеме. Число таких элементов будет учитываться в мере сложности схем, используемой в доказательстве нижней оценки.

Определение 29 (специальный элемент)**.** *Внутренний элемент G называется* специальным*, если он удовлетворяет следующим трём условиям:*

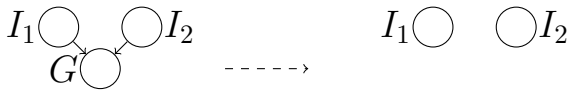
- *G — 1-элемент типа $\wedge$,*
- *оба провода в G идут из входных переменных,*
- *обе эти переменные имеют исходящую степень ровно два.*



В процессе элиминации элементов мы будем пользоваться только правилами нормализации для удаления элементов. Ниже мы покажем, что при удалении элемента появляется не более четырёх новых специальных элементов.

Нормализованной мы будем называть схему, к которой не применимо ни одно из правил, перечисленных ниже. Каждое из этих правил удаляет элемент G, в который идут провода из элементов I_1 и I_2 (I_1 и I_2 могут быть входными или внутренними элементами; мы также рассмотрим вырожденный случай, когда один из них совпадает с G).

Правило 1: если G является 0-элементом и не является выходом, удалить его.



Отметим также, что ситуация, когда из G выходит один провод и он ведёт в G, невозможна, поскольку этому соответствовало бы тривиальное уравнение, что противоречило бы честности схемы.

Правило 2: если G вычисляет константную функцию c от входных переменных (здесь мы имеем в виду именно функцию от входных битов, вычисляющуюся в G, а не его бинарную булеву операцию), удалить G и "провести" эту константу в его последователей. То есть для каждого элемента H, в который идёт провод из G, заменить операцию $h(g,t)$ этого элемента (где g соответствует элементу G, а t — второму входу) на операцию $h'(g,t) = h(c,t)$. (В этом случае H становится проходным и будет впоследствии удалён правилом 2 или правилом 3.)

Правило 3: если G является проходным, то есть вычисляет бинарную булеву операцию, которая зависит только от одного из входов, удалить G, перекинув провода от этого входа к его последователям. Это также может потребовать изменения операций в последователях (если G вычисляет отрицание своего входа), но при этом типы элементов сохранятся ($\oplus$-элемент не может превратиться в $\wedge$-элемент и наоборот).

Если из G ведёт провод в самого себя и G зависит от другого входа, мы также удаляем эту петлю.

Если из G не выходит проводов, он обязан быть выходным элементом (в противном случае он удалился бы правилом 1). В таком случае мы удаляем G и помечаем соответствующий вход G (или его отрицание) как выход.

Правило 4: если G — 1-элемент с последователем $Q \neq G$, в который также ведёт провод из одного из входов G, перебросить провод из I_1 в Q (это также может потребовать изменить операцию в Q) и удалить G. Такой элемент G называется *бесполезным*.

Правило 5: если входы G равны (I_1 и I_2 совпадают), заменить операцию $g(x, y)$, вычисляющуюся в G, на $g'(x, y) = g(x, x)$. После чего к G можно применить правило 2 или 3.

Утверждение 30. *Каждое из правил 1–5 удаляет внутренний элемент и создаёт при этом не более четырёх новых специальных элементов. Если из входного элемента не было пути в выходной, то его не будет и после применения правила.*[1] *Никакое правило не изменяет функции от n входных битов, вычисляющиеся в элементах. Честная полусхема остаётся честной полусхемой.*

Доказательство. Честность. Схема остаётся честной, поскольку ни одно из правил не меняет множество решений соответствующей системы.

[1] Данное простое наблюдение понадобится позже, когда мы будем оценивать количество таких элементов

Новые специальные элементы. Для всех правил стать специальными могут только I_1, I_2 (если они типа $\wedge$) и элементы, в которые они входят после применения правила (если I_1 или I_2 являются входами). Каждый из I_1 и I_2 может создать не более двух специальных элементов. Значит, всего появится не более четырёх новых специальных элементов. □

1.3.2.3 Аффинные подстановки

В данном разделе мы рассмотрим аффинные подстановки, создающие новые циклы. Мы возьмём элемент, вычисляющий аффинную функцию $x_1 \oplus \bigoplus_{i \in I} x_i \oplus c$ (где $c \in \mathbb{F}_2$ — константа), и перекинем провода так, что данный элемент заменится на тривиальный элемент (вычисляющий константу $b \in \mathbb{F}_2$), а x_1 заменится на внутренний элемент. Полученная схема от переменных $x_2, \ldots, x_n$ будет соответствовать исходной схеме после подстановки $x_1 \leftarrow \bigoplus_{i \in I} x_i \oplus c \oplus b$. Перед формальным описанием процесса перекидывания проводов мы докажем структурную лемму (которая очевидна для ациклических схем), гарантирующую успех этого процесса.

Будем говорить, что элемент G аффинной схемы зависит от переменной x, если G вычисляет аффинную функцию x, одним из членов которой является. В схеме без циклов это означает, что ровно один из входов зависит от x, поэтому, идя таким образом от G назад, можно найти путь из x в G. В следующей лемме мы показываем, что такой путь можно найти и в циклической схеме. Возможно, однако, что некоторые вершины на этом пути не будут зависеть от x. Таким образом, зависимости в циклическим схемах ведут себя контринтуитивно в некоторых ситуациях. Например, в схеме выше в элемент G_4 идёт провод x_2, но сам элемент от этой переменной не зависит.

Лемма 31. *Пусть $\mathcal{C}$ — честная циклическая аффинная схема и пусть элемент G зависит от x. Тогда есть путь из x в G.*

Доказательство. Подставим константу ноль во все входные элементы схемы

$\mathcal{C}$ кроме x. Поскольку G зависит от x, после подстановки он будет вычислять x или её отрицание.

Пусть $\mathcal{R}$ — множество элементов, достижимых из x, а $\mathcal{U}$ — множество всех остальных элементов. Перенумеруем элементы так, чтобы сначала шли элементы из $\mathcal{U}$, а потом — из $\mathcal{R}$. Тогда схема $\mathcal{C}$ соответствует системе

$$\begin{bmatrix} U & 0 \\ R_1 & R_2 \end{bmatrix} \times \mathcal{G} = \begin{bmatrix} L_U \\ L_R \end{bmatrix},$$

где $\mathcal{G} = (g_1, \ldots, g_{\mathtt{gates}(\mathcal{C})})^T$ — вектор неизвестных (значений элементов), U — главная подматрица, соответствующая $\mathcal{U}$ (квадратная подматрица, строки и столбцы которой соответствуют элементам $\mathcal{U}$). Отметим следующее:

- правая верхняя часть матрицы равна 0, поскольку нет проводов из $\mathcal{R}$ в $\mathcal{U}$, а значит, неизвестные, соответствующие элементам из $\mathcal{R}$, не участвуют в уравнениях, соответствующие элементам из $\mathcal{U}$,
- вектор констант L_U не содержит x, поскольку элементы из $\mathcal{U}$ недостижимы из x,
- L_R — вектор аффинных функций от x, поскольку все остальные переменные подставлены.

Если бы подматрица U была вырождена, тогда была бы вырождена и вся матрица системы, что противоречило бы честности $\mathcal{C}$. Значит, U невырождена, то есть значения $\mathcal{G}' = (g_1, \ldots, g_{|\mathcal{U}|})^T$ однозначно определяются по $U \times \mathcal{G}' = L_U$ и являются константами (не зависят от x). Это, в свою очередь, означает, что G не может лежать в $\mathcal{U}$. □

Опишем теперь формально процесс перестроения схемы при аффинной подстановке.

Лемма 32. *Пусть $\mathcal{C}$ — честная полусхема с входными элементами $x_1, \ldots, x_n$ и внутренними элементами $G_1, \ldots, G_m$. Пусть G — элемент типа $\oplus$, не достижимый ни из какого элемента типа $\wedge$. Пусть G вычисляет функцию $x_1 \oplus \bigoplus_{i \in I} x_i \oplus c$, где $I \subseteq \{2, \ldots, n\}$. Пусть $b \in \mathbb{F}_2$ — константа. Тогда можно преобразовать схему $\mathcal{C}$ в схему $\mathcal{C}'$, удовлетворяющую следующим свойствам:*

- *схема $\mathcal{C}'$ имеет те же элементы, что и $\mathcal{C}$, а также дополнительный элемент Z; при этом некоторые провода изменены; в частности, из x_1 уже нет исходящих проводов;*
- *операция, вычисляющаяся в G, заменяется на константную операцию b;*
- $\operatorname{outdeg}_{\mathcal{C}'}(Z) = 2$, $\operatorname{outdeg}_{\mathcal{C}'}(G) = \operatorname{outdeg}_{\mathcal{C}}(G) + 1$, $\operatorname{outdeg}_{\mathcal{C}'}(x_1) = 0$, $\operatorname{outdeg}_{\mathcal{C}'}(Z) = \operatorname{outdeg}_{\mathcal{C}}(x_1) - 1$.
- *все остальные входящие и исходящие степени в $\mathcal{C}$ и $\mathcal{C}'$ совпадают;*
- *$\mathcal{C}'$ является честной;*
- *все общие элементы схем $\mathcal{C}'$ и $\mathcal{C}$ вычисляют одни и те же функции в аффинном подпространстве, задаваемом уравнением $x_1 \oplus \bigoplus_{i \in I} x_i \oplus c \oplus b = 0$; другими словами, если функция $f(x_1, \ldots, x_n)$ вычисляется во внутреннем элементе $\mathcal{C}$, а $f'(x_2, \ldots, x_n)$ вычисляется в аналоге этого элемента в $\mathcal{C}'$, то*

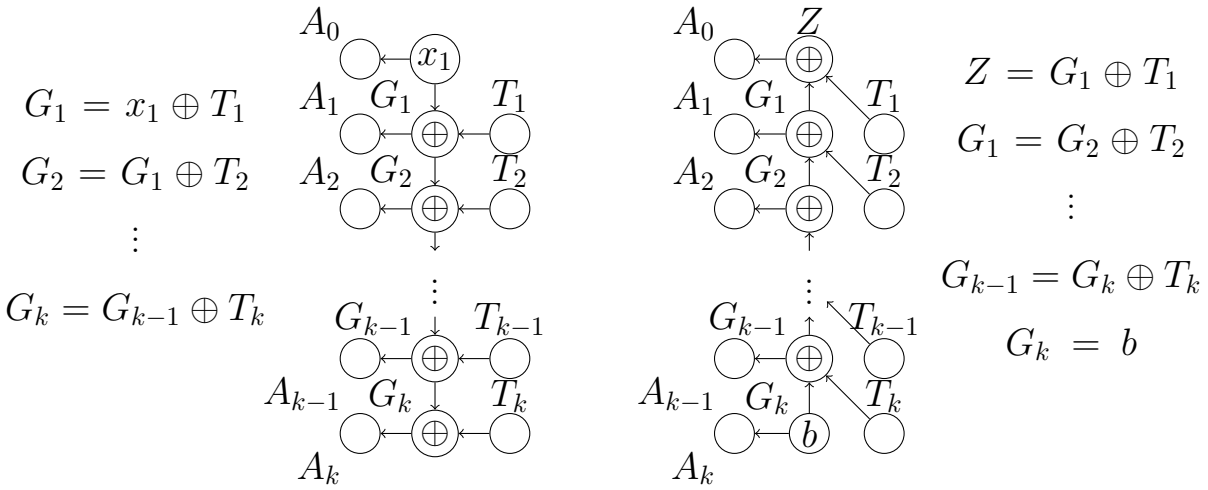
$$f(\bigoplus_{i \in I} x_i \oplus c \oplus b, x_2, \ldots, x_n) = f'(x_2, \ldots, x_n).$$

  Элемент Z вычисляет функцию $\bigoplus_{i \in I} x_i \oplus c \oplus b$ (которая на данном аффинном подпространстве равна просто x_1).

Доказательство. Рассмотрим путь из x_1 в G, имеющийся по лемме 31. Обозначим внутренние вершины этого пути через $G_1, \ldots, G_k = G$. Через

$T_1, \ldots, T_k$ обозначим другие входы этих элементов. Отметим, что элементы $G_1, \ldots, G_k$ попарно различны, в то время как некоторые $T_1, \ldots, T_k$ могут совпадать друг с другом и с элементами из $G_1, \ldots, G_k$ (может даже оказаться, что $T_i = G_i$).

Преобразование показано на рисунке ниже. Элементы $A_0, \ldots, A_k$ показаны исключительно для удобства: каждый из $x_1, Z, G_1, \ldots, G_k$ может входить в любое число элементов, а не только в A_i.



Чтобы показать честность $\mathcal{C}'$, предположим противное, то есть что сумма какого-то подмножества строк матрицы равна нулю. Строка, соответствующая $G_k = b$, обязана быть среди этих строк (иначе были бы один строки из $\mathcal{C}$). Это, однако, означало бы, что если мы сложим соответствующие строки $\mathcal{C}$, мы получим $G_k = \text{const} \oplus \bigoplus_{j \in J} x_j$, где $J \not\ni 1$ (заметим, что x_1 заменена на Z в новой системе, а Z сокращён по предположению). Это противоречит условию леммы, что G_k вычисляет $x_1 \oplus \bigoplus_{i \in I} x_i \oplus c$. Следовательно, матрица для $\mathcal{C}'$ имеет полный ранг.

Программы, показанные рядом со схемами, доказывают, что при $x_1 = \bigoplus_{i \in I} x_i \oplus c \oplus b$ элементы $G_1, \ldots, G_k$ вычисляют одни и те же функции в $\mathcal{C}'$ и $\mathcal{C}$; значение Z также корректно. □

Следствие 33. *Данное преобразование не вводит новых специальных элементов.*

Доказательство. Элементы, в которые вели провода из $G_1, \ldots, G_{k-1}, G_k, Z$, после преобразования не зависят от двух переменных. Сами элементы $G_1, \ldots, G_{k-1}, G_k, Z$ не могут стать специальными, потому что имеют тип $\oplus$. Для всех остальных элементов не меняется ни их тип, ни тип их предков. $\square$

После применения данного преобразование мы также применяем правило 2 к G. Единственными элементами, которые могут стать специальными при применении этого правила, являются входы удаляемого элемента. Нетрудно видеть, однако, что специальными они не становятся. Таким образом, при применении аффинной подстановки и последующем применении правила 2 не появляется новых специальных элементов.

1.3.3 Однопроходные квадратичные источники глубины два

Нам понадобится следующее обобщение аффинных источников.

Определение 34 (однопроходный квадратичный источник глубины два)**.** *Пусть переменные $\{x_1, \ldots, x_n\}$ разбиты на три непересекающихся множества $F, L, Q \subseteq \{1, \ldots, n\}$ (от слов свободные (*free*), линейные (*linear*) и квадратичные (*quadratic*)). Рассмотрим систему уравнений, которая содержит*

- *для каждой переменной x_j, где $j \in Q$, квадратичное уравнение в форме*

$$x_j = (x_i \oplus c_i)(x_k \oplus c_k) \oplus c_j \,,$$

где $i, k \in F$ и c_i, c_k, c_j — константы; все переменные из правых частей квадратичных уравнений должны быть попарно различные;

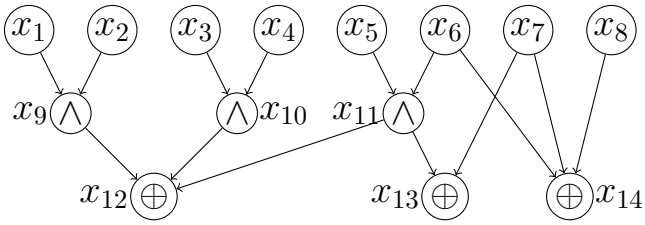
- *для каждой переменной x_j, где $j \in L$, аффинное уравнение в форме*

$$x_j = \bigoplus_{i \in F_j \subseteq F} x_i \oplus \bigoplus_{i \in Q_j \subseteq Q} x_i \oplus c_j$$

для константы c_j.

Подмножество R множества $\{(x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n\}$, удовлетворяющее таким уравнениям, называется однопроходным квадратичным источником глубины два *(или просто* OK-источников*) размерности $d = |F|$.*

Ниже приведён пример такой системы уравнений, который заодно объясняет происхождение названия источника: каждая переменная может входить в элемент $\wedge$-типа не более раза (и сколько угодно раз в элементы $\oplus$-типа).



Определение 35 (защищённая переменная)**.** *Переменные из правых частей квадратичных уравнений будем называть* защищёнными*. Оставшиеся свободные переменные будем называть* незащищёнными*.*

OK-источник будет получаться постепенно: мы будем добавлять новые уравнения в процессе элиминации элементов. Мы будем брать незащищённую свободную переменную x_j и добавлять либо квадратичное уравнение

$$x_j = (x_i \oplus c_i)(x_k \oplus c_k) \oplus c_j \,,$$

зависящее от свободных незащищённых переменных x_i, x_k, или аффинное уравнение

$$x_j = \bigoplus_{i \in J} x_i \oplus c_j \,,$$

зависящее от любых переменных. Нетрудно видеть, что такая система уравнений может быть переписана в систему, удовлетворяющую определению 34. В общем случае мы не можем производить подстановку вместо защищённой свободной переменной, но есть два частных случая, когда это возможно:

- можно подставить константу вместо защищённой переменной (и соответствующим образом обновить соответствующее квадратичное уравнение: например, $z = xy$ при $x = 1$ превращается в $z = y$ и $x = 1$);
- вместо защищённой переменной можно подставить другую переменную (или её отрицание) из того же квадратичного уравнения (например, $z = xy$ при $x = y$ превращается в $z = y$ и $x = y$).

В дальнейшем через R мы будем обозначать OK-источник, соответствующую ему систему уравнений и соответствующее отображение $R\colon \mathbb{F}_2^d \to \mathbb{F}_2^n$ (получающее на вход значения d свободных переменных и вычисляющее значения всех исходных n переменных). Для $f \in B_n$ через $f|_R$ будем обозначать функцию из B_d:

$$f|_R(x_1, \ldots, x_d) = f(R(x_1, \ldots, x_d)).$$

Отметим, что аффинные источники являются OK-источниками с $Q = \emptyset$.

Определение 36 (OK-дисперсер)**.** OK-дисперсером *размерности* $d(n)$ *называется семейство функций* $f_n \in B_n$*, такое что при всех достаточно больших* n *и всех OK-источниках* R *размерности хотя бы* $d(n)$ *функция* $f_n|_R$ *не является константой.*

Ниже мы показываем, что аффинные дисперсеры также являются OK-дисперсерами с достаточно хорошими параметрами.

Утверждение 37. *Пусть* $R \subseteq \mathbb{F}_2^n$ *— источник размерности* d*. Тогда* R *содержит аффинное подпространство размерности* $d/2$*.*

Доказательство. Для каждого квадратичного уравнения $x_j = (x_i \oplus c_i)(x_k \oplus c_k) \oplus c_j$ добавим в R уравнение $x_i = 0$. Это заменяет исходное квадратичное уравнение на $x_i = 0$ и $x_j = c_i(x_k \oplus c_k) \oplus c_j$. Количество свободных переменных при этом уменьшается на один. Поскольку свободные переменные не входят в левые части уравнений, новое уравнение совместно с уже существующими.

Поскольку переменные из правых частей квадратичных уравнений не пересекаются, для исходного источника верно $2|Q| \leq |F| = d$. Поэтому количество добавленных уравнений не больше $d/2$. □

Отметим, что для данного доказательства существенно, что защищённые переменные не используются в левых частях уравнений. Утверждение выше неверно для *квадратичных многообразий*: никакая булева функция не может быть неконстантной на множестве корней произвольных $n - o(n)$ квадратичных уравнений. Например, система $n/2$ квадратичных уравнений

$$x_1x_2 = x_3x_4 = \ldots = x_{n-1}x_n = 1$$

задаёт одну точку (и на ней любая функция, конечно же, является константой).

Следствие 38. *Аффиный дисперсер размерности d является OK-дисперсером размерности $2d$. В частности, аффинный дисперсер сублинейной размерности является OK-дисперсером сублинейной размерности.*

1.3.4 Мера сложности

Для схемы $\mathcal{C}$ и системы уравнений R, задающей OK-источник (над тем же множеством переменных), определим следующую меру сложности:

$$\mu(\mathcal{C}, R) = \texttt{gates}(\mathcal{C}) + \alpha_Q \cdot \texttt{quad}(R) + \alpha_S \cdot \texttt{spec}(R) + \alpha_I \cdot \texttt{infl}(\mathcal{C}),$$

где $\texttt{gates}(\mathcal{C})$, как обычно, обозначает множество (внутренних) элементов $\mathcal{C}$, $\texttt{quad}(R)$ — это количество квадратичных подстановок в R, $\texttt{spec}(R)$ — число специальных элементов в $\mathcal{C}$, а $\texttt{infl}(\mathcal{C})$ — число *важных* входных элементов в $\mathcal{C}$. Входной элемент является важным, если из него идёт хотя бы один провод или если соответствующая переменная является защищённой (то есть входит в правую часть квадратичного уравнения в R). Константы $\alpha_Q, \alpha_S, \alpha_I > 0$ будут подобраны позже.

Утверждение 30 гарантирует, что при удалении элемента одним из правил нормализации мера μ уменьшается хотя бы на $\beta = 1 - 4\alpha_S$. Константа α_S будет выбрана очень близкой к нулю (точно меньше, чем $1/4$), поэтому $\beta > 0$. Это, в частности, подтверждает, что μ является мерой сложности схем.

Нам понадобится следующая лемма, чтобы оценить изначальное значение меры.

Лемма 39. *Пусть $\mathcal{C}$ — схема, вычисляющая аффинный дисперсер $f \in B_n$ размера d. Тогда*

$$\texttt{spec}(\mathcal{C}) \leq \frac{n}{2} + \frac{5d}{2}.$$

Доказательство. Обозначим через V множество входов, $|V| = n$. В дальнейшем через $\sqcup$ мы будем обозначать объединение непересекающихся множеств. Назовём входы x и y соседями, если они входят в один и тот же специальный элемент. Предположим противное: $\texttt{spec}(\mathcal{C}) \geq \frac{n}{2} + \frac{5d}{2}$. Через v_i обозначим число входов, из которых идут провода в ровно i специальных элементов. Поскольку входные переменные специальных элементов имеют исходящую степень два, то $v_i = 0$ при $i > 2$. Подсчётом проводов получаем, что $2\texttt{spec}(\mathcal{C}) = v_1 + 2v_2$. Поскольку $v_1 + v_2 \leq n$,

$$n + 5d \leq 2\texttt{spec}(\mathcal{C}) = v_1 + 2v_2 \leq n + v_2.$$

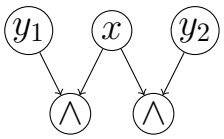
Пусть T — множество входов, из которых идут провода в ровно два специаль-

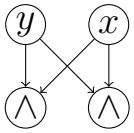
ных элемента, $|T| = v_2 \geq 5d$. Мы построим два непересекающихся множества $X \subset T$ и $Y \subset V$, таких что

- $|X| = d$,
- есть $|Y|$ совместных линейных уравнений, делающих схему $\mathcal{C}$ независимой от переменных из $X \sqcup Y$.

Когда множества X и Y будут построены, теорема будет доказана. Действительно, возьмём $|Y|$ уравнений, делающих $\mathcal{C}$ независимой от $X \sqcup Y$, подставим во все оставшиеся переменные $V \setminus (X \sqcup Y)$ константы. Схема $\mathcal{C}$ тогда станет константой. Значит, получили $|Y| + |V \setminus (X \sqcup Y)| = |V \setminus X| = n - d$ линейных уравнений, что противоречит тому, что f является аффинным дисперсером размерности d.

Теперь построим множества X и Y. Для этого повторим следующее d раз. Выберем произвольную переменную $x \in T$. Она входит в два специальных элемента. Пусть y_1 и y_2 — соседи x (y_1 может совпадать с y_2). Добавим x к X, также добавим y_1, y_2 к Y. Заметим, что можно подставить y_1 и y_2 так, чтобы сделать $\mathcal{C}$ независимой от x.





Действительно, если y_1 отличается от y_2, подставим в них константы, тривиализирующие элементы и, следовательно, забивающие зависимость от x. Если же y_1 совпадает с y_2, тогда либо $x = c$, либо $y_1 = c$, либо $y_1 = x \oplus c$ забивает зависимость от x и y для некоторой константы c (если делаем подстановку $x = c$, тогда нужно поменять местами x и y: добавить y вместо x в X.

Каждый из y_1 и y_2 имеет не более одного соседа, отличного от x. Удалим x, y_1, y_2 и соседей y_1 и y_2 (не более пяти элементов всего) из T. Поскольку на каждом шаге мы удаляем не более пяти элементов из T, мы можем выполнить хотя бы d шагов. Поскольку мы удаляем соседей y_1 и y_2 из T, мы гарантируем,

что на всех последующих шагах при выборе элемента его соседи не лежат в Y, поэтому их можно произвольно подставлять, оставляя систему совместной. □

Сформулируем теперь основные оценки данного подраздела.

Лемма 40. *Пусть $f \in B_n$ — ОК-дисперсер размерности d, а $\mathcal{C}$ — честная полусхема, вычисляющая f. Пусть $\alpha_Q, \alpha_S, \alpha_I \geq 0$ — константы, где $\alpha_S \leq 1/4$. Тогда*

$$\mu(C, \emptyset) \geq \delta(n - d - 2),$$

где

$$\delta := \alpha_I + \min\left\{\frac{\alpha_I}{2}, 4\beta, 3 + \alpha_S, 2\beta + \alpha_Q, 5\beta - \alpha_Q, 2.5\beta + \frac{\alpha_Q}{2}\right\} \tag{1.1}$$

и $\beta = 1 - 4\alpha_S$.

Данная лемма будет доказана в следующем подразделе. Пока что покажем, что вместе со следствием 38 она даёт необходимую нижнюю оценку на схемную сложность аффинного дисперсера сублинейной размерности.

Следствие 41. *Пусть $\delta, \beta, \alpha_Q, \alpha_S, \alpha_I$ — константы из теоремы выше, тогда схемная сложность аффинного дисперсера сублинейной размерности не меньше*

$$\left(\delta - \frac{\alpha_S}{2} - \alpha_I\right) n - o(n)\,.$$

Доказательство. Пусть $\mathcal{C}$ — схема, вычисляющая аффинный дисперсер сублинейной размерности из B_n. Заметим, что $\texttt{quad}(\mathcal{C}) = 0$, $\texttt{infl}(\mathcal{C}) \leq n$,

$\texttt{spec}(\mathcal{C}) < \frac{n}{2} + \frac{5d}{2}$ (по лемме 39). Значит,

$$\begin{aligned}\texttt{gates}(\mathcal{C}) &= \mu(\mathcal{C}) - \alpha_Q \cdot \texttt{quad}(\emptyset) - \alpha_S \cdot \texttt{spec}(\mathcal{C}) - \alpha_I \cdot \texttt{infl}(\mathcal{C}) > \\ &> \delta(n - 2d - 2) - \alpha_S \cdot \left(\frac{n}{2} + \frac{5d}{2}\right) - \alpha_I \cdot n = \\ &= \left(\delta - \frac{\alpha_S}{2} - \alpha_I\right) n - \left(2\delta + \frac{5\alpha_S}{2}\right) d - 2\delta = \\ &= \left(\delta - \frac{\alpha_S}{2} - \alpha_I\right) n - o(n)\,.\end{aligned}$$

□

Максимальное значение выражения $\delta - \frac{\alpha_S}{2} - \alpha_I$, удовлетворяющее условиям следствия 41, находится при помощи следующей линейной программы: максимизировать $\delta - \frac{\alpha_S}{2} - \alpha_I$ при условии, что

$$\begin{aligned}\beta + 4\alpha_S &= 1 \\ \alpha_S, \alpha_Q, \alpha_I, \beta &\geq 0 \\ \delta &\leq \alpha_I + \min\left\{\frac{\alpha_I}{2}, 4\beta, 3 + \alpha_S, 2\beta + \alpha_Q, 5\beta - \alpha_Q, 2.5\beta + \frac{\alpha_Q}{2}\right\}.\end{aligned}$$

Оптимальными значениями являются

$$\begin{aligned}\alpha_S &= \frac{1}{43}, \\ \alpha_Q &= 1 + 22\alpha_S = \frac{65}{43}, \\ \alpha_I &= 6 + 2\alpha_S = 6 + \frac{2}{43}, \\ \beta &= 1 - 4\alpha_S = \frac{39}{43}, \\ \delta &= 9 + 3\alpha_S = 9 + \frac{3}{43}.\end{aligned}$$

Это завершает доказательство теоремы 23.

1.3.5 Доказательство нижней оценки

Чтобы доказать лемму 40, мы покажем, что всегда найдётся подстановка, уменьшающая меру μ хотя бы на δ.

Теорема 42. *Пусть $f \in B_n$ — ОК-дисперсер размерности d, R — ОК-источник размерности $s \geq d+2$, а $\mathcal{C}$ — честная полусхема с минимальным значением $\mu(\mathcal{C}, R)$, вычисляющая $f|_R$. Тогда найдётся ОК-источник R' размерности $s' < s$ и честная полусхема $\mathcal{C}'$, вычисляющая функцию $f|_{R'}$, такие что*

$$\mu(\mathcal{C}', R') \leq \mu(\mathcal{C}, R) - \delta(s - s').$$

Перед тем, как доказать эту теорему, мы покажем, как именно она влечёт основную оценку.

Доказательство леммы 40. Покажем, что для оптимальной (относительно μ) схемы $\mathcal{C}$, вычисляющей $f|_R$, выполнено $\mu(\mathcal{C}, R) \geq \delta(s - d - 2)$. Покажем это индукцией по s, размерности R. База индукции $s \leq d + 2$ выполнена, поскольку μ неотрицательна. Для перехода допустим, что утверждение выполняется для всех размерностей меньше s для $s > d + 2$, и пусть R — ОК-источник размерности s. Пусть $\mathcal{C}$ — честная полусхема, вычисляющая $f|_R$. Пусть R' — ОК-источник размерсности s', гарантированный теоремой 42, и пусть $\mathcal{C}'$ — честная полусхема, вычисляющая $f|_{R'}$. Тогда

$$\mu(\mathcal{C}, R) \geq \mu(\mathcal{C}', R') + \delta(s - s') \geq \delta(s - d - 2),$$

где второе неравенство выполнено по предположению индукции. □

1.3.5.1 Основные идеи доказательства

Доказательство теоремы 42 основано на аккуратном анализе множества случаев. Перед тем, как проводить данный анализ, мы приводим его основные идеи. Зафиксируем оптимальные значения констант $\alpha_S, \alpha_Q, \alpha_I, \beta, \delta$:

$\alpha_S = \frac{1}{43}, \alpha_Q = \frac{65}{43}, \alpha_I = 6\frac{2}{43}, \beta = \frac{39}{43}, \delta = 9\frac{3}{43}$. Достаточно показать, что всегда найдётся подстановка, уменьшающая меру хотя бы на $\delta = 9\frac{3}{43}$. Сразу нормализуем рассматриваемую схему. По утверждению 30 при удалении элемента добавляется не более четырёх новых специальных элементов, что означает, что мера уменьшается хотя бы на $1 - 4\alpha_S = \frac{39}{43}$. В частности, нормализация никогда не увеличивает меру схемы.

Мы будем производить константные, аффинные и простые квадратичные подстановки. После этого мы будем удалять подставленную переменную из схемы, так что для любого набора значений подставленным переменным функция определена. Сделать константную подстановку $x \leftarrow c$ для $c \in \mathbb{F}_2$ нетрудно: подставим c в элементы, в которые идёт провод из x, и удалим x из схемы. Аффинную подстановку $x \leftarrow \bigoplus_{i \in I} x_i \oplus c$ делать уже сложнее, поскольку чтобы удалить x при этом, нужно вычислить где-то в схеме $\left(\bigoplus_{i \in I} x_i \oplus c\right)$ и использовать это вместо x. У нас всегда будет такой элемент G, вычисляющий $\bigoplus_{i \in I} x_i \oplus c$ и не достижимый ни из какого $\wedge$-элемента. В такой ситуации лемма 32 позволяет произвести подстановку без увеличения числа элементов.

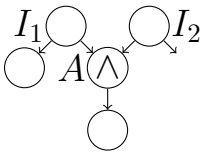
Таким образом, в данном подразделе мы будем делать аффинные подстановки, не заботясь о том, как именно они производятся (подразумевая при этом, что при подстановке запускается процедура преобразования). Мы также будем производить простые квадратичные подстановки $z \leftarrow (x \oplus c_1)(y \oplus c_2) \oplus c_3$ только в случае, если после подстановки все элементы, в которые входила z, пропадают (из-за чего нам не нужно передавать это квадратичное значение дальше).

Чтобы оставаться в классе OK-источников, мы не будем делать аффинную подстановку в переменную x, если она уже используется в правой части некоторого уравнения $z = (x \oplus c_1)(y \oplus c_2) \oplus c_3$. Мы также не будем производить квадратичных подстановок, пересекающихся по переменным. В данном подразделе мы не будем внимательно проверять выполнение данных двух

условий, но аккуратно проверим это в следующем подразделе, где приводится полное формальное доказательство.

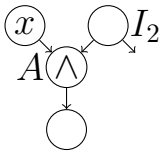
Рассмотрим минимальный $\wedge$ элемент A (то есть $\wedge$-элемент, не достижимый ни из одного другого $\wedge$-элемента) и его предков I_1 и I_2 (I_1 и I_2 могут быть как входными, так и внутренними элементами).

Случай 1. Хотя бы один из I_1 и I_2 (скажем, I_1) является внутренним 2^+-элементом.



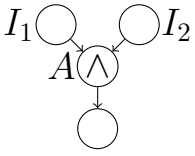
Найдётся константа c, такая что при подстановке $I_1 = c$ элемент A тривиализируется. При этом удаляется он сам и его потомки. Удаляется также I_1. Следовательно мера уменьшается хотя бы на $\alpha_I + 4\beta = 9\frac{29}{43} > \delta$.

Случай 2. Хотя бы один из I_1 и I_2 (скажем, I_1) является 1-переменной.



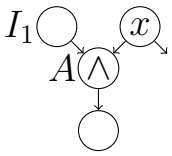
Подставим соответствующую константу в I_2. Это удалит I_2, A, потомков A и I_1. Такая подстановка удаляет хотя бы два внутренних элемента и хотя бы два входных, что даёт уменьшение меры хотя бы $2\alpha_I + 2\beta = 13\frac{39}{43} > \delta$.

Случай 3. I_1 и I_2 являются внутренними 1-элементами.



Подставив нужную константу в I_1, удалим I_1, A, потомков A и I_2 (поскольку I_2 больше не входит ни в какие элементы). Уменьшим при этом меру на $\alpha_I + 4\beta > \delta$.

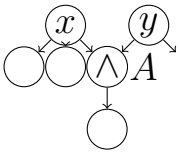
Случай 4. I_1 — внутренний 1-элемент, I_2 — входной 2^+-элемент.



Подставим нужную константу в I_2. Это удалит I_2, хотя бы двух его потомков (включая A), потомка A и I_1 (который больше никуда не входит). Мера уменьшится хотя бы на $\alpha_I + 4\beta > \delta$.

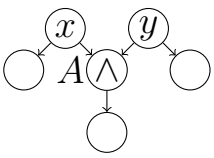
Случай 5. I_1 и I_2 — 2^+-переменные.

Случай 5.1. I_1 или I_2 (скажем, I_1) имеет исходящую степень хотя бы 3.



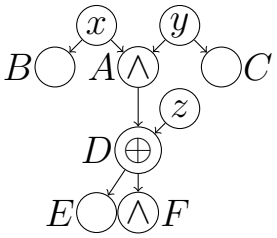
Подставив нужную константу в I_1, удалим хотя бы трёх его потомков и потомка A, то есть хотя бы четыре элемента.

Случай 5.2. I_1 и I_2 являются 2-переменными. Если A есть 2^+-элемент, удалим четыре элемента, подставив правильную константу в I_1, поэтому в дальнейшем предположим, что A есть 1-элемент. Тогда A является специальным. Мы сделаем подстановку, удаляющую I_1 (или I_2), его потомка, A и потомка A.



Случай 5.2.1. Если эта подстановка не создаёт новых специальных элементов, то мы удаляем один вход и три элемента, один из которых является специальным. Мера уменьшается на $\alpha_I + 3 + \alpha_S = 9\frac{3}{43} = \delta$.

Случай 5.2.2. Если же такая подстановка вводит новые специальные элементы, тогда мы аккуратно разбираемся, в результате удаления какого элемента появился новый специальный элемент. В зависимости от этого мы находим другую более выгодную подстановку. Подробный анализ данного случая проведён в следующем подразделе. Здесь же мы разберём один такой подслучай. Пусть новый специальный элемент появляется, когда мы удаляем потомка A: переменная z будет входить в новый специальный элемент после подстановки $x \leftarrow 0$ или $y \leftarrow 0$.



Сделаем тогда подстановку $z \leftarrow (x \oplus c_1)(y \oplus c_2) \oplus c_3$. Это удалит элементы A, D, E, F и потомка F. Всего удалится одна переменная, пять элементов, но добавится квадратичное уравнение. Мера уменьшится хотя бы на $\alpha_I + 5\beta - \alpha_Q = 9\frac{3}{43} = \delta$.

В данном неформальном доказательстве также опущены случаи, когда некоторые из трёх/четырёх/пяти удаляемых элементов совпадают. Такие ситуации разобраны в следующем разделе.

1.3.6 Полное доказательство

Доказательство теоремы 42. Поскольку нормализация не увеличивает значение меры и не изменяет R, можно считать, что схема $\mathcal{C}$ нормализована.

Мы будем сужать R, уменьшая число свободных переменных на один или два, делая соответствующие подстановки в $\mathcal{C}$ и нормализуя $\mathcal{C}$. Мы делаем это следующим образом:

- Добавляем одно или два уравнения в R.
- Поскольку теперь нам нужно вычислять функцию на меньшем множестве, мы упрощаем $\mathcal{C}$ (в частности, отсоединяем подставленные переменные от схемы). Для этого мы
  - изменяем операции в элементах, в которые шли провода из подставленных переменных, или перестраиваем линейную часть схемы в соответствии с леммой 32,
  - применяем правила упрощения, чтобы удалить некоторые элементы и отсоединить подставленные переменные.
- Оцениваем, на сколько уменьшилась мера μ.

Поскольку $s \geq d + 2$, даже если мы добавим два новых уравнения к R, дисперсер не может обратиться в константу. Это, в частности, означает, что если некоторый элемент стал константой при подстановке, то он не является выходным элементом, а значит, из него выходит хотя бы один провод. Рассмотрев несколько случаев, мы покажем, что всегда найдётся одна или две подстановки одного из следующих типов (через $\Delta\mu$ мы обозначаем изменение меры после нормализации).

1. Сделать две аффинных подстановки подряд и уменьшить количество важных переменных хотя бы на три. В расчёте на одну подстановку это даёт $\Delta\mu \geq 1.5\alpha_I$.
2. Сделать одну аффинную подстановку и уменьшить число важных входов хотя бы на два: $\Delta\mu \geq 2\alpha_I$ (этот случай мажорируется предыдущим случаем).

3. Сделать аффинную подстановку и удалить хотя бы четыре внутренних элемента: $\Delta\mu \geq 4\beta + \alpha_I$.

4. Сделать константную подстановку, удалив три элемента, один из которых специальный, и не добавив при этом ни одного нового специального элемента: $\Delta\mu \geq \alpha_I + 3 + \alpha_S$.

5. Сделать квадратичную подстановку, удалив хотя бы пять внутренних элементов: $\Delta\mu \geq 5\beta - \alpha_Q + \alpha_I$.

6. Сделать две аффинные подстановки, удалив хотя бы пять внутренних элементов и заменив сделанную ранее квадратичную подстановку на аффинную: $5\beta + \alpha_Q + 2\alpha_I$. В расчёте на одну подстановку это даёт $\Delta\mu \geq 2.5\beta + \frac{\alpha_Q}{2} + \alpha_I$.

7. Сделать одну аффинную подстановку, удалив при этом два внутренних элемента и заменив сделанную ранее квадратичную подстановку на аффинную: $\Delta\mu \geq 2\beta + \alpha_Q + \alpha_I$.

Все производящиеся подстановки будут таковы, что при добавлении соответствующего уравнения к текущему ОК-источнику R даёт новый ОК-источник.

Ниже мы рассматриваем несколько случаев. Переходя к очередному случаю, мы предполагаем, что условия всех предыдущих случаев не выполнены. Мы также предполагаем, что правила нормализации применяются в том порядке, в котором они перечислены.

При оценке уменьшения меры мы будем аккуратно следить за тем, не появилось ли новых специальных элементов. Напомним, что утверждение 30 гарантирует, что мера уменьшается хотя бы на β при каждом удалении элемента. Если также дополнительно удаляются какие-то другие элементы, это не приводит к увеличению меры, поскольку $\beta \geq 0$.

Случаи.

Случай 1. Есть защищённая переменная q, которая входит либо в один $\wedge$-элемент, либо хотя бы в два внутренних элемента. Тогда есть подстановка типа 7 константы вместо переменной q.

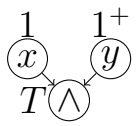
Случай 2. В схеме есть защищённая 0-переменная q, встречающаяся в правой части квадратичной подстановки вместе с другой переменной q'. Подставим константу в q'. После этого и q, и q' перестают быть важными, поэтому получаем подстановку типа 2.

Отметим, что после этого все защищённые переменные являются 1-переменными, входящими в $\oplus$-элементы.

Случай 3. В схеме есть переменная x, входящая в $\wedge$-элемент T, и $\mathrm{outdeg}(x) + \mathrm{outdeg}(T) \geq 4$. При подстановке в x константы, тривиализирующей T, удалятся четыре элемента: T по правилу 2 и его последователи x и T по правилу 3. Если какие-то два последователя совпадают, то этот последователь сам тривиализируется (не становится константой) и удаляется по правилу 2 (а не правилу 3), что влечёт удаление его последователей. Получаем подстановку типа 3.

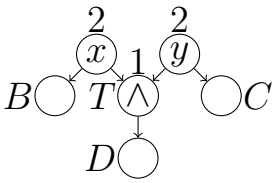
После этого случая все переменные, входящие в $\wedge$-элементы имеют исходящую степень один или два.

Случай 4. Есть $\wedge$-элемент T, зависящий от двух переменных x и y, одна из которых (скажем, x) имеет исходящую степень 1. Используем обозначения со следующего рисунка. На этом и последующих рисунках над некоторыми из элементов мы будем показывать их исходящую степень, если это важно для анализа.



Подставим в y константу, тривиализирующую T. Это удалит зависимость от x и y (которые являются важными и незащищёнными), что даёт подстановку типа 2.

Случай 5. Есть $\wedge$-элемент T, зависящий от двух переменных x и y. Тогда мы знаем (по случаям 3 и 4), что $\mathrm{outdeg}(T) = 1$ и $\mathrm{outdeg}(x) = \mathrm{outdeg}(y) = 2$, то есть T является специальным. Введём такие обозначения:



Поскольку схема нормализована, $B \neq D$ и $C \neq D$ (по правилу 4). При подстановке в x константы, тривиализирующей T, удаляются три элемента. Если удаляется ещё хотя бы один элемент, то это уже даёт необходимое уменьшение меры (подстановка типа 3). В противном случае удаляются всего три элемента, но также пропадает и специальный элемент T. Если это не приводит к появлению новых специальных элементов, получаем подстановку типа 4. Если одна из этих ситуаций оказывается при подстановке переменной y, мы тоже получаем необходимое уменьшение меры.

Соответственно, в оставшихся подслучаях случая 5 мы будем аккуратно перебирать все возможные ситуации, когда удаляются ровно три элемента и при этом появляются новые специальные элементы.

Откуда может взяться новый специальный элемент? Это означает, что при подстановке и последующей нормализации что-то изменилось для некоторого $\wedge$-элемента E. Что бы ни случилось, это произошло потому, что B и D стали проходными (если бы один из них стал тривиальным, удалилось бы хотя бы четыре элемента). Возможные варианты таковы:

- Входом E *становится переменная* вместо внутреннего элемента (поскольку какой-то другой элемент стал проходным).

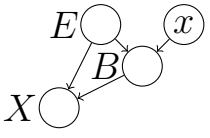
- *Исходящая степень переменной увеличивается* с одного до двух (поскольку элемент исходящей степени хотя бы два стал проходным), и эта переменная начинает входить в E (отметим, что она не могла входить в E до этого, поскольку тогда после подстановки она начала бы входить два раза).

- Исходящая степень переменной *уменьшается* до двух. Эта переменная не могла ранее входить в E, поскольку иначе мы оказались бы в случае 3. Должен быть проходной элемент X, благодаря которому в E стала входить переменная. Значит, уменьшение исходящей степени произошло из-за проходного элемента Y. Чтобы уменьшить исходящую степень переменной, этот элемент должен иметь исходящую степень, равную единице, но тогда он был бы удалён правилом 4 как бесполезный.

- Исходящая степень E *уменьшается* до одного.

  – Это могло произойти, если элементы B и D стали проходными и они входят в один и тот же элемент. Но тогда входами E сразу должны быть 2-переменные, что привело бы нас в случай 3.

  – Это также могло произойти, если E входит в B и некоторый элемент X, а B становится проходным в X. Но в таком случае элемент B является бесполезным (правило 4). (Отметим, что $\mathrm{outdeg}(B) = 1$, поскольку в противном случае исходящая степень E не уменьшилась бы до единицы.)



  – Аналогично если E входит в D и некоторый элемент X, а D становится проходным в X.

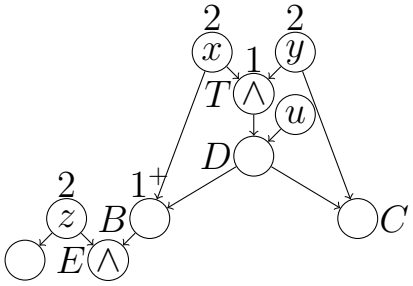
Резюмируя, реализоваться могут только первые две возможности, и в них

обеих некая переменная начинает входить в E из-за того, что B или D (или оба) стал проходным.

В анализе случаев ниже мы разбираем возможные ситуации связей между B, D и C. В первую очередь мы рассмотрим некоторые "вырожденные" случаи связей между этими элементами.

Случай 5.1. Если $B = C$, то можно тривиализировать T и B подстановкой константы в x или y или аффинной подстановкой $y = x \oplus c$ (используя утверждение 28) для правильной выбранной константы c (это нетрудно увидеть, рассмотрев возможные пары операций, вычисляющихся в этих двух элементах). Поскольку x и y незащищены, количество важных переменных уменьшается на два, что даёт подстановку типа 2.

Случай 5.2. Пусть D входит и в B, и в C. В данном случае новый специальный элемент мог появиться только по причине того, что в D входит переменная u, которая пройдёт в $\wedge$-элемент E. Заметим, что $\mathrm{outdeg}(D) \leq 2$, поскольку в противном случае u стала бы 3-переменной и E не стал бы специальным. Следовательно, u не может пройти в E из-за D напрямую, и проходит через B.

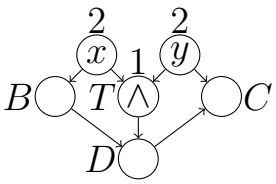


Если $\mathrm{outdeg}(B) \geq 2$, то даже если $\mathrm{outdeg}(u) = 1$, должно быть, что $C = E$ или что B входит C, поскольку в противном случае u стала бы 3-переменной после подстановки x. Но ни то, ни другое невозможно: $C = E$ означало бы, что $B = D$ и что $y = z$, что противоречило бы предположению, что $D \neq B$ (из случая 5); если B входит C, то $B = D$, что невозможно. Следовательно, $\mathrm{outdeg}(B) = 1$. Поэтому можно

подставить константу в z, чтобы сделать B 0-элементом, и константу в y, чтобы тривиализировать T. В следствие этого x перестаёт быть важной, и мы получаем $\Delta\mu \geq 3\alpha_I$ за две подстановки (тип 1).

После этого случая можно предположить, что D не входит в B: если входит, поменяем ролями переменные x и y.

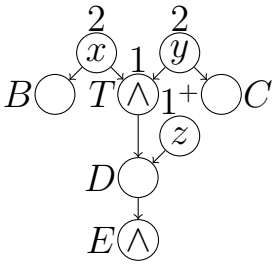
Случай 5.3. Допустим, что B входит в D, а D входит в C. (Или, наоборот, C входит в D, а D входит в B.) Подставив в y константу, тривиализирующую T, удалим T, D, и C. Покажем, что данная подстановка *не вводит новых специальных элементов.* Элемент, в который входил C, теперь зависит от B, поэтому этот элемент не мог стать специальным. Единственным элементом, для которого что-то локально изменилось, является B, но в него после подстановки входит 1-переменная x, поэтому и он не мог стать специальным.



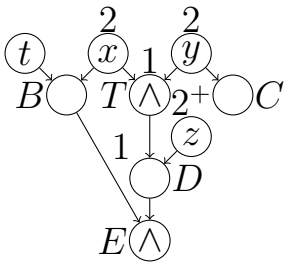
Случай 5.4. Теперь можно считать, что B и D не соединены (ни в каком направлении).

Действительно, если B входит в D, можно поменять ролями x и y, если только C не входит в D (что невозможно, поскольку у D было бы тогда три входа: T, B и C) или мы уже не меняли x и y до этого (то есть D входит в C, случай 5.3).

Случай 5.4.1. Пусть D входит в новый специальный элемент после подстановки x. Специальный элемент E получает новую входную переменную z через D (напрямую, поскольку D и B не соединены).



Случай 5.4.1.1. Если $\mathrm{outdeg}(z) \geq 2$, то $\mathrm{outdeg}(D) = 1$ и E зависит от другой переменной t напрямую или через B. В первом случае подставим t, чтобы тривиализировать E: это удалит E и его потомка, а также сделает D, а потом и T 0-элементами; подстановка типа 3. Во втором случае:



Случай 5.4.1.1.1. Если $out(B) \geq 2$, то B — $\oplus$-элемент (см. случай 3). Сделав подстановку $x = t \oplus c$ для правильной константы c, можно сделать B константой, тривиализирующей E, и удалить ещё двух потомков B и E — подстановка типа 3.

Случай 5.4.1.1.2. Если $\mathrm{outdeg}(B) = 1$, тогда подставим константы в z и y, чтобы тривиализировать T и E, соответственно. Тогда B становится 0-элементом и удаляется, что означает, что x становится 0-переменной. Получаем подстановку типа 1.

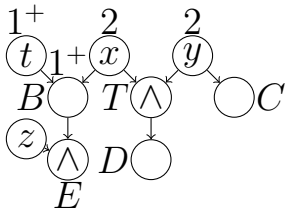
Теперь можно считать, что $\mathrm{outdeg}(z) = 1$*, а следовательно,* $\mathrm{outdeg}(D) \geq 2$*, поскольку исходящая степень* z *должна стать равной двум.*

Случай 5.4.1.2. Если D является $\wedge$-элементом, подставим в z константу, тривиализирующую D, что удалит его потомков; также T становится 0-элементом, что даёт подстановку типа 3.

Случай 5.4.1.3. Если z защищена, подставим в x и z константы, чтобы тривиализировать T, D и E. Это удаляет B и элементы, которые зависят от E — всего хотя бы пять элементов. Поскольку мы также удаляем квадратичную подстановку, это даёт подстановку типа 6.

Случай 5.4.1.4. Теперь можно считать, что z не защищена и что D — $\oplus$-элемент. Сделаем подстановку $z = (x \oplus c_1)(y \oplus c_2) \oplus c_3$ для подходящих констант c_1, c_2, c_3, чтобы сделать D равным константе, тривиализирующей E. Это делает T 0-элементом и удаляет D, E, другой элемент, в который входит D, и элементы, в которые входит E. Как обычно, если какие-то из удаляемых проходных элементов совпадают, то такой элемент тривиализируется и поэтому удаляются и его потомки. Учитывая введение новой квадратичной подстановки, получаем подстановку типа 5.

Случай 5.4.2. Поскольку D не входит в новый специальный элемент, в него входит B, и B зависит от переменной t напрямую (поскольку B и D не связаны). Новый специальный элемент E также зависит напрямую от z (поскольку D не входит в него).



Случай 5.4.2.1. Если $\mathrm{outdeg}(B) \geq 2$ (что означает, что B — $\oplus$-элемент, см. случай 3), тогда подставив $x = t \oplus c$ (используя утверждение 28) для подходящей константы c, сделаем B константой, тривиализирующей E, и удалим двух потомков B и E, что даст подстановку типа 3.

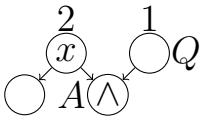
Случай 5.4.2.2. Если $\mathrm{outdeg}(B) = 1$, тогда можно подставить в z и y

константы, тривиализирующие T и E, соответственно. Тогда B становится 0-элементом и удаляется, что означает, что x становится 0-переменной. Получается подстановка типа 1.

Начиная со следующего случая мы будем рассматривать топологически первый $\wedge$-элемент A, то есть такой $\wedge$-элемент, который не достижим ни из какого другого $\wedge$-элемента. Он существует, потому что циклы возможны только в линейной части схемы.

Отметим, что схема $\mathcal{C}$ обязана содержать хотя бы один $\wedge$-элемент (иначе она считала бы аффинную функцию и её можно было бы превратить в константу одной линейной подстановкой). Минимальность A означает, что оба входа A вычисляются честными аффинными полусхемами (заметим, что подсхема честной схемы является честной, поскольку соответствует подматрице матрицы полного ранга); в частности, они могут быть входами.

Случай 6. Один из входов A — 2-переменная x, а второй — внутренний 1-элемент Q.



Заметим, что x незащищена по случаю 1 и x не может входить в Q по правилу 4. Подставив в x константу, тривиализирующую A, удалим двух потомков x, всех потомков A и сделаем Q 0-элементом, который удалится правилом 1. Это даст подстановку типа 3. (Как обычно, если единственный потомок A совпадает с другим потомком x, тогда этот элемент становится константой и удаляются ещё и его потомки. Значит, в любом случае удаляются хотя бы четыре элемента.)

Случай 7. Один из входов A — внутренний элемент Q. Обозначим другой вход через P. Если P — тоже внутренний элемент и его исходящая степень больше Q, поменяем P и Q ролями.

Мы подставим в Q константу, тривиализирующую A. Элемент Q вычисляется аффинной честной схемой, значит, вычисляет функцию $c \oplus \bigoplus_{i \in I} x_i$. Заметим, что $I \neq \emptyset$ по правилу 2. Для этого мы будем использовать перестроение из леммы 32. Чтобы выполнить его, нам понадобится хотя бы одна незащищённая переменная x_i, где $i \in I$.

Случай 7.1. Такая переменная x_1 есть.

Добавим подстановку $x_1 = b \oplus c \oplus \bigoplus_{i \in I \setminus \{1\}} x_i$ в OK-источник R для подходящей константы b (так что Q на R вычисляет константу, тривиализирующую A). Мы хотели бы просто заменить операцию, вычисляющуюся в Q, на эту константу. Однако нам нужно удалить только что подставленную переменную x_1 из схемы. Для этого мы используем перестройку из леммы 32. Заметим, что она изменяет входящие и исходящие степени только у x_1 (заменяя его на Z) и Q. Новые специальные элементы при этом не появляются, и последующее применение правила 2 к Q удаляет Q тоже без введения новых специальных элементов.

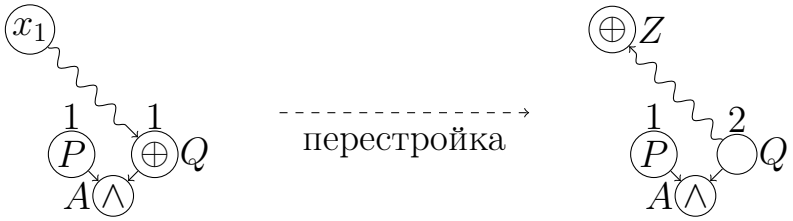
Более того, нормализация удаляет всех потомков Q, всех потомков A и, в случае $\mathrm{outdeg}(P) = 1$, правило 1 удаляет P, если он является внутренним элементом, или P становится 0-переменной, если он был переменной. Остаётся оценить, насколько уменьшилась мера.

Ниже мы проводим такую оценку, рассмотрев несколько случаев в зависимости от типа P.

Случай 7.1.1. Q является 2^+-элементом. Напомним, как происходит перестройка.

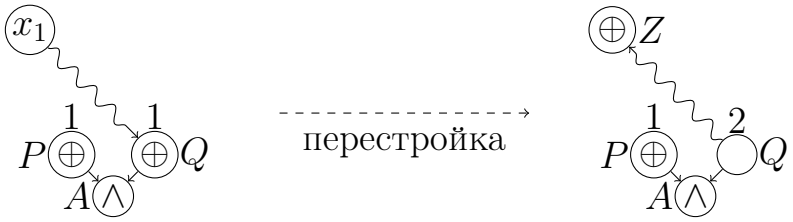
После перестройки удалятся хотя бы три потомка Q и хотя бы один потомок A, что даёт подстановку типа 3.

Случай 7.1.2. Q является внутренним 1-элементом, P — входным. Тогда P — 1-переменная и не защищена (см. случаи 6, 1).



Отметим, что $P \neq x_1$, поскольку единственное исходящее из P ребро идёт в $\wedge$-элемент. Это означает, что P не затрагивается перестройкой. После тривиализации A схема становится независимой от x_1 и P, что даёт подстановку типа 2.

Случай 7.1.3. Q — внутренний 1-элемент, а P — внутренний. Тогда исходящая степень P равна одному (если бы она была больше, мы поменяли бы P и Q ролями).



Опять же, P не затрагивается перестройкой, поскольку у него всего один потомок и он типа $\wedge$, в то время как перестройка затрагивает линейную часть схемы. После подстановки удаляются два потомка Q, хотя бы один потомок A, а P становится 0-элементом. Это даёт подстановку типа 3. Заметим, что P не может быть потомков Q по правилу 4.

Случай 7.2. Все переменные из аффинной функции, вычисляемой в Q, защищены.

Случай 7.2.1. Оба входа Q — скажем, x_j и x_k — являются переменными, входящими в правую часть одного и того же квадратичного урванения $w = (x_j \oplus c)(x_k \oplus c') \oplus c''$. Сделаем подстановку $x_j = x_k \oplus c'''$ (используя утверждение 28), чтобы тривиализировать A. Это удалит квадратичную подстановку (и не затронет другие квадратичные подстановки, потому что x_j и x_k в них не входят), Q, A, их потомка (и даже больше, но это уже неважно), что даёт $\Delta\mu \geq 3\beta + \alpha_Q + \alpha_I$, то есть подстановку типа 7.

Случай 7.2.2. Q является 2^+-элементом. Возьмём $j \in I$. Пусть x_j входит в квадратичную подстановку $x_p = (x_j \oplus a)(x_k \oplus b) \oplus c$. Напомним, что к этому моменту все защищённые переменные являются 1-переменными, входящими в $\oplus$-элементы (см. случаи 1 и 2). Подставим в x_k константу d и нормализуем схему. Это удалит последователя x_k, квадаратичную подстановку и сделает x_j незащищённой. Если при этом удалится хотя бы два элемента, то $\Delta\mu \geq 2\beta + \alpha_Q + \alpha_I$, подстановка типа 7. В дальнейшем мы предполагаем, что после подстановки $x_k \leftarrow d$ удаляется только потомок x_k.

Если элемент Q не зависит от x_k, тогда его исходящая степень будет хотя бы 2 после подстановки $x_k \leftarrow d$ и нормализации. Если в Q идёт провод из x_k, тогда его второй вход должен быть внутренним $\oplus$-элементом Q' (если бы это был входной элемент, это была бы переменная x_j, но тогда мы бы оказались в случае 7.2.1). Тогда после подстановки $x_k \leftarrow d$ и нормализации Q элемент Q' входит в A и имеет исходящую степень хотя бы 2. Обозначим Q' через Q в этом случае.

Таким образом, в любом случае в схеме после подстановки $x_k \leftarrow d$ в элемент A идёт провод из 2^+-элемента Q, который вычисляет аффинную функцию от переменных, одна из которых — незащищённая переменная x_j. Сделаем Q константой, тривиализирующей A, с помощью аффинной подстановки в x_j. Это удаляет четыре элемента. Вместе с подстановкой $x_k \leftarrow d$ это даёт $\Delta\mu \geq 5\beta + \alpha_Q + 2\alpha_I$, то есть подстановку типа 6.

В дальнейшем мы считаем, что $\mathrm{outdeg}(Q) = 1$. *Следовательно,* P *— либо переменная, либо 1-элемент типа* $\oplus$.

Случай 7.2.3. P — входной элемент. Возьмём $j \in I$ и пусть x_j входит с x_k в квадратичную подстановку. Подставим $x_k \leftarrow d$ и нормализуем схему. После этого второй вход A по-прежнему вычисляет линейную функцию, зависящую от x_j, которая теперь не является защищённой. Сделаем аффинную подстановку в x_j, тривиализирующую A. Это сделает P 0-переменной, что даёт подстановку типа 1.

Случай 7.2.4. P является внутренним 1-элементом типа $\oplus$. Если P вычисляет аффинную функцию от переменных, одна из которых является незащищённой, тогда мы оказываемся в случае 7.1.3 (с P и Q поменянными ролями). В дальнейшем мы предполагаем, что P и Q вычисляют аффинные функции от защищённых переменных.

Случай 7.2.4.1. Оба входа P или Q (скажем, P) являются переменными x_p и x_q. Пусть x_j — переменная из аффинной функции, вычисляющейся в Q, а x_k — её сосед. Отметим, что $x_j \neq x_p, x_q$, в то время как возможно, что $x_k = x_p$ or $x_k = x_q$. Подставим в x_k константу, чтобы сделать x_j незащищённой. После этого тривиализируем A аффинной подстановкой в x_j. Таким образом, удалим зависимость от трёх переменных двумя подстановками, что даст подстановку типа 1.

Таким образом, в дальнейшем считаем, что и у P, и у Q одним из входов является внутренний $\oplus$-элемент.

Случай 7.2.4.2. Один из P и Q (скажем, Q) вычисляет аффинную функцию, одна из переменных которой (назовём её x_j) имеет соседа x_k, который не входит в P. Подставим в x_k константу и нормализуем потомка x_k. Это удаляем только $\oplus$-элемент, в которой идёт провод из x_k, и делает x_j незащищённой. Отметим, что к этому моменту P по-прежнему является 1-элементом типа $\oplus$. Тривиализируем A, подставив в x_j аффинную функцию. Аналогично случаю 7.1.3, это удаляет четыре элемента и даёт (за две подстановки) $\Delta\mu \geq 5\beta + \alpha_Q + 2\alpha_I$. Подстановка типа 6.

Случай 7.2.4.3. Поскольку P и Q и все элементы, которые в них входят, вычисляют нетривиальные функции (по правилу 2), единственной оставшейся ситуацией является следующая: P вычисляет линейную функцию от одной переменной x_i, Q вычисляет линейную функцию от одной переменной x_j, переменные x_i и x_j входят вместе в квадратичную подстановку, и более того, x_i входит Q, в то время как x_j входит P. Но это просто невозможно. Действительно, поскольку x_i защищена, она входит только в Q. Поскольку Q вычисляет линейную функцию от x_i, лемма 31 гарантирует, что есть путь из x_i в Q. Но это путь должен проходить через P и A, что привело бы к циклу, содержащему $\wedge$-элемент A.

□

1.4 Нижняя оценка $3.11n$ для квадратичных дисперсеров сублинейной размерности

1.4.1 Квадратичные дисперсеры

Две рассмотренные ранее функции, MOD_n^3 и аффинные дисперсеры, можно рассматривать как функции, не обращающиеся в константу ни на каком достаточно большом множестве $S \subseteq \mathbb{F}_2^n$, которое определяется как множество корней k многочленов:

$$S = \{x \in \mathbb{F}_2^n \colon p_1(x) = p_2(x) = \cdots = p_k(x) = 0\}.$$

Для функции MOD_n^3 выполнено $k \leq n - 4$ и каждый p_i равен переменной или её отрицанию, в то время как для аффинного дисперсера размерности d выполнено $k \leq n - d$ и p_i — линейный многочлен. Легко видеть, что в данных двух случаях размер S определяется количеством задающих его многочленов:

$$|S| = 2^{n-k}. \tag{1.2}$$

Естественным обобщением данной идей является рассмотрение многочленов степени не более два. Соответствующее множество S называется квадратичным многообразием.

Определение 43 (квадратичное многообразие)**.** *Множество $S \subseteq \mathbb{F}_2^n$ называется* (n, k)-квадратичным многообразием*, если оно может быть задано как множество корней $t \leq k$ многочленов степени не более два:*

$$S = \{x \in \mathbb{F}_2^n \colon p_1(x) = \cdots = p_t(x) = 0\},$$

где p_i — многочлен степени не более двух для всех $1 \leq i \leq t$.

Определение 44. *Функция* $f \in B_n$ *называется* (n, k, s)-квадратичным дисперсером, *если* f *не обращается в константу ни на каком* (n, k)-*квадратичном многообразии* $S \subseteq \mathbb{F}_2^n$ *размера хотя бы* s.

Следующая лемма показывает, что почти все функции из B_n являются $(n, 2^{o(n)}, 2^{o(n)})$-квадратичными дисперсерами.

Лемма 45. *Пусть* $\omega(1) \leq s \leq 2^{o(n)}, k = o\left(\frac{s}{n^2}\right)$. *Пусть* $D_n \subseteq B_n$ — *множество всех* (n, k, s)-*квадратичных дисперсеров. Тогда*

$$\frac{|D_n|}{|B_n|} \to 1 \text{ при } n \to \infty .$$

Доказательство. Всего есть $q = \frac{n(n+1)}{2} + 1 = \Theta(n^2)$ мономов степени не более два в $\mathbb{F}_2^n$. Есть 2^q многочленов степени не более двух, (n, k)-квадратичных многообразий не больше 2^{qk}. Каждая функция, не являющаяся (n, k, s)-квадратичным дисперсером, может быть задана

1. (n, k)-квадратичным многообразием, на котором она обращается в константу;
2. соответствующей константой;
3. значениями в оставшихся не более чем $2^n - s$ точках.

Значит, число функций, не являющихся (n, k, s)-квадратичными дисперсерами, не больше

$$2^{qk} \cdot 2 \cdot 2^{2^n - s} = 2^{2^n} 2^{qk+1-s} = 2^{2^n} 2^{-\Theta(s)} = o\left(|B_n|\right) .$$

□

Основным результатом данного раздела является следующая оценка:

Теорема 46 ([21]). *Пусть константы* $0 < \alpha \leq 1$ *и* $0 < \beta$ *удовлетворяют следующим неравенствам:*

$$2^{-\frac{2+\alpha}{\beta}} + 2^{-\frac{4+2\alpha}{\beta}} \leq 1, \tag{1.3}$$

$$2^{-\frac{2}{\beta}} + 2^{-\frac{5+2\alpha}{\beta}} \leq 1, \tag{1.4}$$

$$2^{-\frac{3+3\alpha}{\beta}} + 2^{-\frac{2+2\alpha}{\beta}} \leq 1, \tag{1.5}$$

$$2^{-\frac{3}{\beta}} + 2^{-\frac{4+\alpha}{\beta}} \leq 1, \tag{1.6}$$

и пусть $f \in B_n$ *—* (n, k, s)*-квадратичный дисперсер. Тогда*

$$\texttt{gates}(f) \geq \min\{\beta n - \beta \log_2 s - \beta, 2k\} - \alpha n\,.$$

Например, для $(n, 1.83n, 2^{o(n)})$-квадратичного дисперсера теорема 46 с параметрами $\alpha = 0.535$ и $\beta = 3.6513$ даёт нижнюю оценку $3.1163n - o(n) > 3.116n$. Для $(n, 1.78n, 2^{0.03n})$-квадратичного дисперсера получается нижняя оценка $3.006n$.

На данный момент явные конструкции квадратичных дисперсеров с такими параметрами неизвестны. Теорема 46 может рассматриваться как дополнительная мотивация к изучению таких объектов.

Г. Коэн и А. Таль [11] в 2014 г. доказали, что каждый аффинный дисперсер (экстрактор) является также дисперсером (экстрактором) для полиномиальных многообразий с более слабыми параметрами. В частности, их результат в комбинации с аффинным дисперсером, построенным Р. Шалтиелом [51] в 2011 г., даёт явную конструкцию $\left(n, \Theta\left(\frac{n}{2^{\log^{0.9} n}}\right), 2^{o(n)}\right)$-квадратичного дисперсера. Две явные конструкции экстракторов для многообразий в полях больших размеров были представлены З. Двиром [17] в 2012 г. Для схожих, но другого типа полиномиальных источников, явные конструкции дисперсеров (экстракторов) были даны З. Двиром, А. Габизоном и А. Вигдерсоном [18] в 2009 г. для полей большого размера и Э. Бен-Сассоном и А. Габизоном [4]

в 2012 г. для $\mathbb{F}_2$.

1.4.2 Доказательство нижней оценки

Индукция в доказательстве теоремы 46 ведётся по размеру многообразия S. Отметим, что для квадратичных многообразий равенство (1.2) перестаёт быть верным: например, множество корней $n/2$ многочленов $x_1x_2 \oplus 1$, $x_3x_4 \oplus 1$, $\ldots$, $x_{n-1}x_n \oplus 1$ содержит всего одну точку. По этой причине метод элиминации элементов применяется следующим образом. Мы выбираем многочлен p степени не выше двух и рассматриваем два подмногообразия S: $S_0 = \{x \in S \colon p(x) = 0\}$ и $S_1 = \{x \in S \colon p(x) = 1\}$. Для каждого из них мы оцениваем, насколько уменьшается его размер (относительно размера S) и насколько уменьшается при этом размер схемы. Грубо говоря, мы показываем, что в одной из этих веток размеры схемы уменьшается сильно, а размер многообразия — наоборот, несильно.

Нам понадобится следующая техническая лемма.

Лемма 47. *Пусть $0 < \alpha \leq 1$ и $0 < \beta$ суть константы, удовлетворяющие неравенствам* (1.6), (1.3)*:*

$$\begin{aligned} 2^{-\frac{3}{\beta}} + 2^{-\frac{4+\alpha}{\beta}} &\leq 1, \\ 2^{-\frac{2+\alpha}{\beta}} + 2^{-\frac{4+2\alpha}{\beta}} &\leq 1. \end{aligned}$$

Тогда

$$2^{-\frac{4}{\beta}} + 2^{-\frac{4}{\beta}} \leq 1, \tag{1.7}$$

$$2^{-\frac{3+\alpha}{\beta}} + 2^{-\frac{3+2\alpha}{\beta}} \leq 1. \tag{1.8}$$

Доказательство. Поскольку $2 \leq x + \frac{1}{x}$ при положительных x, то

$$2^{-\frac{4}{\beta}} + 2^{-\frac{4}{\beta}} \leq 2^{-\frac{4}{\beta}}(2^{\frac{1}{\beta}} + 2^{-\frac{1}{\beta}}) = 2^{-\frac{3}{\beta}} + 2^{-\frac{5}{\beta}} \leq 2^{-\frac{3}{\beta}} + 2^{-\frac{4+\alpha}{\beta}} \leq 1\,.$$

Для доказательства неравенства (1.8) мы используем неравенство Хайнца [25]:

$$\frac{x^{1-t}y^t + x^t y^{1-t}}{2} \leq \frac{x+y}{2} \text{ для } x, y > 0, 0 \leq t \leq 1.$$

Положим $x = 2^{-\frac{2+\alpha}{\beta}}, y = 2^{-\frac{4+2\alpha}{\beta}}, t = \frac{1}{2+\alpha}$:

$$2^{-\frac{3+\alpha}{\beta}} + 2^{-\frac{3+2\alpha}{\beta}} = x^{1-t}y^t + x^t y^{1-t} \leq x + y = 2^{-\frac{2+\alpha}{\beta}} + 2^{-\frac{4+2\alpha}{\beta}} \leq 1.$$

□

В следующей лемме мы используем меру сложности

$$\mu(\mathcal{C}) = \mathtt{gates}(\mathcal{C}) + \alpha \cdot \mathtt{inputs}(\mathcal{C}),$$

где $0 < \alpha \leq 1$ — константа, которая будет выбрана позже. Теорема 46 следует из этой леммы при $S = \mathbb{F}_2^n$, которое является $(n, 0)$-квадратичным многообразием.

Лемма 48. *Пусть $f \in B_n$ — (n, k, s)-квадратичный дисперсер, $S \subseteq \mathbb{F}_2^n$ — (n, t)-квадратичное многообразие, $0 < \alpha \leq 1, 0 < \beta$ — константы, удовлетворяющие неравенствам* (1.3), (1.4), (1.5), (1.6), *$\mathcal{C}$ — XOR-схема, вычисляющая f на S. Тогда*

$$\mu(C) \geq \min\left\{\beta(\log_2 |S| - \log_2 s - 1), 2(k-t)\right\}\,.$$

Доказательство. Доказательство проводится индукцией по $|S|$. Базовый случай $|S| \leq 2s$ выполняется по очевидным соображениям. Для перехода

допустим, что $|S| > 2s$. Если $t \geq k$, то правая часть неравенства не больше нуля, поэтому предположим, что $t < k$. Допустим, что схема $\mathcal{C}$ оптимальна относительно меры μ (то есть $\mathcal{C}$ имеет минимальное значение μ среди всех схем, вычисляющих f на S). Мы найдём элемент G в схеме $\mathcal{C}$, вычисляющий функцию g степени не более 2 и рассмотрим два $(n, t+1)$-квадратичных многообразия S: $S_0 = \{x \in S\colon g(x) = 0\}$ и $S_0 = \{x \in S\colon g(x) = 1\}$. Пусть $|S_0| = p_0|S|$ и $|S_1| = p_1|S|$, где $0 < p_0, p_1 < 1$ и $p_0 + p_1 = 1$ (заметим, что $p_i = 0$ или $p_i = 1$ означало бы, что G вычисляет константу на S, что противоречило бы тому, что $\mathcal{C}$ оптимальна). Устранив из $\mathcal{C}$ все элементы, которые не зависят от хотя бы от одного из своих входов на S_i, получим схему $\mathcal{C}_i$, вычисляющую f на S_i. Допустим, что $\mu(\mathcal{C}) - \mu(\mathcal{C}_i) \geq \Delta_i$. Тогда по предположению индукции

$$\begin{aligned}\mu(\mathcal{C}) &\geq \mu(\mathcal{C}_i) + \Delta_i \geq \min\left\{\beta(\log_2|S_i| - \log_2 s - 1), 2(k-(t+1))\right\} + \Delta_i \\ &= \min\left\{\beta\left(\log_2|S| - \log_2 s - 1\right) + \left(\Delta_i - \beta\log_2\frac{1}{p_i}\right), 2(k-t) + (\Delta_i - 2)\right\}.\end{aligned}$$

Следовательно, если $\Delta_i \geq \beta\log_2\frac{1}{p_i}$ и $\Delta_i \geq 2$ при $i = 0$ или $i = 1$, то требуемое неравенство следует по предположению индукции. Неравенство $\Delta_i \geq \beta\log_2 1/p_i$ верно, если $p_i \geq 2^{-\frac{\Delta_i}{\beta}}$. Поскольку мы хотим, чтобы это неравенство выполнялось хотя бы для одного из $i = 0$ и $i = 1$ и поскольку $p_0 + p_1 = 1$, заключаем, что достаточно, чтобы выполнялось такое неравенство:

$$2^{-\frac{\Delta_0}{\beta}} + 2^{-\frac{\Delta_1}{\beta}} \leq 1 \text{ и } \Delta_0, \Delta_1 \geq 2\,. \tag{1.9}$$

Рассмотрев несколько случаев, мы покажем, что всегда найдётся элемент G, для которого соответствующие Δ_0 и Δ_1 удовлетворяют неравенству (1.9). Для этого мы будем использовать неравенства (1.3)–(1.8).

Для начала покажем, что схема $\mathcal{C}$ обязана быть непустой (напомним, что $\mathcal{C}$ является XOR-схемой, что означает, что её входами являются не только переменные, но и произвольные аффинные комбинации переменных). Действи-

тельно, если $\mathcal{C}$ пуста, то она вычисляет линейную функцию l. Следовательно f — константа и на $S_0 = \{x \in S\colon l(x) = 0\}$, и на $S_1 = \{x \in S\colon l(x) = 1\}$. Однако $\max\{|S_0|, |S_1|\} \geq |S|/2 > s$, что противоречит тому факту, что f является (n, k, s)-квадратичным дисперсером.

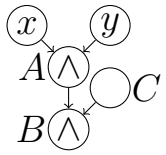
Пусть A — $\wedge$-элемент с максимальным числом $\wedge$-элементов от него до выхода схемы $\mathcal{C}$. Другими словами, для фиксированного $\wedge$-элемента мы рассматриваем все ориентированные пути от этого элемента до выхода и выбираем среди таких путей тот, на котором больше всего $\wedge$-элементов; после этого мы выбираем $\wedge$-элемент, для которого этот показатель максимален. Поскольку $\mathcal{C}$ — XOR-схема, можно считать, что A — минимальный элемент, то есть его непосредственными предками являются входные элементы. Обозначим эти входные элементы через x и y.

Случай 1. $\mathrm{outdeg}(x) = \mathrm{outdeg}(y) = 1$.

Случай 1.1. $\mathrm{outdeg}(A) = 1$ и из A идёт провод в $\wedge$-элемент B.

Пусть C — второй вход B (он может быть входным или внутренним элементом).

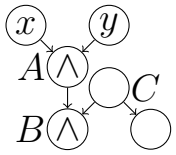
Случай 1.1.1. $\mathrm{outdeg}(C) = 1$.



Тривиализируем A соответствующей квадратичной подстановкой. Тогда элемент B удалится. Более тога, $A = 0$ или $A = 1$ тривиализирует также B, поэтому все его потомки, а также элемент C тоже удаляются (поскольку C используется только для вычисления B, а B стал константой). В обоих случаях x и y также элиминируются (единственный элемент A, в который шли провода из этих входов, теперь вычисляет константу). Получаем $\{\Delta_0, \Delta_1\} = \{2{+}2\alpha, 3{+}3\alpha\}$ (здесь и далее при удалении элемента, про который неизвест-

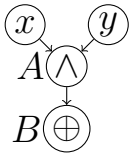
но, входной он или внутренний, мы будем считать, что он даёт вклад хотя бы α в уменьшение меры; так можно делать, поскольку $\alpha < 1$). Необходимые неравенства (1.9) следуют из (1.5).

Случай 1.1.2. $\mathrm{outdeg}(C) \geq 2$.



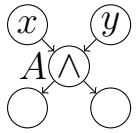
По выбору A, элемент C вычисляет функцию степени не выше 2. Сделаем C константой. В обоих случаях мы удаляем потомков C и сам элемент C. Это уменьшает меру хотя бы на $2 + \alpha$. В одном из случаев B также становится константой, что удаляет B, A, и входы x и y. Получаем $\{\Delta_0, \Delta_1\} = \{2 + \alpha, 4 + 3\alpha\}$. Данные Δ_0, Δ_1 удовлетворяют неравенствам (1.9) по (1.3).

Случай 1.2. $\mathrm{outdeg}(A) = 1$ и из A идёт провод в $\oplus$-элемент B.



По выбору A второй вход B вычисляет функцию степени не выше 2. Поэтому сам B вычисляет функцию степени не более 2. Сделаем B константой. Это удаляет B и его потомков. Элемент A и входы x и y также удаляются. Следовательно, $\Delta_0 = \Delta_1 = 3 + 2\alpha$. Неравенства (1.9) выполняются по (1.8).

Случай 1.3. $\mathrm{outdeg}(A) \geq 2$.



Сделав A константой, получаем $\Delta_0 = \Delta_1 = 3 + 2\alpha$, поскольку$A$ и все его потомки (хотя бы два элемента) удаляются. Аналогично предыдущем случаю, неравенство (1.8) гарантирует, что неравенство (1.9)

верно.

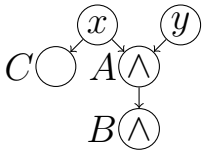
Случай 2. Исходящая степень x или y хотя бы 2. Не умаляя общности, $\mathrm{outdeg}(x) \geq 2$.

Случай 2.1. $\mathrm{outdeg}(A) = 1$ и из A идёт провод в $\wedge$-элемент B.

Сделаем A константой. Допустим, что A вычисляет $(x \oplus c_1)(y \oplus c_2) \oplus c$. Тогда A может равняться $c \oplus 1$, только если $x = c_1 \oplus 1$ и $y = c_2 \oplus 1$. То есть когда A равно $c \oplus 1$, удаляется не только его потомок, но и потомки x и y. В обоих случаях B удаляется, но в одном из них он становится константой и удаляются также и все его потомки.

Обозначим через C другой элемент, в который идёт провод из x. Заметим, что $B \neq C$ (иначе схема не была бы оптимальной).

Случай 2.1.1. $\mathrm{outdeg}(y) = 1$.



Случай 2.1.1.1. B становится константой, когда $A = c$.

Если $A = c$, то удаляются A, B, потомки B и y. Если $A = c \oplus 1$, то удаляются A, B, C, x, и y. Следовательно $\{\Delta_0, \Delta_1\} = \{3 + \alpha, 3 + 2\alpha\}$. Неравенство (1.8) гарантирует, что неравенство (1.9) выполнено.

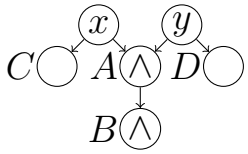
Случай 2.1.1.2. B становится константой, когда $A = c \oplus 1$.

Если $A = c$, мы удаляем A, B, и y. Если $A = c \oplus 1$, то удаляются A, B, C, потомки B, x, и y (если C является единственным потомком B, тогда он становится константой и удаляются также и его потомки). Поэтому $\{\Delta_0, \Delta_1\} = \{2 + \alpha, 4 + 2\alpha\}$. Неравенство (1.9) выполнено по (1.3).

Случай 2.1.2. $\mathrm{outdeg}(y) \geq 2$.

Обозначим через D другого потомка y. Заметим, что D может

совпадать с C, но $D \neq B$.



Случай 2.1.2.1. B становится константой, когда $A = c$.

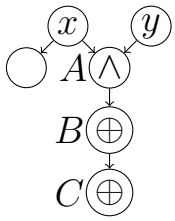
Если $A = c$, то удаляются A, B и потомки B. Если $A = c \oplus 1$, удаляются A, B, C, D, x, и y. Если $C = D$, то данный элемент становится константой и все его потомки удаляются. Следовательно, $\{\Delta_0, \Delta_1\} = \{3, 4 + 2\alpha\}$. Неравенство (1.9) выполнено по (1.6).

Случай 2.1.2.2. B становится константой, когда $A = c \oplus 1$.

Если $A = c$, удаляются A и B. Если $A = c \oplus 1$, удаляются A, B, C, D, потомки B, x и y. В этом случае нужно аккуратно показать, что удалятся хотя бы пять элементов (поскольку некоторые из перечисленных пяти могут совпадать). Если $C \neq D$ и, скажем, C является потомком B, то C становится константой и удаляются все его потомки. Если $C = D$, то C становится константой и все его потомки удаляются. Следовательно, $\{\Delta_0, \Delta_1\} = \{2, 5 + 2\alpha\}$. Неравенства (1.4) влекут (1.9).

Случай 2.2. $\mathrm{outdeg}(A) = 1$ и из A идёт провод в $\oplus$-элемент B.

Случай 2.2.1. $\mathrm{outdeg}(B) = 1$ и из B идёт провод в $\oplus$-элемент C.

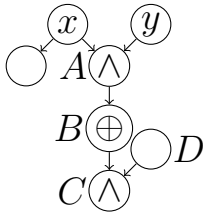


По выбору A, элемент C вычисляет квадратичную функцию. Сделаем C константой. В обоих случаях удаляются A, B, C и потомки C. Значит, $\Delta_0 = \Delta_1 = 4$. Неравенства (1.9) выполнены по (1.7).

Случай 2.2.2. $\mathrm{outdeg}(B) = 1$ и из B идёт провод в $\wedge$-элемент C.

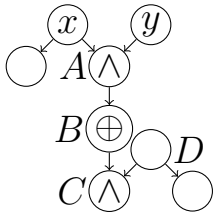
Пусть D — второй предок C. Если $D = A$, то схема неоптимальна (C зависит от A и второго входа B, поэтому можно вычислить C напрямую, без использования B).

Случай 2.2.2.1. $\mathrm{outdeg}(D) = 1$.



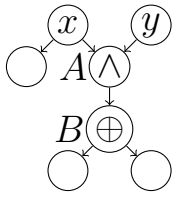
Сделаем B константой. В обоих случаях удаляются A, B и C. Более того, когда B равен константе, тривиализирующей C, удаляются также D и потомки C. Элемент D вносит вклад (в уменьшение меры) $\alpha \leq 1$, если он является входом, и 1, если он является внутренним элементом. Следовательно, $\{\Delta_0, \Delta_1\} = \{3, 4 + \alpha\}$. Неравенство (1.6) гарантирует выполнение неравенств (1.9).

Случай 2.2.2.2. $\mathrm{outdeg}(D) \geq 2$.



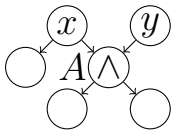
Сделаем D константой (мы можем это сделать, поскольку этот элемент считает функцию степени не выше 2). В обоих случаях мы удаляем D и его потомков и уменьшаем меру хотя бы на $2 + \alpha$ (поскольку D может быть входом). В случае, когда C становится константой, удаляется также C и A, B. Таким образом, $\{\Delta_0, \Delta_1\} = \{2 + \alpha, 5 + \alpha\}$ (все пять элементов различны, поскольку если из D идёт провод в B или потомка C, тогда схема неоптимальна). Неравенства (1.9) выполнены в силу (1.3) и $\alpha \leq 1$.

Случай 2.2.3. $\mathrm{outdeg}(B) \geq 2$.



Элемент B вычисляет функцию степени не более 2. Сделав его константой, мы удаляем B, его потомков и A, поэтому $\Delta_0 = \Delta_1 = 4$. Неравенства (1.9) выполнены по (1.7).

Случай 2.3. $\mathrm{outdeg}(A) \geq 2$.



Сделаем A константой. В обоих случаях удаляется A и его потомки. Когда x и y становятся константами (напомним, что если A вычисляет $(x \oplus c_1)(y \oplus c_2) \oplus c$, то $A = c \oplus 1$ влечёт за собой $x = c_1 \oplus 1$ и $y = c_2 \oplus 1$), удаляется ещё хотя бы один потомок x. Таким образом, $\{\Delta_0, \Delta_1\} = \{3, 4 + 2\alpha\}$. Неравенство (1.6) гарантирует, что неравенства (1.9) выполнены.

□

1.5 Нижняя оценка $5n - o(n)$ в базисе U_2 для линейных функций

1.5.1 Линейные функции

В данном разделе мы рассматриваем только схемы над U_2. Напомним, что такие схемы не содержат элементов типа $\oplus$. Основное свойство элемента такой схемы заключается в следующем: если в этот элемент идёт провод из

переменной, то этой переменной можно присвоить константу так, что элемент начнёт вычислять тоже константу.

Будем рассматривать линейные функции $f \in B_{n,m}$ вида $f(x) = Ax \oplus b$, где $A \in \mathbb{F}_2^{m \times n}$ — матрица с n различными ненулевыми столбцами, а $b \in \{0,1\}^m$ — произвольный вектор. Сразу же отметим, что при присвоении значения любой переменной мы получим такую же функцию (при этом удаляется соответствующий столбец из A и изменяются некоторые биты вектора b; при этом все столбцы матрицы по-прежнему ненулевые и различны). Это позволяет нам доказывать нижнюю оценку по индукции.

Основным результатом данного раздела является следующая оценка.

Теорема 49. *Пусть $A \in {\mathbb{F}_2}^{m \times n}$ — матрица с n различными ненулевыми столбцами, а $b \in \mathbb{F}_2^m$ — произвольный вектор. Определим $f \in B_{n,m}$ как $f(x) = Ax \oplus b$. Тогда*

$$\mathtt{gates}_{U_2}(f) \geq 5(n-m)\,.$$

В частности. при $n = 2^k - 1$ для функции $f \in B_{n,k}$, матрица которой является проверочной матрицей кода Хэмминга (все её столбцы попарно различны и ненулевые), получаем нижнюю оценку $5n - o(n)$. Это совпадает с наилучшей известной на сегодняшней день нижней оценкой $5n - o(n)$ для базиса U_2, полученной К. Ивамой и Х. Морицумой в 2002 г. [27]. Отметим, что рассматривающаяся здесь функция имеет $o(n)$ выходов в отличие от функции К. Ивамы и Х. Морицумы, являющейся предикатом. Более сильных оценок для функций с $o(n)$ выходами неизвестно.

1.5.2 Доказательство нижней оценки

Перед доказательством основного результата мы докажем две элементарные леммы о схемах, вычисляющих линейные функции.

Лемма 50. *Пусть $f \in B_{n,m}$ — функция вида $f(x) = Ax \oplus b$, где $A \in \mathbb{F}_2^{m \times n}$ — матрица с n различными ненулевыми столбцами, а $b \in \{0,1\}^m$ — произвольный вектор. Если какая-то переменная схемы, вычисляющей f, имеет исходящую степень один, то схема неоптимальна.*

Доказательство. Предположим, что исходящая степень переменной x_i равна одному.

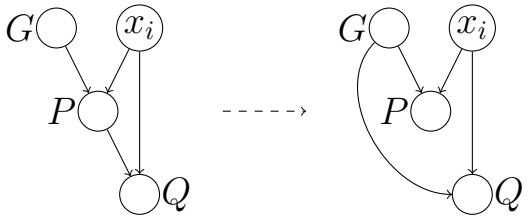
Если все выходы функции зависят только от x_i (то есть каждый выход считает функцию x_i или $\neg x_i$) или все выходы не зависят от x_i вообще, то x_i не требуется ни для какого элемента. В данном случае мы можем подставить константу вместо x_i в соответствующий элемент.

В противном случае найдётся хотя бы один выход, зависящий от x_i и ещё хотя бы одной другой переменной. Обозначим через G другой вход элемента, в который ведёт x_i. Ясно, что G не зависит от x_i. Рассматриваемый элемент вычисляет функцию вида $(x_i \oplus a)(G \oplus b) \oplus c$. Допустим, существует присваивание всем переменным кроме x_i, при котором G становится равным b. В такой ситуации этот элемент становится константой и вся схема теряет зависимость от x_i. Это противоречит тому, что при такой подстановке рассматриваемый выход всё ещё зависит от x_i. Следовательно, при любой подстановке G равен $b \oplus 1$, то есть G — константный элемент. Это, в частности, означает, что схема неоптимальна. □

Лемма 51. *Пусть в схеме в элементы P и Q ведут провода из переменной x_i и пусть также в Q идёт провод из P. Тогда можно перестроить схему, не изменяя её размера, так что P и Q перестанут быть связаны проводом.*

Доказательство. Обозначим через G второго непосредственного предка элемента P. Тогда Q зависит от G и x_i. Ясно, что Q не может вычислять функцию $\oplus$-типа от x_i и G, поскольку для этого потребовалось бы хотя бы три элемента. Значит, мы можем изменить функцию, вычисляемую в Q, и доба-

вить провод в Q не из P, а прямо из G. Описанное преобразование показано ниже.



□

Доказательство теоремы 49. Доказательство ведётся индукцией по $n - m$.

Есть два случая, в которых утверждение леммы выполняется по очевидным причинам: $n \leq m$ и $m = 1$ (если $m = 1$, то $n = 1$, поскольку все столбцы должны быть различны и ненулевые).

Предположим теперь, что $n > m$ и $m > 1$. Если есть строка, в которой всего одна единица (скажем, в i-м столбце), то соответствующий выход зависит только от x_i. В таком случае сделаем подстановку $x_i \leftarrow 0$. При этом удалится i-й столбец из матрицы вместе с соответствующей строкой. При этом также пропадает выходной элемент схемы. Легко видеть, что полученная схема по-прежнему вычисляет полученную функцию. Ясно также, что все столбцы матрицы по-прежнему различны и не равны нулевому.

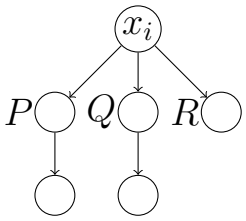
Пусть теперь все строки A содержат хотя бы по две единицы. Рассмотрим оптимальную схему, вычисляющую f. Ни один из выходов не равен входной переменной. Отметим также следующее свойство функции f: даже если присвоить константы всем переменным кроме x_i, останется хотя бы один выход, который зависит от x_i. Также по лемме 50 все переменные имеют исходящую степень хотя бы два.

Рассмотрим верхний элемент P, то есть элемент, в который входят переменные x_i и x_j. Поскольку схема оптимальна, эти переменные различны и каждая из них входит ещё хотя бы в один элемент.

Ниже мы рассмотрим несколько случаев. В каждом из них мы находим

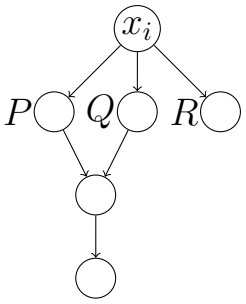
переменную, при присвоении константы которой из схемы удаляются хотя бы пять элементов. Отметим, что элементы, которые при этом становятся константными, не могут быть выходными, поскольку все выходы зависят хотя бы от двух переменных. Поэтому у каждого такого элемента есть хотя бы один потомок. Лемма 51 также гарантирует, что если два элемента имеют непосредственным предком одну и ту же переменную, то они не связаны проводом.

Случай 1. Одна из x_i и x_j (скажем, x_i) имеет исходящую степень три или более (назовём три элемента P, Q, R). Тогда для некоторой константы $c \in \mathbb{F}_2$ подстановка $x_i \leftarrow c$ делает хотя бы два из этих трёх элементов константными (скажем, P и Q). Следовательно, эта подстановка удаляет P, Q, R и всех потомков P и Q.



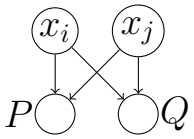
Случай 1.1. Если общее число потомков P и Q хотя бы 2, тогда хотя бы пять элементов будут удалены.

Случай 1.2. Если у P и Q есть ровно один потомок, то этот потомок тоже становится константой при подстановке (если этим потомком оказывается элемент R, то он тоже становится константным и удаляется также и его потомок).



Случай 2. И x_i, и x_j имеют исходящую степень два. Обозначим других потомков переменных x_i и x_j через R и Q, соответственно.

Случай 2.1. $Q = R$.



Покажем, что этот случай невозможен. Действительно, поскольку P и Q являются элементами $\wedge$-типа,

$$\begin{aligned} P &= (x_i \oplus a_1)(x_j \oplus b_1) \oplus c_1\,, \\ Q &= (x_i \oplus a_2)(x_j \oplus b_2) \oplus c_2\,, \end{aligned}$$

для некоторых констант $a_1, b_1, c_1, a_2, b_2, c_2 \in \mathbb{F}_2$. Заметим, что $a_1 \neq a_2$. В противном случае подстановка $x_i \leftarrow a_1$ сделала бы схему независимой от x_j. По точно той же причине $b_1 \neq b_2$. Но тогда схема не различает подстановки $\{x_i \leftarrow a_1, x_j \leftarrow b_1 \oplus 1\}$ и $\{x_i \leftarrow a_1 \oplus 1, x_j \leftarrow b_1\}$ (при этих подстановках $P = c_1$ и $Q = c_2$). Такого не может быть, поскольку из различности всех столбцов матрицы A следует, что у функции обязательно есть выход, который зависит от одной из рассматриваемых двух переменных и не зависит от другой.

Случай 2.2. $Q \neq R$. Поскольку схема является ациклическим графом, все её элементы можно топологически упорядочить. Зафиксируем некоторый такой порядок и предположим, что R в нём предшествует Q. Это, в частности, означает, что R не зависит от Q.

Покажем, что для некоторой константы $c \in \mathbb{F}_2$ подстановка $x_j \leftarrow c$ делает оба элемента P и R константами. Обозначим через G другой вход R. Поскольку P и R являются элементами $\wedge$-типа, найдутся

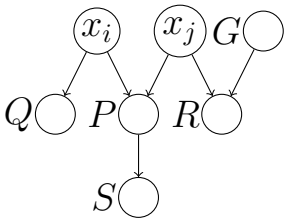
константы $a_1, b_1, c_1, a_2, b_2, c_2 \in \mathbb{F}_2$, такие что

$$\begin{aligned} P &= (x_j \oplus a_1)(x_i \oplus b_1) \oplus c_1\,, \\ R &= (x_j \oplus a_2)(g \oplus b_2) \oplus c_2\,. \end{aligned}$$

Заметим, что относительно подстановки $x_i \leftarrow b_1$ элемент P превращается в константу c_1. Тогда если для некоторой подстановки всем переменным кроме x_j на выходе элемента G появляется константа b_2, то элемент R тоже превращается в константу c_2 и схема теряет зависимость от x_j, что приводит к противоречию. Таким образом, из $x_i = b_1$ следует $G = b_2 \oplus 1$. Из этого, в свою очередь, следует, что $x_j = a_1$ влечёт за собой $G = b_2 \oplus 1$. Действительно, в схеме нет пути из Q в G (из предположения, что R предшествует Q), следовательно, G может зависеть от x_i только через P. А мы знаем, что из $P = c_1$ следует $G = b_2 \oplus 1$.

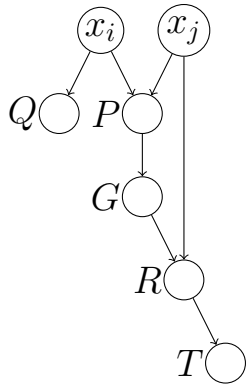
Таким образом, при подстановке $x_j \leftarrow a_1$ элементы P и G превращаются в константы, R также становится константой, поскольку оба его входа стали константными. Все потомки P, G, R удаляются. Также в полученной схеме исходящая степень переменной x_i равна 1, поэтому по лемме 50 может быть удалён ещё хотя бы один элемент. Чтобы показать, что в этом случае всегда удаляются хотя бы пять элементов, мы рассматриваем два подслучая в зависимости от потомков P. Обозначим через S какого-нибудь потомка P. Отметим, что лемма 51 гарантирует, что $S \neq Q$ и $S \neq R$.

Случай 2.2.1. $S \neq G$.



Тогда подстановка $x_j \leftarrow a_1$ удаляет элементы P, R, G, S. После этого ещё хотя бы один элемент удаляется благодаря лемме 50.

Случай 2.2.2. $S = G$.



Заметим, что если хотя бы один из P, G, R имеет исходящую степень больше одного, мы снова удаляем хотя бы пять элементов. Пусть теперь у всех трех элементов исходящая степень равна одному. Обозначим единственного потомка R через T и рассмотрим второй вход элемента T. Он не является константой и не зависит от x_j. Тогда соответствующей подстановкой всем переменным кроме x_j мы можем сделать элемент T константой и сделать схему независимой от x_j, противоречие.

□

1.6 Нижняя оценка $3.24n$ на схемную сложность в среднем в базисе U_2 для дисперсера относительно проекций

Большинство известных *нижних оценок* на размер бинарных булевых схем (без ограничений на глубину и на исходящую степень элементов) доказано методом элиминации элементов. Наиболее эффективные известные *алгоритмы* для задачи выполнимости булевых схем используют разбор слу-

чаев, аналогичный методу элиминации элементов. Р. Ченем и В. Кабанцом в 2015 г. [8] было показано, что такой разбор случаев может также быть использован для получения нижних оценок на *схемную сложность в среднем случае*.

В данном разделе мы, используя идеи Р. Ченя и В. Кабанца, доказываем общую теорему, которая по данному разбору случаев автоматически даёт нижние оценки на схемную сложность в среднем и худшем случаях, а также верхнюю оценку для задачи #CircuitSAT, заключающуюся в нахождении числа выполняющих наборов данной булевой схемы. Доказательство перечисленных оценок с использованием такой теоремы проводится следующим образом. Сначала фиксируются три параметра: класс схем (мы будем рассматривать только два класса: схемы над полным бинарным базисом B_2 и над базисов $U_2 = B_2 \setminus \{\oplus, \equiv\}$), мера сложности схем и класс разрешённых подстановок. Далее разбором случаев доказывается, что для любой схемы из заданного класса найдётся разрешённая подстановка, уменьшающая заданную меру значительно. Данный разбор случаев сразу даёт верхнюю оценку на время работы алгоритма для задачи #CircuitSAT. Для получения нижних оценок на схемную сложность в среднем/худшем случае необходимо также представить явную конструкцию экстрактора/дисперсера относительно разрешённых подстановок.

Используя данный метод, мы получаем алгоритм, который решает задачу #CircuitSAT для данной схемы над U_2 с n входами и не более чем $3.24n$ элементами за время $(2 - \delta)^n$, где $\delta > 0$ — константа. Мы также получаем нижнюю оценку $3.24n$ на схемную сложность в среднем явно заданной булевой функции. Обе полученные оценки улучшают оценки Р. Ченя и В. Кабанца и являются самыми сильными на данный момент.

1.6.1 Предварительные определения и леммы

1.6.1.1 Подстановки, дисперсеры и экстракторы

В данном разделе нам понадобится чуть более общее (чем использованное ранее) определение дисперсеров и экстракторов.

Определение 52 (подстановки)**.** *Для множества функций* $\mathcal{F} = \{f\colon \mathbb{F}_2^* \to \mathbb{F}_2\}$ *определим множество* $\mathcal{S}(\mathcal{F})$ *как следующее множество подстановок: для всех* $1 \le i \le n, c \in \mathbb{F}_2, f \in \mathcal{F}$*, данное множество* $\mathcal{S}$ *содержит подстановку* $x_i \leftarrow f(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \oplus c$.

Определение 53 ($(\mathcal{S}, n, r)$-источник)**.** *Пусть* $\mathcal{S}$ *— множество подстановок. Множество* $K \subseteq \mathbb{F}_2^n$ *называется* $(\mathcal{S}, n, r)$-источником*, если его можно получить из* $\mathbb{F}_2^n$ *применением не более чем* r *подстановок из* $\mathcal{S}$*. (Обычно источником называется распределение на подмножестве* $\mathbb{F}_2^n$*. В данной работе мы рассматриваем только равномерные распределения, поэтому отождествляем источник с его носителем.)*

Определение 54 ($(\mathcal{S}, n, r)$-дисперсер)**.** *Функция* $f \in B_n$ *называется* $(\mathcal{S}, n, r)$-дисперсером*, если она не обращается в константу ни на каком* $(\mathcal{S}, n, r)$*-источнике. Функция* $f \in B_n$ *называется* $(\mathcal{S}, n, r, \varepsilon)$-экстрактором*, если* $|\Pr_{x \leftarrow K}[f(x) = 1] - 1/2| \le \varepsilon$ *для любого* $(\mathcal{S}, n, r)$*-источника* K.

Четыре уже рассмотренных в данной работе класса источников/дисперсеров таковы:

1. Дисперсеры/экстракторы, устойчивые относительно класса константных подстановок $\mathcal{S} = \{x_i \leftarrow c\}$, имеют множество приложений в криптографии (обзор можно найти в [16]). Стандартной функцией, являющейся дисперсером и экстрактором относительно таких источников, является функция чётности $x_1 \oplus \cdots \oplus x_n$.

2. Дисперсеры относительно класса проекций $\mathcal{S} = \{x_i \leftarrow c, x_i \leftarrow x_j \oplus c\}$ используются для доказательства сильных нижних оценок для схем глубины три [43]. Как показано в [43], двоичный код Боуза–Чоудхури–Хоквингемас подходящими параметрами является дисперсером относительно $n - o(n)$ проекций. В работе [45] также строятся экстракторы относительно проекций.

3. Аффинные подстановки $\mathcal{S} = \{x_i \leftarrow \bigoplus_{j \in J} x_j \oplus c\}$ задают аффинные дисперсеры, уже обсуждавшиеся ранее в этой работе.

4. Квадратичные подстановки $\mathcal{S} = \{x_i \leftarrow p\colon \deg(p) \leq 2\}$ задают квадратичные дисперсеры, которые также ранее уже обсуждались.

1.6.1.2 Корреляция двух функций

Определение 55 (корреляция)**.** *Корреляция двух функций $f, g \in B_n$ определяется как*

$$\begin{aligned}\mathrm{Cor}(f,g) &= \left| \Pr_{x \leftarrow \{0,1\}^n}[f(x) = g(x)] - \Pr_{x \leftarrow \mathbb{F}_2^n}[f(x) \neq g(x)] \right| = \\ &= 2 \cdot \left| \frac{1}{2} - \Pr_{x \leftarrow \mathbb{F}_2^n}[f(x) \neq g(x)] \right|.\end{aligned}$$

Для функции $f \in B_n$ и параметра $0 \leq \varepsilon \leq 1$ через $\mathtt{gates}_\Omega(f, \varepsilon)$ обозначим минимальный размер схемы над базисом Ω, вычисляющей функцию g, такую что $\mathrm{Cor}(f, g) \geq \varepsilon$.

1.6.1.3 Вектора расщепления

Определение 56 (вектор расщепления и число расщепления)**.** *Рассмотрим меру сложности схем μ и схему $\mathcal{C}$. Рассмотрим рекурсивный алгоритм, решающий задачу #CircuitSAT для схемы $\mathcal{C}$ методом расщепления. Пусть на данном шаге алгоритм выбирает k переменных $x_1, \ldots, x_k$ и k функций*

$f_1, \ldots, f_k$ и расщепляется на 2^k случаев:

$$x_1 \leftarrow f_1 \oplus c_1, \ldots, x_k \leftarrow f_k \oplus c_k$$

для всех возможных $c_1, \ldots, c_k \in \mathbb{F}_2$ (другими словами, алгоритм разбивает булев куб $\mathbb{F}_2^n$ на 2^k подмножеств). Для каждой из этих подстановок алгоритм упрощает схему (удаляет константные и проходные элементы) и продолжает работу рекурсивно. Обозначим полученные 2^k схем через $\mathcal{C}_1, \ldots, \mathcal{C}_{2^k}$. Будем говорить, что данному шагу алгоритма соответствует вектор расщепления *$v = (a_1, \ldots, a_{2^k})$ относительно меры μ, если для всех $1 \leq i \leq 2^k$, $\mu(\mathcal{C}) - \mu(\mathcal{C}_i) \geq a_i > 0$. Таким образом, вектор расщепления задаёт нижние оценки на уменьшение меры.* Числом расщепления $\tau(v)$ *называется единственный положительный корень уравнения $\sum_{i \in [2^k]} x^{-a_i} = 1$.*

Вектора расщепления активно используются при анализе алгоритмов расщепления (для многих NP-трудных задач именно с помощью метода расщепления построены самые эффективные известные алгоритмы).

В данной работе мы будем рассматривать только одну или две подстановки подряд (то есть $k = 1$ или $k = 2$). Если каждому шагу алгоритма соответствует число расщепления не более β, то его время работы не больше $O^*(\beta^{\mu(\mathcal{C})})$ ($O^*(\cdot)$ скрывает множители, зависящие полиномиально от размера входа). Чтобы доказать это, заметим, что дерево рекурсии алгоритма имеет коэффициент ветвления 2^k. Поскольку $k = O(1)$, достаточно оценить количество листьев в дереве. Обозначим его через $T(\mu)$. Оно удовлетворяет рекуррентному соотношению

$$T(\mu) \leq \sum_{i \in [2^k]} T(\mu - a_i),$$

которое влечёт за собой $T(\mu) = O(\tau(v)^\mu)$ (мы также предполагаем, что $T(\mu) = O(1)$, если $\mu = O(1)$). Формальное доказательство можно найти,

например, в работе О. Кульманна [34].

Определение 57. *Для вектора расщепления $v = (a_1, \ldots, a_{2^k})$ введём следующие величины:*

$$\overline{v}_{\max} = \max_{i \in [2^k]} \left\{ \frac{a_i}{k} \right\}, \quad \overline{v}_{\min} = \min_{i \in [2^k]} \left\{ \frac{a_i}{k} \right\}, \quad \overline{v}_{\text{avg}} = \frac{\sum_{i \in [2^k]} a_i}{k 2^k}.$$

Интуитивно $\overline{v}_{\max}$ ($\overline{v}_{\min}$, $\overline{v}_{\text{avg}}$) — это максимальное (минимальное, среднее, соответственно) уменьшение меры на одну подстановку.

Нам понадобятся следующие оценки на числа расщепления.

Лемма 58 ([34])**.** *Сбалансированный вектор расщепления даёт лучшее число расщепления, чем несбалансированный:*

$$2^{1/a} = \tau(a, a) < \tau(a + b, a - b) \text{ для } 0 < b < a.$$

Верхняя оценка на число расщепления:

$$\tau(a, b) \leq 2^{1/\sqrt{ab}}.$$

В следующей лемме мы доказываем асимптотическую оценку на разность приведённых выше двух оценок:

Лемма 59. *Пусть $a_1 > a_2 > 0$, $a' = (a_1 + a_2)/2$ и $\delta(b) = \tau(a_1 + b, a_2 + b) - 2^{\frac{1}{a'+b}}$. Тогда*

$$\delta(b) = O((a_1 - a_2)^2 / b^3) \text{ при } b \to \infty.$$

Доказательство. Пусть $x = \tau(a_1 + b, a_2 + b)$. Тогда

$$1 = \frac{1}{x^{a_1+b}} + \frac{1}{x^{a_2+b}} = \frac{x^{-(a_1-a_2)/2} + x^{(a_1-a_2)/2}}{x^{a'+b}}.$$

Поскольку

$$x = 2^{\frac{1}{a'+b}} + \delta(b) = 1 + \frac{\ln 2}{a'+b} + \delta(b) + O\left(\frac{1}{(a'+b)^2}\right)$$

и

$$(1+\epsilon)^{(a_1-a_2)/2} = 1 + (a_1-a_2)\epsilon/2 + (a_1-a_2)(a_1-a_2-1)\epsilon^2/4 + O(\epsilon^3),$$

имеем

$$x^{-(a_1-a_2)/2} + x^{(a_1-a_2)/2} = \\ = 2 + \frac{(a_1-a_2)^2}{2}\left(\frac{\ln 2}{a'+b} + \delta(b)\right)^2 + O\left(\left(\frac{\ln 2}{a'+b} + \delta(b)\right)^3\right).$$

Также

$$x^{a'+b} = 2\left(1 + \delta(b)/2^{\frac{1}{a'+b}}\right)^{a'+b} = 2\left(1 + (a'+b)\delta(b)/2^{\frac{1}{a'+b}} + O(\delta(b)^2)\right).$$

По определению x имеем

$$\lim_{b\to\infty} \frac{(a_1-a_2)^2 \ln^2 2}{2b^2}/(2b\delta(b)) = 1.$$

Следовательно,

$$\delta(b) = \frac{(a_1-a_2)^2 \ln^2 2}{4b^3} + o(1/b^3).$$

□

1.6.1.4 Неравенство Ацумы

Следуя подходу Р. Ченя и В. Кабанца, мы используем вариант неравенства Ацумы для доказательства нижних оценок на схемную сложность в среднем. В стандартной версии данного неравенства необходимо, что был

ограничен модуль разности двух соседних случайных величин. Р. Чень и В. Кабанец рассматривают вариант, когда разность принимает всего два различных значения и ограничена с одной из сторон. Нам понадобится чуть более общий вариант: разность принимает не более k различных значений и ограничена с одной из сторон. Приводящееся ниже доказательство — адаптация стандартных доказательств из [38, 1, 8].

Определение 60 (супермартингал)**.** *Последовательность $X_0, \ldots, X_m$ случайных величин называется* супермартингалом*, если для всякого $0 \le i < m$ выполнено*

$$\mathrm{E}[X_{i+1} \mid X_i, \ldots, X_0] \le X_i \,.$$

Лемма 61. *Пусть $X_0, \ldots, X_m$ — супермартингал и пусть $Y_i = X_i - X_{i-1}$. Если $Y_i \le c$ и если при фиксированных значения величин $(X_0, \ldots, X_{i-1})$ величина Y_i распределена равномерно на не более чем $k \ge 2$ (необязательно различных) значениях, тогда для любого $\lambda \ge 0$ выполнено*

$$\Pr[X_m - X_0 \ge \lambda] \le \exp\left(\frac{-\lambda^2}{2mc^2(k-1)^2}\right).$$

Отметим, что в данном варианте (по сравнению со стандартным вариантом неравенства Ацумы) появляется дополнительный множитель $(k-1)^2$, но мы не предполагаем, что $X_i - X_{i-1}$ ограничено снизу.

Доказательство. Для любого $t > 0$,

$$\Pr\left[X_m - X_0 \ge \lambda\right] = \Pr\left[\sum_{i=1}^{m} Y_i \ge \lambda\right] =$$
$$\Pr\left[\exp\left(t \cdot \sum_{i=1}^{m} Y_i\right) \ge e^{\lambda t}\right] \le e^{-\lambda t} \cdot \mathrm{E}\left[\exp\left(t \cdot \sum_{i=1}^{m} Y_i\right)\right].$$

Для начала покажем, что для любого $t > 0$ выполнено

$$\mathrm{E}[e^{tY_i}] \leq \exp\left(t^2c^2(k-1)^2/2\right).$$

Поскольку $\{X_i\}$ — супермартингал, имеем $\mathrm{E}[Y_i|X_{i-1},\ldots,X_0] \leq 0$. Не умаляя общности, предположим, что $\mathrm{E}[Y_i|X_{i-1},\ldots,X_0] = 0$ (в противном случае можно увеличить значения отрицательных величин Y_i, что только увеличит значение $\mathrm{E}[e^{tY_i}]$). Отметим, что из $\mathrm{E}[Y_i] = 0, Y_i \leq c$ и равномерности Y на k значениях следует, что $|Y_i| \leq c(k-1)$. Пусть

$$h(y) = \frac{e^{tc(k-1)} + e^{-tc(k-1)}}{2} + \frac{e^{tc(k-1)} - e^{-tc(k-1)}}{2} \cdot \frac{y}{c(k-1)}$$

есть прямая, проходящая через точки $\left(-c(k-1), e^{-tc(k-1)}\right)$ и $\left(c(k-1), e^{tc(k-1)}\right)$. Из выпуклости e^{tY} и $e^{tY} \leq h(y)$ имеем $|y| \leq c(k-1)$. Значит,

$$\mathrm{E}[e^{tY_i}] \leq \mathrm{E}[h(Y_i)] = h\left(\mathrm{E}[Y_i]\right) = h(0) = \cosh(tc(k-1)) \leq \exp\left(t^2c^2(k-1)^2/2\right),$$

где последнее неравенство $\cosh(x) \leq \exp\left(x^2/2\right)$ для $x > 0$ получается сравнением рядов Тейлора двух функций.

Далее,

$$\mathrm{E}\left[\exp\left(t \cdot \sum_{i=1}^{m} Y_i\right)\right] = \mathrm{E}\left[\exp\left(t \cdot \sum_{i=1}^{m-1} Y_i\right) \cdot \mathrm{E}\left[\exp\left(t \cdot Y_m\right)|X_{m-1},\ldots,X_0\right]\right] \leq$$
$$\mathrm{E}\left[\exp\left(t \cdot \sum_{i=1}^{m-1} Y_i\right)\right] \cdot \exp\left(t^2c^2(k-1)^2/2\right) \leq \exp\left(mt^2c^2(k-1)^2/2\right),$$

из чего при $t = \lambda/mc^2(k-1)^2$ следует $\Pr[X_m - X_0 \geq \lambda] \leq \exp\left(\frac{-\lambda^2}{2mc^2(k-1)^2}\right)$. $\square$

1.6.2 Основная теорема

Определение 62. *Пусть* $\{v_1, \ldots, v_m\}$ *— множество векторов расщепления и пусть* v_i *имеет длину* $2^{t_i} \geq 2$*. Для класса схем над базисом* Ω *(*$\Omega = B_2$ *или* $\Omega = U_2$*), множества подстановок* $\mathcal{S}$ *и меры сложности* μ *будем писать*

$$Splitting(\Omega, \mathcal{S}, \mu) \preceq \{v_1, \ldots, v_m\}$$

для выражения следующего факта: для любой нормализованной схемы $\mathcal{C}$ *над* Ω *можно найти за время* $\mathrm{poly}(\mathtt{gates}(\mathcal{C}))$ *подстановку из* $\mathcal{S}$*, которая либо тривиализирует выходной элемент, либо даёт вектор расщепления относительно* μ *из множества* $\{v_1, \ldots, v_m\}$*; при этом подстановка должна удалять хотя бы один элемент из схемы.*

Теорема 63 ([22])**.** *Если* $Splitting(\Omega, \mathcal{S}, \mu) \preceq \{v_1, \ldots, v_m\}$ *и максимальная длина вектора расщепления есть* 2^k*, то*

1. *Существует алгоритм, решающий задачу #CircuitSAT для схемы* $\mathcal{C}$ *над* Ω *за время* $O^*(\gamma^{\mu(\mathcal{C})})$*, где*

$$\gamma = \max_{i \in [m]} \{\tau(v_i)\}\,.$$

2. *Если* $f \in B_n$ *является* $(\mathcal{S}, n, r)$*-дисперсером, то*

$$\mu(f) \geq \beta_w \cdot (r - k + 1)\,,\ \text{где}\ \beta_w = \min_{i \in [m]} \{\overline{v_i}_{\max}\}\,.$$

3. *Если* $f \in B_n$ *является* $(\mathcal{S}, n, r, \varepsilon)$*-экстрактором, то для любого* $\mu < \beta_a \cdot r$*,*

$$\mu\,(f, \delta) \geq \mu,\ \text{где}\ \beta_a = \min_{i \in [m]} \left\{\overline{v_i}_{\mathrm{avg}}\right\}\ \ \text{и}\ \beta_m = \min_{i \in [m]} \{\overline{v_i}_{\min}\}\,,$$

$$\delta = \varepsilon + \exp\left(\frac{-(r \cdot \beta_a - \mu)^2}{2r(\beta_a - \beta_m)^2(2^k - 1)^2}\right)\,.$$

Доказательство. Мы докажем теорему для частного случая, когда все вектора имеют длину 2 (то есть $k = 1$): $\{v_1, \ldots, v_m\} = \{(a_1, b_1), \ldots, (a_m, b_m)\}$. Это уменьшает количество технических деталей. Доказательство для общего случая проводится аналогично. В этом случае

$$\beta_w = \min_{i\in[m]}\{\max\{a_i, b_i\}\}, \quad \beta_a = \min_{i\in[m]}\left\{\frac{a_i + b_i}{2}\right\}, \quad \beta_m = \min_{i\in[m]}\{\min\{a_i, b_i\}\}.$$

1. Рассмотрим следующий алгоритм расщепления для #CircuitSAT. Мы описываем дерево расщепления, каждая вершина которого содержит текущую схему и множество сделанных подстановок. Корню дерева соответствует пара $(\mathcal{C}, \emptyset)$ — исходная схема и пустое множество подстановок. Вершины, в которых схема тривиализируется, являются листьями. В каждой внутренней вершине алгоритм находит за полиномиальное время (от размера текущей схемы) подстановки $x_i \leftarrow f$ и $x_i \leftarrow f \oplus 1$, гарантируемые условием теоремы. После этого алгоритм производит два рекурсивных вызова для схем, полученных из текущей подстановками $x_i \leftarrow f$ и $x_i \leftarrow f \oplus 1$. Другими словами, к текущей вершине $(\mathcal{C}, S)$ добавляются два ребёнка $(\mathcal{C}|x_i \leftarrow f, S \cup \{x_i \leftarrow f\})$ и $(\mathcal{C}|x_i \leftarrow f \oplus 1, S \cup \{x_i \leftarrow f \oplus 1\})$. Из условия следует, что подстановки $x_i \leftarrow f$ и $x_i \leftarrow f \oplus 1$ или дают (a_i, b_i)-расщепление для некоторого i (то есть уменьшают меру μ хотя бы на a_i в одной ветке и хотя бы на b_i в другой), или тривиализируют схему и добавляют два листа.

   В каждом листе алгоритм вычисляет количество V выполняющих наборов: если текущая схема вычисляет константу ноль, то $V = 0$; в противном случае $V = 2^v$, где v — количество неподставленных переменных в текущей схеме. Общее количество выполняющих наборов исходной схемы равно сумме выполняющих наборов схем в листьях. Поскольку время работы алгоритма в каждой вершине полиномиально, общее время работы

есть $O^*\left(\gamma^{\mu(C)}\right)$, где $\gamma = \max\limits_{i\in[m]}\{\tau(a_i, b_i)\}$.

2. Для пары неотрицательных целых чисел (n, r), таких что $n \geq r \geq 0$, обозначим через $\mathcal{F}_{n,r} \subseteq B_n$ класс функций, не обращающихся в константу после никаких r подстановок из $\mathcal{S}$. Мы покажем, что для любой $f \in \mathcal{F}_{n,r}$ выполняется $\mu(f) \geq \beta_w \cdot (r - k + 1)$.

   Доказательство проведём индукцией по r. Для $r < k$ утверждение очевидно (напомним, что по определению мера сложности любой схемы неотрицательна). Допустим теперь, что $r \geq k$. Рассмотрим подстановки $x_i \leftarrow f$ и $x_i \leftarrow f \oplus 1$, гарантированные условием теоремы. Выберем константу $c \in \mathbb{F}_2$ так, что подстановка $x_i \leftarrow f \oplus c$ уменьшает меру хотя бы на β_w. Рассмотрим функцию g от $n-1$ переменных, полученную из f подстановкой $x_i \leftarrow f \oplus c$. Имеем $\mu(f) \geq \beta_w + \mu(g)$ и по предположению индукции $\mu(g) \geq \beta_w \cdot (r-1-k+1)$. Таким образом, $\mu(f) \geq \beta_w \cdot (r-k+1)$.

3. Рассмотрим схему $\mathcal{C}$, для которой $\mu(\mathcal{C}) \leq \beta_a \cdot r$. Рассмотрим дерево рекурсии из доказательства первого пункта теоремы. Будем спускаться по дереву от корня, в каждой вершине выбирая равновероятно одного из сыновей. Обозначим через δ_i случайную величину, равную уменьшению меры на i-м шаге (i-м уровне дерева рекурсии, где нулевой уровень соответствует корню). Для $i \geq 0$ определим случайную величину

$$X_i = (i+1) \cdot \beta_a - \sum_{j=0}^{i} \delta_j.$$

   Покажем, что последовательность $\{X_i\}$ является супермартингалом:

$$\mathrm{E}[X_i | X_{i-1}, \ldots, X_0] = i \cdot \beta_a - \sum_{j=0}^{i-1} \delta_j + (\beta_a - \mathrm{E}[\delta_i | X_{i-1}, \ldots, X_0]) =$$
$$= X_{i-1} + (\beta_a - \mathrm{E}[\delta_i]) \leq X_{i-1}.$$

Пусть $Y_i = X_i - X_{i-1}$. Тогда Y_i распределена равномерно на не более чем 2^k значениях и $Y_i \leq \beta_a - \beta_m$. Пусть теперь $\lambda = \beta_a \cdot r - \mu(\mathcal{C})$. Тогда по лемме 61:

$$\Pr[X_r - X_0 \geq \lambda] \leq \exp\left(\frac{-\lambda^2}{2r(\beta_a - \beta_m)^2(2^k - 1)^2}\right).$$

Мы хотим ограничить сверху корреляцию между f и функцией, задающейся деревом рекурсии. Все листья дерева рекурсии, имеющие глубину не более r, дают вклад в корреляцию не более ε для экстрактора f (поскольку каждый из таких листьев задаёт $(\mathcal{S}, n, r)$-источник). Посчитаем теперь число листьев на глубине хотя бы r. Всего есть 2^r потенциальных кандидатов, но каждый из них "выживает" до r-го уровня с вероятностью $\Pr[X_r - X_0 \geq \lambda]$. Действительно, если $X_r - X_0 < \lambda$, то $\sum_{j=1}^{r} \delta_j > \mu(C)$, что означает, что функция становится константой до r-го уровня. Значит, есть не более $2^r \cdot \Pr[X_r - X_0 \geq \lambda]$ листьев на глубине хотя бы r. Поскольку каждый лист на уровне r фиксирует хотя бы r входов, он покрывает не более 2^{n-r} точек куба. Таким образом, корреляция ограничена сверху

$$\begin{aligned}\mathrm{Cor}(f, \mathcal{C}) \leq \varepsilon + \exp\left(\frac{-\lambda^2}{2r(\beta_a - \beta_m)^2(2^k - 1)^2}\right) = \\ = \varepsilon + \exp\left(\frac{-(r \cdot \beta_a - \mu(\mathcal{C}))^2}{2r(\beta_a - \beta_m)^2(2^k - 1)^2}\right).\end{aligned}$$

□

1.6.3 Нижняя оценка на схемную сложность в среднем

В данном подразделе мы улучшаем оценки, полученные Р. Ченем и В. Кабанцом для базиса U_2. Их оценки получаются применением теоремы 63 к разбору случаев, проведённому К. Шнорром [47] при доказательстве нижней

оценки $3n-3$. Грубо говоря, К. Шнорр показал, что

$$\mathrm{Splitting}(U_2, \{x_i \leftarrow c\}, \mathtt{gates}) \preceq \{(3,3)\}.$$

Мы улучшаем данный разбор с использованием следующих двух идей. Во-первых, мы используем проекции вместо подстановок, фиксирующих бит. Во-вторых, мы используем более хитрую меру сложности:

$$\mu(\mathcal{C}) = \mathtt{gates}(\mathcal{C}) + \alpha\mathtt{inputs}(\mathcal{C}) - \sigma\mathtt{inputs}_1(\mathcal{C}).$$

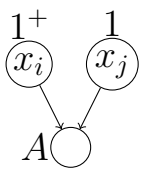
Здесь α, σ — неотрицательные константы, значения которых будут выбраны позже. Как показано в лемме 13, если $\sigma \leq \min\{1, \alpha\}$, то μ действительно является мерой сложности схем: если при удалении проходного элемента увеличивается исходящая степень переменной, то мера всё равно не увеличивается ($\mathtt{gates}$ и $\mathtt{inputs}_1$ уменьшаются при этом на один). Идея использования переменных исходящей степени один была впервые использована У. Цвиком [62].

Лемма 64 ([22])**.** *Для любых* $0 \leq \alpha$ *и* $0 \leq \sigma \leq \min\{1/2, \alpha\}$

$$\begin{aligned}\mathit{Splitting}(U_2, \{x_i \leftarrow c, x_i \leftarrow x_j \oplus c\}, \mathtt{gates} + \alpha\mathtt{inputs} - \sigma\mathtt{inputs}_1) \preceq \\ \preceq \{(\alpha, 2\alpha), (2\alpha, 2\alpha, 2\alpha, 3\alpha), (3+\alpha+\sigma, 3+\alpha), (4+\alpha+\sigma, 2+\alpha)\}.\end{aligned}$$

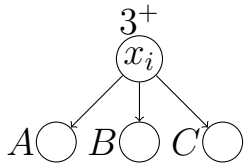
Доказательство. При удалении проходного элемента мера уменьшается хотя бы на $1-\sigma \geq 1/2$. Если же элемент тривиализируется, то мера уменьшается хотя бы на 1.

Случай 1. Есть минимальный элемент A, в который идёт провод из 1-переменной x_j.



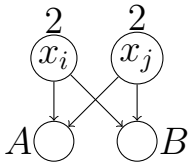
Подставляя в x_i константу, в одной из веток мы тривиализируем A и удаляем зависимость от x_j. Это даёт $(\alpha, 2\alpha)$-расщепление.

Случай 2. Есть 3^+-переменная x_i.



Ни в один из элементов A, B, C не ведёт провод из 1-переменной, иначе мы оказались бы в случае 1. При подстановке константы x_i каждый из элементов A, B, C тривиализируется в одной из веток. Значит, хотя бы в одной из веток мы удалим хотя бы один дополнительный элемент. Это даёт вектор расщепления $(4 + \alpha - \sigma, 3 + \alpha)$, который доминирует вектор $(3 + \alpha + \sigma, 3 + \alpha)$ (поскольку $\sigma \leq 1/2$).

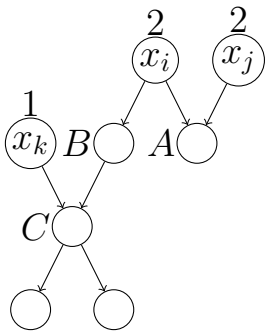
Случай 3. Есть две 2-переменные, которые входят в одни и те же два минимальных элемента.



Пусть A и B вычисляют функции $f_A(x_i, x_j) = (x_i \oplus a_A)(x_j \oplus b_A) \oplus c_A$ и $f_B(x_i, x_j) = (x_i \oplus a_B)(x_j \oplus b_B) \oplus c_B$, соответственно. Если $a_A = a_B$ или $b_A = b_B$, подставим $x_i \leftarrow a_A$ или, соответственно, $x_j \leftarrow b_A$ и тривиализируем оба элемента одновременно. В противном случае $f_B(x_i, x_j) = (x_i \oplus a_A \oplus 1)(x_j \oplus b_A \oplus 1) \oplus c_B$. Нетрудно видеть, что если $x_i \oplus a_A \oplus x_j \oplus b_A = 1$, то обе функции тривиализируются. Следовательно, подстановка $x_i \leftarrow a_A \oplus x_j \oplus b_A \oplus 1$ также тривиализирует A и B. В обоих случаях, таким образом, A и B становятся константами и пропадает зависимость от x_j. Следовательно, получаем вектор расщепления $(\alpha, 2\alpha)$.

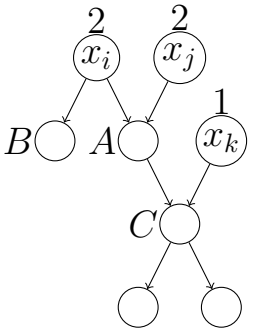
Случай 4. В минимальный элемент A входят две 2-переменные x_i, x_j, переменная x_i также входит в элемент B, а переменная x_j в него не входит. В первых трёх подслучаях мы разберём ситуации, когда при подстановке констант в переменные x_i, x_j приводит к увеличению исходящей степени некоторой 1-переменной x_k.

Случай 4.1. Элемент B является 1-элементом, входящим в 2^+-элемент C, зависящий от 1-переменной x_k.



Подставив константы в x_j и x_k, мы удалим также зависимость от x_i в одной из веток, что даёт вектор расщепления $(2\alpha, 2\alpha, 2\alpha, 3\alpha)$.

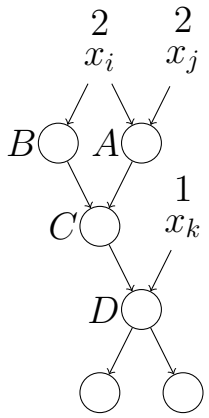
Случай 4.2. Элемент A является 1-элементом, входящим в 2^+-элемент C, зависящим от 1-переменной x_k.



Как и в предыдущем случае, подставляя константы в x_i и x_j, мы удалим зависимость от x_k в одной из веток, что даёт вектор расщепления $(2\alpha, 2\alpha, 2\alpha, 3\alpha)$.

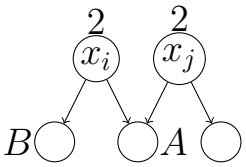
Случай 4.3. Элементы A и B и их общий потомок C являются 1-элементами, а единственный потомок C является 2-элементом, зави-

сящим от 1-переменной x_k.



Подставляя константу в x_k, мы тривиализируем элемент D в одной из ветвей, из-за чего удаляются также элементы A, B, C и пропадает зависимость от x_i. Это даёт вектор расщепления $(\alpha, 2\alpha)$.

Случай 4.4. Ни один из предыдущих трёх случае не применим.



Предыдущие случаи исключают возможность того, что A или B имеют ровно одного потомка, удаление которого уменьшает меру всего на $1 - \sigma$ (то есть увеличивает исходящую степень некой 1-переменной): мы знаем, что один из A и B или является 2^+-элементом и его последователи дают вклад $2(1-\sigma) \geq 1$, или 1-элементом, чей последователь не является 2^+-элементом, зависящим от 1-переменной, и даёт вклад хотя бы 1. Мы также знаем, что если A и B являются 1-элементами с общим последователем, этот последователь не является 2^+-элементом, зависящим от 1-переменной, и поэтому даёт вклад хотя бы 1.

Значит, если подставить в x_i константу, оба элемента A и B тривиализируются в одной из веток, поэтому последователи A и B либо дают вклад хотя бы 1 в обеих ветках, либо дают вклад хотя бы 2 в одной из веток. Более того, x_j становится 1-переменной в ветке, где

A тривиализируется. Таким образом, получаем $(3+\alpha+\sigma, 3+\alpha)$ или $(4+\alpha+\sigma, 2+\alpha)$.

□

Следствие 65 ([22]). *1. Для любого $\epsilon > 0$ найдётся $\delta = \delta(\epsilon) > 0$, так что задача #CircuitSAT для схем над U_2 размера не более $(3.25-\epsilon)n$ может быть решена за время $(2-\delta)^n$.*

2. Пусть $f \in B_n$ есть $\left(n, r(n) = n - \log^{O(1)}(n)\right)$-дисперсер относительно проекций [37]. Тогда

$$\mathtt{gates}_{U_2}(f) \geq 3.5n - \log^{O(1)}(n).$$

3. Пусть $f \in B_n$ есть $\left(n, r(n) = n - \sqrt{n}, \varepsilon(n) = 2^{-n^{\Omega(1)}}\right)$-экстрактор относительно проекций [45]. Тогда

$$\mathtt{gates}_{U_2}(f, \delta) \geq 3.25n - t, \text{ где } \delta = 2^{-n^{\Omega(1)}} + \exp\left(\frac{-(t-10.25\sqrt{n})^2}{190.125(n-\sqrt{n})}\right).$$

В частности, $\mathrm{Cor}(f, \mathcal{C})$ *пренебрежимо мало для любой схемы $\mathcal{C}$ размера* $3.25n - \omega(\sqrt{n \log n})$.

Доказательство. 1. Пусть $\sigma = 1/2$. Заметим, что при достаточно большом α имеем

$$\tau(\alpha, 2\alpha) < \tau(2\alpha, 2\alpha, 2\alpha, 3\alpha) < \tau(3.25+\alpha, 3.25+\alpha) = \\ = 2^{\frac{1}{3.25+\alpha}} < \tau(3.5+\alpha, 3+\alpha) < \tau(4.5+\alpha, 2+\alpha).$$

Пусть $\gamma(\alpha) = \tau(4.5+\alpha, 2+\alpha) - 2^{\frac{1}{3.25+\alpha}}$. По лемме 59 имеем $\gamma(\alpha) = O(1/\alpha^3)$.

Время работы алгоритма не больше

$$\begin{aligned}(\tau(4.5+\alpha,2+\alpha))^{s+\alpha n} &\le \left(2^{\frac{1}{3.25+\alpha}}(1+\gamma(\alpha))\right)^{s+\alpha n} \le 2^{\frac{s+\alpha n}{3.25+\alpha}} 2^{(s+\alpha n)\gamma(\alpha)\log_2 e} \\ &\le 2^{\frac{(3.25-\epsilon)n+\alpha n}{3.25+\alpha}+O(n/\alpha^2)} \le (2-\delta)^n\end{aligned}$$

для некоторого $\delta > 0$, если мы положим $\alpha = c/\epsilon$ для достаточно большого $c > 0$.

2. Лемма 64 гарантирует, что при $\alpha = 7$, $\sigma = 0.5$ всегда найдётся проекция, уменьшающая меру хотя бы на 10.5. Функция f устойчива к любым $r(n)$ проекциями. Поэтому для любой схемы $\mathcal{C}$, вычисляющей f, имеем $\texttt{gates}(\mathcal{C}) + 7n \ge 10.5r(n)$.

3. Рассмотрим схему $\mathcal{C}$ размера не более $3.25n - t$, то есть $\mu(\mathcal{C}) \le (3.25n - t) + \alpha n$. Зафиксируем $\alpha = 7$, $\sigma = 0.5$, тогда

$$\beta_a = \min\{10.5, 15.75, 10.25, 10.25\} = 10.25, \quad \beta_m = \min\{7, 7, 10, 9\} = 7.$$

Воспользуемся теоремой 63 при $k = 2$, $r = n - \sqrt{n}$, $\varepsilon = 2^{-n^{\Omega(1)}}$, $\mu = (3.25n - t + 7n)$:

$$\begin{aligned}\delta = 2^{-n^{\Omega(1)}} + \exp\left(\frac{-(10.25(n-\sqrt{n}) - (10.25n - t))^2}{2(n-\sqrt{n})\cdot 3.25^2 \cdot 3^2}\right) = \\ = 2^{-n^{\Omega(1)}} + \exp\left(\frac{-(t - 10.25\sqrt{n})^2}{190.125(n-\sqrt{n})}\right).\end{aligned}$$

□

1.7 Ограничения метода элиминации элементов

В предыдущих разделах показывается, что метод элиминации элементов может быть использован для доказательства более сильных оценок, если у нас в распоряжении имеется функция, устойчивая относительно достаточно сильных подстановок. Например, аффинные дисперсеры позволяют доказать нижнюю оценку $\left(3+\frac{1}{86}\right)n-o(n)$, квадратичные — оценку $3.11n$. Естественно задаться вопросом: можно ли доказать нелинейные нижние оценки для функций, устойчивых относительно подстановок типа $p=0$, где p — произвольный многочлен степени, скажем, 10 или даже $\log n$? (Отметим в скобках, что на настоящий момент у нас нет явных конструкций даже для функций, где многочлену p разрешается иметь степень всего лишь два.) В данном разделе мы дадим отрицательный ответ на данный вопрос. Показывается, что относительно любых, сколь угодно сильных подстановок, есть схемы, из которых удаляется только константное число элементов. Таким образом, для доказательства нелинейных нижних оценок на схемную сложность будет недостаточно просто построить более сильные дисперсеры.

Лемма 66 ([20])**.** *Для любого целого $m>0$ и любой булевой функции $h\in B_n$ найдётся функция $f\in B_{n(2m+1)}$, такая что*

- *схемная сложность f мало отличается от схемной сложности h:*

$$\mathtt{gates}(h)\le\mathtt{gates}(f)\le\mathtt{gates}(h)+5(2m+1)n\,;$$

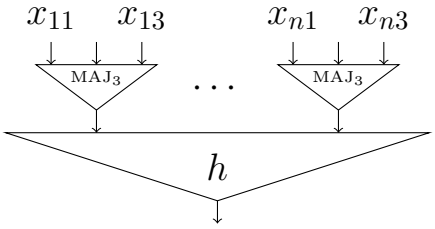
- *любая m-подстановка (то есть подстановка m переменным) ρ удаляет из оптимальной схемы, вычисляющей f, не более $O(m^2)$ элементов:*

$$\mathtt{gates}(f)-\mathtt{gates}(f|_\rho)\le 5(2m+1)m.$$

Доказательство. Мы докажем лемму для $m = 1$, для больших значений m доказательство проводится совершенно аналогично. В качестве функции f рассмотрим функцию $h \diamond \mathrm{MAJ}_3$, полученную заменой каждой входной переменной h на выход функции голосования от трёх переменных:

$$(f \diamond \mathrm{MAJ}_3)(x_{11}, x_{12}, \ldots, x_{n3}) = f(\mathrm{MAJ}_3(x_{11}, x_{12}, x_{13}), \ldots, \mathrm{MAJ}_3(x_{n1}, x_{n2}, x_{n3}))\,,$$

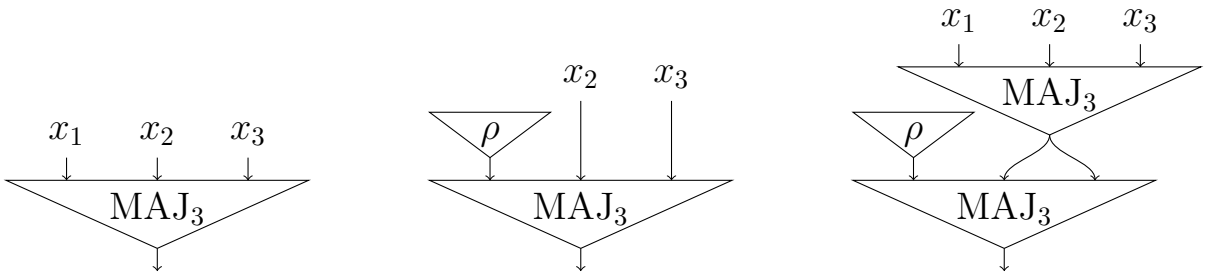
Такая функция может быть посчитана схемой следующим образом:



Рассмотрим схему $\mathcal{C}$ минимального размера, вычисляющую $h \diamond \mathrm{MAJ}_3$. Покажем, что никакая подстановка $x_{ij} \leftarrow \rho$, где ρ — произвольная функция от оставшихся переменных, не может удалить из $\mathcal{C}$ больше пяти элементов:

$$\mathtt{gates}(\mathcal{C}) - \mathtt{gates}(\mathcal{C}|_{x_{ij} \leftarrow \rho}) \leq 5.$$

Для этого мы покажем, что добавлением пяти элементов к схеме, вычисляющей $C|_{x_{ij} \leftarrow \rho}$, можно получить схему, вычисляющую h. Для этого достаточно заметить, что если в схему, вычисляющую $\mathrm{MAJ}_3(x_1, x_2, x_3)$ произведена подстановка $x_1 \leftarrow \rho$, то можно "подавить" эту подстановку, дописав к схеме её копию:



Более формально, допустим, не умаляя общности, что подставлена переменная x_{11}. Рассмотрим схему $\mathcal{C}'$, вычисляющую $f|_{x_{11} \leftarrow \rho}$ и используем выход но-

вого блока $\mathrm{MAJ}_3(x_{11}, x_{12}, x_{13})$ вместо x_{12} и x_{13}. Этим мы подавим эффект подстановки $x_{11} \leftarrow \rho$, получив схему $\mathcal{C}''$, вычисляющую исходную функцию $f = h \diamond \mathrm{MAJ}_3$. Поскольку функцию голосования от трёх входов можно вычислить схемой размера пять, получаем:

$$\mathtt{gates}(\mathcal{C}) \leq \mathtt{gates}(\mathcal{C}'') \leq \mathtt{gates}(\mathcal{C}|_{x_{11}\leftarrow\rho}) + 5\,.$$

Для доказательства леммы для произвольного значения m нужно использовать блоки, вычисляющие функцию голосования от $2m+1$ входов. Тогда в каждом блоке можно подавить каждую подстановку, используя $m+1$ незатронутых битов. □

Данная лемма показывает, что существуют функции, для которых не получится доказать нелинейную нижнюю оценку простым методом элиминации элементов. В работе [20] доказываются более сильные ограничения: рассматриваются также более сложно устроенные меры, строятся примеры схем, из которых m-подстановкой не удастся удалить не только суперквдратичного по m числа элементов, но даже и суперлинейного.

Глава 2

Верхние оценки

2.1 Автоматическое нахождение эффективных схем

В данном разделе мы описываем программу поиска оптимальных схем для функций от малого количества переменных. Факт существования схемы необходимого размера записывается как формула в конъюнктивной нормальной форме (КНФ), после чего для этой формулы запускается специальная программа для решения задачи пропозициональной выполнимости (так называемый SAT-солвер).

Есть несколько причин интересоваться точной схемной сложностью функций от малого числа переменных. Во-первых, как будет показано далее в данном разделе, для некоторых функций эффективные схемы строятся из блоков константного размера. Уменьшение размера такого блока автоматически даёт более сильную оценку для такой функции в общем случае. Во-вторых, как отмечает Р. Вильямс [56], было бы очень полезно иметь энциклопедию оптимальных схем для функций от малого количества переменных. Скажем, знание оптимальных схем для задачи умножения булевых матриц размера $n \times n$ для, скажем, $n = 2, 3, \ldots, 10$ потенциально могло бы помочь нам по-

нять, как устроен эффективный алгоритм для этой задачи. К сожалению, современные компьютеры и программы не позволяют найти оптимальный размер схем для этой задачи даже при $n = 3$. Д. Кнут [29] недавно реализовал сведение, аналогичное описываемому далее в данном разделе, и нашёл точную схемную сложность всех функций от четырёх переменных, а также некоторых функций от пяти переменных.

2.1.1 Сведение

По данной таблице истинности функции $f \in B_{n,m}$ мы хотим найти схему над базисом $\Omega \subseteq B_2$ минимального размера, вычисляющую f. Для заданного числа N мы построим формулу в КНФ, которая выполнима тогда и только тогда, когда для функции f существует схема с N элементами. Формула будет использовать такие переменные: (входные элементы нумеруются от 0 до $n-1$, внутренние — от n до $n + N - 1$):

1. Переменная $t_{ib_1b_2}$ ($n \le i < n + N$, $0 \le b_1 < 2$, $0 \le b_2 < 2$) кодирует выходное значение i-го элемента в случае, когда значение его первого входа равно b_1, а второго — b_2. Таким образом, четыре булевых переменных t_{i00}, t_{i01}, t_{i10}, t_{i11} полностью задают бинарную булеву операцию, вычисляющуюся в элементе. Общее количество переменных этого типа — $O(N)$.

2. Переменная c_{ikj} ($n \le i < n + N - 1$, $0 \le k < 2$, $0 \le j < n + N$) истинна, если и только если на k-й вход i-го элемента идёт провод из j-го элемента. Данные переменные полностью задают ориентированный ациклический граф схемы. Всего переменных: $O(N^2)$.

3. Переменная o_{ij} ($n \le i < n + N$, $0 \le j < m$) истинна, если и только если j-й выход схемы вычисляется в i-м элементе. Данные переменные контролируют, где именно вычисляются выходы функции. Всего переменных: $O(Nm)$.

4. Переменная v_{it} ($0 \le i \le n + N$, $0 \le t < 2^n$) задаёт выходное значение i-го элемента на входном наборе t. Данные переменные используются для того, чтобы гарантировать, что при всех значениях входных переменных значения, вычисляющиеся в элементах, согласованы друг с другом. Всего переменных: $O(2^n N)$.

Следующие ограничения записываются как дизъюнкты формулы.

1. Бинарные булевы операции в элементах принадлежат базису Ω. Всего: $O(N)$ дизъюнктов.

2. Для всех (i, k) ровно одна из переменных c_{ikj} истинна (в k-й вход i-го элемента идёт ровно один провод). Всего: $O(N^3)$ 2-дизъюнктов (то есть дизъюнктов длины два) и $O(N)$ $O(N)$-дизъюнктов.

3. Для всех j ровно одна переменная o_{ij} истинна (j-й выход функции вычисляется ровно в одном элементе). Всего: $O(N^2 m)$ 2-дизъюнктов и $O(m)$ $O(N)$-дизъюнктов.

4. Для всех $0 \le i < n$ и $0 \le t < 2^n$ переменная v_{it} равна соответствующему биту t. Всего: $O(n \cdot 2^n)$ 1-дизъюнктов.

5. Для всех $n \le i < n + N$ и $0 \le t < 2^n$ переменная v_{it} равна значению i-го элемента на входном наборе t. Всего: $O(N^3 \cdot 2^n)$ 6-дизъюнктов. Дизъюнкты в данном случае записываются для всех $n \le i < n + N$, $n \le j_0 < i$, $j_0 < j_1 < i$, $0 \le i_0 < 2$, $0 \le i_1 < 2$, $0 \le r < 2^n$ и выглядят так:

$$\neg c_{i0j_0} \vee \neg c_{i1j_1} \vee \neg(v_{j_0 r} = i_0) \vee \neg(v_{j_1 r} = i_1) \vee (v_{ir} = t_{ii_0 i_1}) .$$

Здесь первые два литерала задают элементы, от которых зависит i-й элемент, следующие два литерала задают значения этих элементов на входе r, а последний литерал проверят, верно ли вычислено значение i-го элемента.

6. Выходы функции вычислены корректно. Всего: $O(N2^n m)$ 2-дизъюнктов. Дизъюнкты данного типа записываются для всех $0 \leq k < m$, $0 \leq r < 2^n$, $n \leq i < n + N$ и выглядят так:

$$\neg o_{ik} \vee (v_{ir} = value_{kr}),$$

где $value_{kr}$ — это значение k-го выхода на входе r, взятое из таблицы истинности.

Мы также записываем следующие дополнительные ограничения.

1. Входы каждого элемента являются элементами с меньшими номерами (то есть элементы схемы топологически упорядочены относительно используемой нумерации).

2. Для каждого элемента первый провод в него идёт из меньшего элемента.

3. Элементы вычисляют невырожденные функции (то есть зависящие от обоих входов).

4. Хотя бы один из выходов вычисляется последним элементом.

2.1.2 Верхняя оценка для $\mathbf{MOD}_n^3$

В большинстве интересных случаев получающиеся формулы в КНФ очень сложны для современных программ, решающих задачу выполнимости. Например, для нахождения схемы из двенадцати элементов программам обычно требуется несколько суток, а доказать, что такой схемы нет (то есть что соответствующая формула невыполнима), у них не получается вообще. Объяснением этому, по-видимому, является банально то, что пространство поиска огромно: число различных схем как функция от количества элементов растёт

очень быстро. По этой причине были также использованы некоторые дополнительные эвристики, которые уменьшают пространство поиска и в случае, когда необходимая схема существует, помогают программе найти её быстрее. Перечислим основные две из них, которые действительно помогли на практике:

1. Исходящая степень каждого элемента не больше двух.
2. В i-й элемент ведёт провод из $(i-1)$-го (то есть существует путь, проходящий через все элементы).

Ниже мы приводим эффективный блок, который был найден компьютером с использованием описанного выше подхода и который помог значительно улучшить верхнюю оценку на схемную сложность функции MOD_n^3.

Теорема 67 ([12])**.** *Для любого целого r*

$$\mathtt{gates}(MOD_n^{3,r}) \leq 3n + \Theta(1).$$

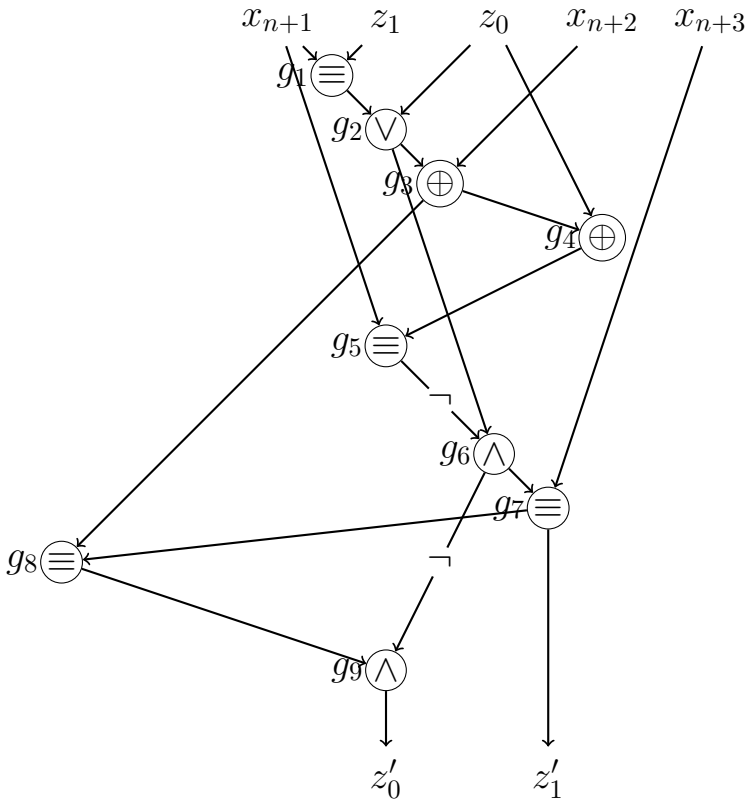
В работе [12] также доказывается оценка $\mathtt{gates}_{U_2}(\mathrm{MOD}_n^{3,r}) \leq 5.5n + \Theta(1)$. Данные две оценки улучшают известные ранее верхние оценки $4.5n + o(n)$ и $7n + o(n)$ для базисов B_2 и U_2, соответственно. Отметим, что для функции $\mathrm{MOD}_n^{2,r}$ в обоих базисах известна её точная сложность (оценка в базисе B_2 доказывается очевидно, в базисе U_2 оценка доказана К. Шнорром [47]):

$$\mathtt{gates}(\mathrm{MOD}_n^{2,r}) = n - 1,$$
$$\mathtt{gates}_{U_2}(\mathrm{MOD}_n^{2,r}) = 3(n-1).$$

Также довольно хорошо изучена сложность функции $\mathrm{MOD}_n^{4,r}$ (нижние оценки доказаны Л. Стокмайером и У. Цвиком [53, 62]):

$$2.5n - \Theta(1) \leq \mathtt{gates}(\mathrm{MOD}_n^{4,r}) \leq 2.5n + \Theta(1),$$
$$4n - \Theta(1) \leq \mathtt{gates}_{U_2}(\mathrm{MOD}_n^{4,r}) \leq 5n + \Theta(1).$$

Доказательство. Блок размера девять и его таблица истинности приведены ниже.



x_{n+1}	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
x_{n+2}	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
x_{n+3}	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
z_0	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
z_1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
g_1	1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
g_2	1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1
g_3	1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0
g_4	1 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 0 0 1 1
g_5	0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 1 1 0 0 1
g_6	1 0 0 0 1 0 0 0 0 1 1 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 1 1 0
g_7	0 1 1 1 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 1 0 0 1 0 1 1 0
g_8	0 0 0 1 1 1 1 0 1 0 1 0 0 1 0 1 0 1 1 1 1 0 0 0 1 0 1 0 0 1 0 1
g_9	0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1
z'_0	0 0 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1
z'_1	0 1 1 1 1 0 0 0 1 0 0 1 0 1 1 0 1 1 1 0 0 0 0 1 1 0 0 1 0 1 1 0

Блок получает на вход значение $\sum_{i=1}^{n} x_i \pmod 3$, закодированное парой битов (z_1, z_2) и три новые переменные. Выходом блока является пара битов (z'_0, z'_1), кодирующих значение $\sum_{i=1}^{n+3} x_i \bmod 3$. Используется следующая кодировка:

$$\sum_{i=1}^{n} x_i \bmod 3 = \begin{cases} 0, & \text{если } (z_0, z_1) = (0, 0), \\ 1, & \text{если } (z_0, z_1) = (0, 1), \\ 2, & \text{если } z_0 = 1. \end{cases}$$

Проверять корректность данного блока вручную, конечно же, утомительно, поэтому ниже мы приводим небольшую программу на языке программирование Python, которая как раз и проверяет, что блок соответствует заявленной спецификации.

```
import itertools

enc = {(0,0): 0, (0,1): 1, (1,0): 2, (1,1): 2}

for (xn1,xn2,xn3,z0,z1) in itertools.product(range(2), repeat=5):
    g1 = 1 - (xn1 ^ z1)
    g2 = g1 | z0
    g3 = g2 ^ xn2
    g4 = g3 ^ z0
    g5 = 1 - (g4 ^ xn1)
    g6 = (1 - g5) & g2
    g7 = 1 - (g6 ^ xn3)
    g8 = 1 - (g3 ^ g7)
    g9 = g8 & (1 - g6)

    y0 = g9
    y1 = g7

    assert((enc[z0,z1]+xn1+xn2+xn3) % 3 == enc[y0,y1])
```

Необходимая верхняя оценка следует непосредственно из существования такого блока. Чтобы получить схему размера $3n + \Theta(1)$, мы соединяем в цепочку $n/3$ копий блока размера девять. Каждый следующий блок прибавляет к текущему остатку по модулю три сумму трёх новых битов. В самом конце получаем остаток по модулю три суммы всех входных n битов, и остаётся просто проверить, равен ли он r. □

2.2 Верхняя оценка для вычисления всех MOD-функций одновременно

Как уже упоминалось ранее, нелинейных нижних оценок на данный момент мы не знаем не только для функций из B_n (то есть булевых предикатов), но и функций из $B_{n,n}$ (булевых функций с n входами и n выходами). Подсчётом можно показать, что схемная сложность n симметрических функций от n переменных есть $\Omega(n^{2-o(1)})$ почти для всех наборов из n симметрических функций. Есть три естественных подкласса симметрических функций:

- $\mathrm{EX}_n^k \in B_n$ равна 1 тогда и только тогда, когда сумма входных n битов равна k;
- $\mathrm{THR}_n^k \in B_n$ равна 1 тогда и только тогда, когда сумма входных n битов хотя бы k;
- $\mathrm{MOD}_n^{m,r} \in B_n$ равна 1 тогда и только тогда, когда сумма входных n битов сравнима с r по модулю m;

Известно (см. следствие 72 далее), что наборы функций

$$\{\mathrm{EX}_n^0, \mathrm{EX}_n^1, \ldots, \mathrm{EX}_n^n\} \text{ и } \{\mathrm{TH}_n^0, \mathrm{TH}_n^1, \ldots, \mathrm{TH}_n^n\}.$$

можно посчитать схемами линейного размера. В данном разделе мы покажем, что все MOD-функции для $1 \leq m \leq n$ также можно посчитать схемами малого (но всё же нелинейного) размера. Для n целых чисел $r_1, r_2, \ldots, r_n$ через $\mathrm{ALLMOD}_n^{r_1,\ldots,r_n}$ обозначим набор

$$\{\mathrm{MOD}_n^{1,r_1}, \mathrm{MOD}_n^{2,r_2}, \ldots, \mathrm{MOD}_n^{n,r_n}\}.$$

Если $r_1 = r_2 = \cdots = r_n = r$, будем писать просто ALLMOD_n^r. Основным результатом данного раздела является следующая оценка.

Теорема 68 ([15])**.**

$$\texttt{gates}(\mathrm{ALLMOD}_n^r) = O(n)\,.$$

В работе [15] также доказывается, что $\texttt{gates}(\mathrm{ALLMOD}_n^{r_1,\ldots,r_n}) = O(n \log\log n)$.

Перед доказательством основного результата мы приводим некоторые вспомогательные факты. В большинстве оценок мы опускаем целые части (то есть пишем, например, $\log_2 n$ вместо $\lceil \log_2 n \rceil$). Это не влияет на асимптотическое поведение оцениваемых величин. Через X мы обозначаем $\sum_{i=1}^n x_i$. Мы также будем опускать n, когда оно ясно из контекста. Через $\log x$ и $\ln x$ будем обозначать логарифм x по основаниям 2 и e, соответственно.

Лемма 69. *1. Гармонический ряд ([3], теорема 2.5.3):*

$$H_n = \sum_{1 \le k \le n} \frac{1}{k} = \ln n + \Theta(1)\,.$$

2. Гармонический ряд простых ([3], теорема 8.8.5):

$$P_n = \sum_{p \le n, p \in \mathbb{P}} \frac{1}{p} = \ln\ln n + \Theta(1).$$

3. Асимптотический закон распределения простых чисел ([3], теорема 8.8.1): число простых чисел, не превосходящих n, есть

$$\pi(n) = \Theta\left(\frac{n}{\ln n}\right)\,.$$

2.2.1 Кодировки

Для заданного $m \in \mathbb{N}$ есть два естественных способа кодировать остаток r по модулю m бинарными строками:

- $bin(r, m) = (b_0, b_1, \ldots, b_{\log m})$ — $\log m$-битовое двоичное представление r:

$$\sum_{0 \le i \le \log m} b_i 2^i = r\,.$$

- $ex(r, m) = (e_0, e_1, \ldots, e_{m-1})$ — m-битовая строка, такая что $e_r = 1$ и $e_k = 0$ для всех $0 \le k \le m-1$, $k \ne r$.

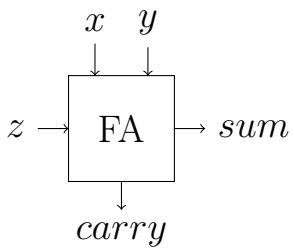
Конечно же, первое представление более компактно. Второе представление, однако, позволяет очень просто проверить, задаёт ли данная битовая строка конкретный остаток (достаточно прочитать соответствующий бит). Например, чтобы проверить, задаёт ли данная строка $b \in \mathbb{F}_2^4$ остаток $6 \bmod 10$, нужно вычислить бит $(\neg b_0) \wedge b_1 \wedge b_2 \wedge (\neg b_3)$, в то время как чтобы проверить, задаёт ли данная 10-битовая строка e тот же остаток, достаточно прочесть бит e_6.

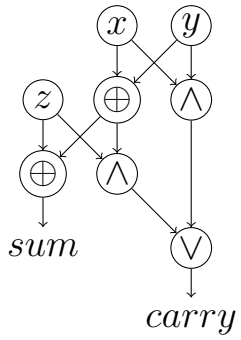
2.2.2 Вспомогательные блоки

Следующие две леммы хорошо известны и могут быть найдены, например, в [55, раздел 3.4]. Их доказательства сравнительно просты, и мы приводим их основные идеи для полноты изложения.

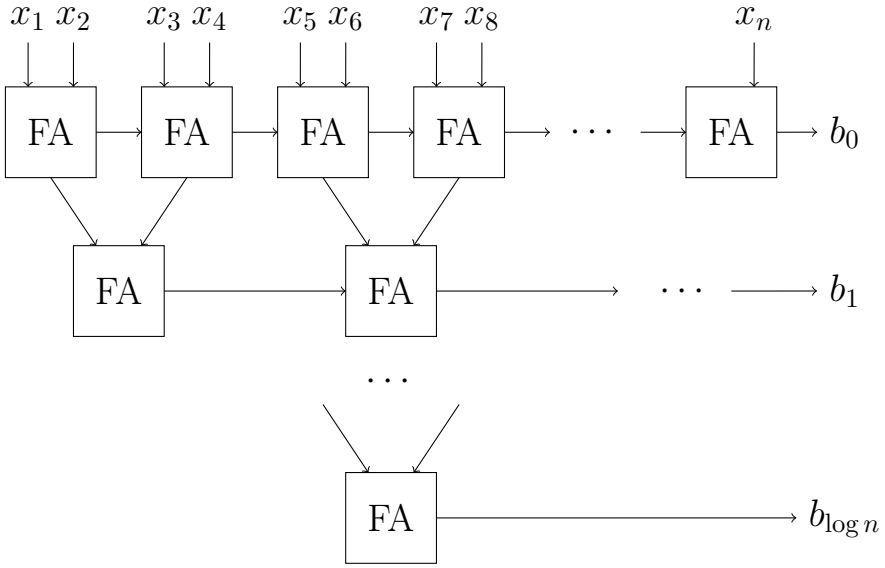
Лемма 70. *Существует схема размера $O(n)$, принимающая на вход n битов $x_1, \ldots, x_n$ и выдающая $bin(X, n+1)$, то есть двоичное представление X.*

Доказательство. Необходимая схема строится из блоков под названием Full Adder (FA). Такой блок вычисляет функцию из $B_{3,2}$, которая на входе (x, y, z) выдаёт два бита $(carry, sum)$, таких что $x + y + z = 2 \cdot carry + sum$ (то есть младший и старший бит суммы). Этот блок можно реализовать пятью элементами.





Используя $\log n$ слоёв таких блоков, можно вычислить X.



□

Лемма 71. *Существует схема размера $O(m)$, которая для данных $\log m$ входных битов $(b_0, \ldots, b_{\log m}) = bin(r, m)$ выдаёт $(e_0, \ldots, e_m) = ex(r, m+1)$.*

Доказательство. Заметим, что каждый e_j — это просто конъюнкция $d_0 \wedge \cdots \wedge d_{\log m}$, где каждый d_i — это либо b_i, либо $\neg b_j$. Таким образом, нам нужно вычислить все такие $m+1$ конъюнкций, используя $O(m)$ элементов. Наивная схема для такой задачи имеет размер $O(m \log m)$, так как для вычисления каждой отдельной конъюнкции понадобится $\log m$ элементов. Для более быстрого вычисления мы сначала вычислим множество C_1 всех возможных конъюнкций первой половины битов b (то есть битов $b_0, \ldots, b_{(\log m)/2}$) и множество C_2 всех возможных конъюнкций второй половины. Это потребует

$O(\sqrt{m}\log m)$ элементов. После этого любая конъюнкция всех битов b вычисляется при помощи одного элемента из двух соответствующих конъюнкций из C_1 и C_2. $\square$

Используя только что доказанные две леммы, получаем такое следствие.

Следствие 72. *Существует схема размера $O(n)$, которая по n входным битам $x_1, x_2, \ldots, x_n$ выдаёт $(e_0, \ldots, e_n) = ex(X, n+1)$.*

Следствие 72 позволяет вычислить любую симметрическую функцию f схемой линейного размера. Действительно, пусть $(e_0, \ldots, e_n) = ex(X, n+1)$ и пусть $(v_0, \ldots, v_n)$ — вектор значений функции f. Тогда

$$f(x_1, \ldots, x_n) = \bigvee_{0 \le i \le n} (e_i \wedge v_i)\,.$$

Ещё одно непосредственное следствие заключается в том, что $\{\mathrm{EX}_n^k\}_{0\le k\le n}$ и $\{\mathrm{TH}_n^k\}_{0\le k\le n}$ могут быть посчитаны схемами линейного размера: EX_n^k — это просто e_k, а

$$\mathrm{TH}_n^{k-1} = \mathrm{TH}_n^k \vee \mathrm{EX}_n^{k-1}\,.$$

Лемма 73. *Для любого m, $bin(X \bmod m, m)$ может быть вычислена из $bin(X, n+1)$, используя $O(\log n \log m)$ элементов.*

Доказательство. По данному двоичному представлению $b = (b_0, \ldots, b_{\log n})$ суммы n входных битов нам нужно вычислить двоичное представление остатка этой суммы по модулю m. Пусть $l = \log n$. Тогда $X = \sum_{0\le i\le l} b_i 2^i$.

Построим $(l+1)$ блоков $B_0, \ldots, B_l$. Блок B_i берёт на вход бит b_i и выдаёт $\log m$ битов, кодирующих остаток по модулю m числа $b_i 2^i$ (то есть $bin(b_i 2^i \bmod m, m)$). Это очень простой блок: его выходом является либо 0, либо $(2^i \bmod m)$. Ясно, что $O(\log m)$ элементов достаточно для построения такого блока.

После этого нужно сложить полученные $(l+1)$ остатков. Для этого нам понадобится блок A, который получает на вход два остатка a и b по модулю m (то есть $2\log m$ входных битов) и выдаёт $(a+b) \bmod m$ ($\log m$ битов). Блок A имеет размер $O(\log m)$, поскольку нужно просто вычислить $(a+b)$ и вычесть m, если $a+b \geq m$. Чтобы посчитать сумму $(l+1)$ остатка по модулю m, нужны l копий блока A. Размер полученной схемы — $O(\log m \log n)$. □

Лемма 74. *$MOD_n^{m,r}$ можно вычислить из $ex(X, n+1)$, используя n/m элементов.*

Доказательство. Это непосредственно следует из

$$\mathrm{MOD}_{m,r}^n = \bigvee_{m|(q-r)} e_q \,.$$

□

Лемма 75. *$MOD_n^{m,r}$-функции для всех $1 \leq m \leq \sqrt{n}$ и $0 \leq r < m$ можно вычислить за линейное числи элементов, i.e.,*

$$\texttt{gates}\left(\{MOD_n^{m,r}\}_{0\leq r<m\leq\sqrt{n}}\right) = O(n)\,.$$

Доказательство. По лемме 73, $bin(X \bmod m, m)$ можно вычислить за $c\log n \log m$ элементов. Поэтому для вычисления $bin(X \bmod m, m)$ для всех $m \leq \sqrt{n}$ потребуется

$$\sum_{1\leq m\leq\sqrt{n}} c\log n\log m \leq \sqrt{n}\cdot c\log n\log\sqrt{n} = o(n)$$

элементов. После этого вычислим

$$ex(X \bmod m, m) = \{\mathrm{MOD}_n^{m,r}\}_{0\leq r<m}$$

из $bin(X \bmod m, m)$, используя $O(m)$ элементов, по лемме 71. Суммируя по

всем $m \le \sqrt{n}$, получаем линейную верхнюю оценку:

$$\sum_{1 \le m \le \sqrt{n}} O(m) = O(n)\,.$$

□

2.2.3 Доказательство оценки

В данном разделе мы предполагаем, что $bin(X, n+1), ex(X, n+1)$ и $\left\{\mathrm{MOD}^n_{m,r}\right\}_{0 \le r < m \le \sqrt{n}}$ уже посчитаны (по леммам 70, 71 и 75 это требует не более $O(n)$ элементов).

Мы доказываем верхнюю оценку для случая, когда все r_i равны. В этом случае задача сводится к $\mathrm{MOD}^{m,r}_n$ для всех m, являющихся степенью простого. Мы вычисляем MOD^{m,r_m}_n для $m = p^k$ в два шага: для $1 < p \le \sqrt{n}$ и для $\sqrt{n} < p \le n$

Доказательство теоремы 68. Вычислим $\mathrm{MOD}^{m,r}_n$ для $m = p^k$, где $p \le \sqrt{n}$ и $p \in \mathbb{P}$. Есть не более $\sqrt{n}\log n$ таких m (поскольку $k \le \log n$), поэтому по лемме 73 для этого шага потребуется не более $O(\sqrt{n}\log^3 n)$ элементов.

После этого вычислим $\mathrm{MOD}^{m,r}_n$ для всех оставшихся простых m (заметим, что m не может быть нетривиальной степенью простого $\sqrt{n} < p \le n$). По лемме 74 для этого потребуется

$$\sum_{\sqrt{n} < p \le n,\, p \in \mathbb{P}} \frac{n}{p} = n(P_n - P_{\sqrt{n}}) = \qquad \text{(лемма 69.2)}$$

$$= n(\ln\ln n - \ln\ln\sqrt{n} + \Theta(1)) = O(n)$$

элементов.

К настоящему моменту мы вычислили $\mathrm{MOD}^{m,r}_n$ для всех m, являющихся степенью простого. Все целые числа $2 \le m \le n$, не являющиеся степенью

простого, представимы как произведение st двух взаимно простых чисел s and t. Тогда

$$\mathrm{MOD}_n^{m,r} = \mathrm{MOD}_n^{s,r} \wedge \mathrm{MOD}_n^{t,r} .$$

Таким образом, каждое m, не являющееся степенью простого, требует одного дополнительного элемента. Следовательно, размер построенной схемы есть $O(n)$. □

Заключение

Основными результатами данной работы являются новые нижние оценки на схемную сложность булевых функций, а также новые методы для получения нижних оценок. Как видно из работы, даже для небольшого улучшения понадобилось много новых идей, а также технических деталей. В работе также показывается, что текущие методы вряд ли позволят доказать нелинейные нижние оценки на схемную сложность. Для получения таких оценок понадобятся, по-видимому, принципиально новые идеи и техника.

Литература

[1] *Alon N., Spencer J.* The Probabilistic Method. John Wiley, 1992.

[2] *Amano K., Tarui J.* A well-mixed function with circuit complexity $5n$: Tightness of the Lachish-Raz-type bounds // Theor. Comput. Sci. 2011. Vol. 412, N. 18. P. 1646–1651.

[3] *Bach E., Shallit J. O.* Algorithmic Number Theory. Massachusetts: MIT Press, 1996. Vol. I: Efficient Algroithms of *Foundations of Computing Series*.

[4] *Ben-Sasson E., Gabizon A.* Extractors for Polynomials Sources over Constant-Size Fields of Small Characteristic // Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings / Ed. by A. Gupta, K. Jansen, J. D. P. Rolim, R. A. Servedio. Vol. 7408 of *Lecture Notes in Computer Science*. Springer, 2012. P. 399–410.

[5] *Ben-Sasson E., Kopparty S.* Affine Dispersers from Subspace Polynomials // SIAM J. Comput. 2012. Vol. 41, N. 4. P. 880–914.

[6] *Ben-Sasson E., Viola E.* Short PCPs with Projection Queries // Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I / Ed. by J. Esparza, P. Fraigniaud, T. Husfeldt, E. Koutsoupias. Vol. 8572 of *Lecture Notes in Computer Science*. Springer, 2014. P. 163–173.

[7] *Blum N.* A Boolean Function Requiring $3n$ Network Size // Theor. Comput. Sci. 1984. Vol. 28. P. 337–345.

[8] *Chen R., Kabanets V.* Correlation Bounds and #SAT Algorithms for Small Linear-Size Circuits // Computing and Combinatorics - 21st International Conference, COCOON 2015, Beijing, China, August 4-6, 2015, Proceedings / Ed. by D. Xu, D. Du, D. Du. Vol. 9198 of *Lecture Notes in Computer Science*. Springer, 2015. P. 211–222.

[9] *Chen R., Kabanets V., Kolokolova A., Shaltiel R., Zuckerman D.* Mining Circuit Lower Bound Proofs for Meta-Algorithms // Computational Complexity. 2015. Vol. 24, N. 2. P. 333–392.

[10] *Cohen G., Shinkar I.* The Complexity of DNF of Parities // Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016 / Ed. by M. Sudan. ACM, 2016. P. 47–58.

[11] *Cohen G., Tal A.* Two Structural Results for Low Degree Polynomials and Applications // CoRR. 2014. Vol. abs/1404.0654.

[12] *Demenkov E., Kojevnikov A., Kulikov A. S., Yaroslavtsev G.* New upper bounds on the Boolean circuit complexity of symmetric functions // Inf. Process. Lett. 2010. Vol. 110, N. 7. P. 264–267.

[13] *Demenkov E., Kulikov A. S.* An Elementary Proof of a $3n - o(n)$ Lower Bound on the Circuit Complexity of Affine Dispersers // Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings / Ed. by F. Murlak, P. Sankowski. Vol. 6907 of *Lecture Notes in Computer Science*. Springer, 2011. P. 256–265.

[14] *Demenkov E., Kulikov A. S., Melanich O., Mihajlin I.* New Lower Bounds on Circuit Size of Multi-output Functions // Theory Comput. Syst. 2015. Vol. 56, N. 4. P. 630–642.

[15] *Demenkov E., Kulikov A. S., Mihajlin I., Morizumi H.* Computing All MOD-Functions Simultaneously // Computer Science - Theory and Applications - 7th International Computer Science Symposium in Russia, CSR 2012, Nizhny Novgorod, Russia, July 3-7, 2012. Proceedings / Ed. by E. A. Hirsch, J. Karhumäki, A. Lepistö, M. Prilutskii. Vol. 7353 of *Lecture Notes in Computer Science*. Springer, 2012. P. 81–88.

[16] *Dodis Y.* Exposure-resilient cryptography: Ph.D. thesis / Massachusetts Institute of Technology. 2000.

[17] *Dvir Z.* Extractors for varieties // Computational Complexity. 2012. Vol. 21, N. 4. P. 515–572.

[18] *Dvir Z., Gabizon A., Wigderson A.* Extractors And Rank Extractors For Polynomial Sources // Computational Complexity. 2009. Vol. 18, N. 1. P. 1–58.

[19] *Find M., Golovnev A., Hirsch E. A., Kulikov A. S.* A Better-than-3n Lower Bound for the Circuit Complexity of an Explicit Function // 57th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2016, New Brunswick, NJ, USA, 2016. Proceedings. 2016. P. 88–97.

[20] *Golovnev A., Hirsch E. A., Knop A., Kulikov A. S.* On the Limits of Gate Elimination // 41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland / Ed. by P. Faliszewski, A. Muscholl, R. Niedermeier. Vol. 58 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. P. 46:1–46:13.

[21] *Golovnev A., Kulikov A. S.* Weighted Gate Elimination: Boolean Dispersers for Quadratic Varieties Imply Improved Circuit Lower Bounds // Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016 / Ed. by M. Sudan. ACM, 2016. P. 405–411.

[22] *Golovnev A., Kulikov A. S., Smal A. V., Tamaki S.* Circuit Size Lower Bounds and #SAT Upper Bounds Through a General Framework // 41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland / Ed. by P. Faliszewski, A. Muscholl, R. Niedermeier. Vol. 58 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. P. 45:1–45:16.

[23] *Håstad J.* Almost Optimal Lower Bounds for Small Depth Circuits // Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA / Ed. by J. Hartmanis. ACM, 1986. P. 6–20.

[24] *Håstad J.* The Shrinkage Exponent of de Morgan Formulas is 2 // SIAM J. Comput. 1998. Vol. 27, N. 1. P. 48–64.

[25] *Heinz E.* Beiträge zur Störungstheorie der Spektralzerleung // Mathematische Annalen. 1951. Vol. 123, N. 1. P. 415–438.

[26] *Impagliazzo R., Nisan N.* The Effect of Random Restrictions on Formula Size // Random Struct. Algorithms. 1993. Vol. 4, N. 2. P. 121–134.

[27] *Iwama K., Morizumi H.* An Explicit Lower Bound of $5n - o(n)$ for Boolean Circuits // Mathematical Foundations of Computer Science 2002, 27th International Symposium, MFCS 2002, Warsaw, Poland, August 26-30, 2002, Proceedings / Ed. by K. Diks, W. Rytter. Vol. 2420 of *Lecture Notes in Computer Science*. Springer, 2002. P. 353–364.

[28] *Jukna S.* Boolean Function Complexity - Advances and Frontiers. Springer, 2012. Vol. 27 of *Algorithms and combinatorics*.

[29] *Knuth D. E.* The Art of Computer Programming. Addison–Wesley, 2015. Vol. 4, pre-fascicle 6a. Section 7.2.2.2. Satisfiability. Draft available at `http://www-cs-faculty.stanford.edu/~uno/fasc6a.ps.gz`.

[30] *Kojevnikov A., Kulikov A. S.* Circuit Complexity and Multiplicative Complexity of Boolean Functions // Programs, Proofs, Processes, 6th Conference on Computability in Europe, CiE 2010 / Ed. by F. Ferreira, B. Löwe, E. Mayordomo, L. M. Gomes. Vol. 6158 of *Lecture Notes in Computer Science*. Springer, 2010. P. 239–245.

[31] *Kojevnikov A., Kulikov A. S., Yaroslavtsev G.* Finding Efficient Circuits Using SAT-Solvers // Theory and Applications of Satisfiability Testing - SAT 2009, 12th International Conference, SAT 2009, Swansea, UK, June 30 - July 3, 2009. Proceedings / Ed. by O. Kullmann. Vol. 5584 of *Lecture Notes in Computer Science*. Springer, 2009. P. 32–44.

[32] *Komargodski I., Raz R., Tal A.* Improved Average-Case Lower Bounds for DeMorgan Formula Size // 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA. IEEE Computer Society, 2013. P. 588–597.

[33] *Kulikov A. S., Melanich O., Mihajlin I.* A $5n - o(n)$ Lower Bound on the Circuit Size over U_2 of a Linear Boolean Function // How the World Computes - Turing Centenary Conference and 8th Conference on Computability in Europe, CiE 2012, Cambridge, UK, June 18-23, 2012. Proceedings / Ed. by S. B. Cooper, A. Dawar, B. Löwe. Vol. 7318 of *Lecture Notes in Computer Science*. Springer, 2012. P. 432–439.

[34] *Kullmann O.* Fundaments of Branching Heuristics // Handbook of Satisfiability / Ed. by A. Biere, M. Heule, H. van Maaren, T. Walsh. IOS Press, 2009. Vol. 185 of *Frontiers in Artificial Intelligence and Applications*. P. 205–244.

[35] *Lachish O., Raz R.* Explicit lower bound of $4.5n-o(n)$ for Boolean circuits // Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece / Ed. by J. S. Vitter, P. G. Spirakis, M. Yannakakis. ACM, 2001. P. 399–408.

[36] *Li X.* A New Approach to Affine Extractors and Dispersers // Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, June 8-10, 2011. IEEE Computer Society, 2011. P. 137–147.

[37] *Li X.* Extractors for Affine Sources with Polylogarithmic Entropy // Electronic Colloquium on Computational Complexity (ECCC). 2015. Vol. 22. P. 121.

[38] *Maurey B.* Espaces de Banach: Construction de suites symetriques // C.R. Acad. Sci. Paris Ser. A-B. 1979. Vol. 288. P. 679–681.

[39] *Muller D. E.* Complexity in Electronic Switching Circuits // IRE Trans. on Electronic Computers. 1956. Vol. 5. P. 15–17.

[40] 41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland / Ed. by P. Faliszewski, A. Muscholl, R. Niedermeier. Vol. 58 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[41] Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016 / Ed. by M. Sudan. ACM, 2016.

[42] *Paterson M., Zwick U.* Shrinkage of de Morgan Formulae under Restriction // Random Struct. Algorithms. 1993. Vol. 4, N. 2. P. 135–150.

[43] *Paturi R., Saks M. E., Zane F.* Exponential lower bounds for depth three Boolean circuits // Computational Complexity. 2000. Vol. 9, N. 1. P. 1–15.

[44] *Paul W. J.* A $2.5n$-Lower Bound on the Combinational Complexity of Boolean Functions // SIAM J. Comput. 1977. Vol. 6, N. 3. P. 427–443.

[45] *Rao A.* Extractors for Low-Weight Affine Sources // Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009. IEEE Computer Society, 2009. P. 95–101.

[46] *Santhanam R.* Fighting Perebor: New and Improved Algorithms for Formula and QBF Satisfiability // 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA. IEEE Computer Society, 2010. P. 183–192.

[47] *Schnorr C.* Zwei lineare untere Schranken für die Komplexität Boolescher Funktionen // Computing. 1974. Vol. 13, N. 2. P. 155–171.

[48] *Schnorr C.* The Combinational Complexity of Equivalence // Theor. Comput. Sci. 1976. Vol. 1, N. 4. P. 289–295.

[49] *Schnorr C.* The Multiplicative Complexity of Boolean Functions // Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 6th International Conference, AAECC-6, Rome, Italy, July 4-8, 1988, Proceedings / Ed. by T. Mora. Vol. 357 of *Lecture Notes in Computer Science*. Springer, 1988. P. 45–58.

[50] *Seto K., Tamaki S.* A satisfiability algorithm and average-case hardness for formulas over the full binary basis // Computational Complexity. 2013. Vol. 22, N. 2. P. 245–274.

[51] *Shaltiel R.* Dispersers for Affine Sources with Sub-polynomial Entropy // IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011 / Ed. by R. Ostrovsky. IEEE Computer Society, 2011. P. 247–256.

[52] *Shannon C. E.* The synthesis of two-terminal switching circuits // Bell Systems Technical Journal. 1949. Vol. 28. P. 59–98.

[53] *Stockmeyer L. J.* On the Combinational Complexity of Certain Symmetric Boolean Functions // Mathematical Systems Theory. 1977. Vol. 10. P. 323–336.

[54] *Tal A.* Shrinkage of De Morgan Formulae by Spectral Techniques // Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on / IEEE. 2014. P. 551–560.

[55] *Wegener I.* The complexity of Boolean functions. Wiley-Teubner, 1987.

[56] *Williams R.* Applying practice to theory // SIGACT News. 2008. Vol. 39, N. 4. P. 37–52.

[57] *Williams R.* Improving exhaustive search implies superpolynomial lower bounds // SIAM J. Comput. 2013. Vol. 42, N. 3. P. 1218–1244. Extended abstract appeared in Proc. STOC-2010.

[58] *Williams R.* Nonuniform ACC circuit lower bounds // JACM. 2014. Vol. 61, N. 1. Extended abstract appears in Proc. CCC-2011.

[59] *Williams R.* CS 354: Topics in Circuit Complexity, Spring 2016. Lecture notes. 2016. `https://web.stanford.edu/~rrwill/cs354.html`.

[60] *Yao A. C.* Separating the Polynomial-Time Hierarchy by Oracles (Preliminary Version) // FOCS. IEEE Computer Society, 1985. P. 1–10.

[61] *Yehudayoff A.* Affine extractors over prime fields // Combinatorica. 2011. Vol. 31, N. 2. P. 245–256.

[62] *Zwick U.* A $4n$ Lower Bound on the Combinational Complexity of Certain Symmetric Boolean Functions over the Basis of Unate Dyadic Boolean Functions // SIAM J. Comput. 1991. Vol. 20, N. 3. P. 499–505.

[63] *Андреев А. Е.* Об одном методе получения более чем квадратичных эффективных нижних оценок сложности схем // Вестник Московского государственного университета. 1987. № 1. С. 70–73.

[64] *Клосс Б. М., Малышев В. А.* Оценки сложности некоторых классов функций // Вестник МГУ, серия 1, Мат.-Мех. 1965. Т. 4. С. 44–51.

[65] *Лупанов О. Б.* О синтезе контактных схем // Доклады Академии наук СССР. 1958. Т. 119, № 1. С. 23–26.

[66] *Нечипорук Э. И.* Об одной булевской функции // Доклады Академии наук СССР. 1966. Т. 169, № 4. С. 765–766.

[67] *Нигматуллин, Р. Г.* Сложность булевых функций. Наука, 1991.

[68] *Нурк С. И.* Верхняя оценка для задачи Circuit SAT: Tech. Rep. 10: Санкт-Петербургское отделение Математического института им. В. А. Стеклова РАН, 2009.

[69] *Разборов А. А.* Нижние оценки монотонной сложности некоторых булевых функций // Доклады Академии наук СССР. 1985. Т. 281, № 4. С. 798–801.

[70] *Савинов С. А.* Верхние оценки для задачи выполнимости булевых схем. Магистерская диссертация, Академический университет РАН. 2014.

[71] *Субботовская Б. А.* О реализации линейных функций формулами в базисе $\vee, \wedge, -$ // Доклады Академии наук СССР. 1961. Т. 136, № 3. С. 553–555.

[72] *Храпченко В. М.* О сложности реализации линейной функции в классе П-схем // Математические заметки. 1971. Т. 9, № 1. С. 35–40.