# Реализация монотонных булевых функций монотонными информационными графами

Ю. С. Шуткин

Рассматривается задача реализации булевых функций с помощью монотонных информационных графов, то есть графов, базовое множество которых состоит из переменных без отрицаний. Получены оценки сложности для различных подклассов этой задачи.

## Введение

Информационно-графовый подход к решению различных задач хорошо зарекомендовал себя на примере задач поиска в базах данных [1]. Однако этот подход является более универсальным и годится для описания достаточно широкого класса алгоритмов, которые не обязательно связаны с информационным поиском. Главным преимуществом такого подхода является возможность оценивать временную сложность описываемого алгоритма. Структура построенного графа задает эту сложность одновременно с самим алгоритмом.

Для реализации булевых функций придумано и исследовано множество различных управляющих систем, таких как контактные схемы и схемы из функциональных элементов. Физические характеристики таких схем, такие как объем и глубина, хорошо изучены, и для них получены достаточно точные оценки. В частности, Лупановым [3] получена асимптотика объемной сложности контактных схем и схем из функциональных элементов.

Однако, в связи с появлением компьютеров и ростом их вычислительных способностей в настоящее время разработка управляющих систем тесно связана с их программным моделированием. Поэтому вместе с физическими характеристиками управляющих систем необходимо учитывать сложность их моделирования на компьютере.

Данная задача рассматривается на примере контактных схем, а именно, исследуется так называемая временная сложность контактных схем, которая характеризует время работы алгоритма, вычисляющего реализуемую булеву функцию.

Для оценки такой сложности применяется информационно-графовый подход, который позволяет формализовать понятие временной сложности и получить ее оценки в терминах элементарных операций.

В работе [2] представлены результаты для реализации различных булевых функций с помощью информационных графов с простым базовым множеством, то есть с базовым множеством, состоящим из селекторов и их отрицаний. Так же там приведено небольшое отступление по поводу возможности варьирования базового множества и классов функций.

В данной работе исследуются классы монотонных функций и монотонное базовое множество, состоящее только из селекторов (или просто переменных). Информационные графы с таким базовым множеством называются монотонными.

Получены следующие оценки сложности реализации монотонных булевых функций с помощью монотонных информационных графов.

Для функции Шеннона сложности реализации монотонных булевых функций монотонными информационными графами получена асимптотика логарифма. В классе древовидных информационных графов (информационных деревьев) эту оценку удалось улучшить до порядка. В обоих случаях сложность имеет экспоненциальный рост от числа переменных функций.

Для почти всех булевых функций получена асимптотика логарифма сложности реализации монотонными информационными графами, при этом сложность также имеет экспоненциальный рост, однако, показатель экспоненты отличается от показателя в оценке функции Шеннона, что не является типичным для оценок сложности булевых функций (например для объемной сложности эти показатели совпадают). В классе информационных деревьев получен порядок сложности реализации почти всех монотонных булевых функций, сложность также имеет экспоненциальный рост, но показатель экспоненты отличается от показателя в оценке функции Шеннона. Это говорит о том, что худший случай в задаче реализации булевых функций монотонными информационными графами — скорее исключение из правил, нежели общая тенденция для большинства булевых функций.

Также в работе отдельно рассмотрен класс симметрических пороговых монотонных функций. Для функций из этого класса получен порядок сложности реализации в классе монотонных информационных деревьев, имеющий экспоненциальный рост, и полиномиальная оценка сложности реализации в классе монотонных информационных графов.

## 1. Постановка задачи и формулировка результатов

Напомним базовые понятия, с которыми мы работаем.

Пусть $F = \{f_1, f_2, \dots\}$ — некоторое (возможно бесконечное) множество булевых функций (предикатов).

*Информационным графом* с базовым множеством $F$ будем называть сеть (связный ориентированный граф с несколькими выделенными вершинами — полюсами), в которой один из полюсов выделен и помечен как *корень* информационного графа, еще несколько полюсов выделены и называются *концевыми* вершинами информационного графа, а ребрам сети приписаны функции из множества $F$. Корень информационного графа обозначается $v_0$ и называется также *начальной* вершиной графа. Множество концевых вершин $w_j$ обозначается через $W$.

В данной работе внимание будет уделено базовому множеству $F^M$, состоящему из всевозможных селекторов $g_i(x_{i_1}, \dots, x_{i_s})$. Для удобства мы будем отождествлять селекторы и переменные, которые они реализуют. Тогда множество $F^M$ может быть записано как $\{x_1, x_2, \dots\}$, подразумевая, что $x_i$ представляет множество всех селекторов, равных $x_i$.

Когда это понятно из контекста, вместо термина *информационный граф* будет использоваться термин *граф*.

Пусть фиксирован некоторый информационный граф $G$.

Считается, что ребро графа $G$ с приписанным ему $n$-местным предикатом $f$ проводит набор $\alpha = (a_1, a_2, \ldots, a_n) \in \{0,1\}^n$ тогда и только тогда, когда $f(\alpha) = 1$. Например, функция $f = x_i$ равна единице тогда и только тогда, когда $a_i = 1$. Считаем, что если местность предиката, приписанного ребру, не совпадает с длинной набора, то проводимость ребра на таком наборе не имеет смысла, или не определена.

Путем в информационном графе мы будем называть ориентированную цепь.

Говорим, что набор $\alpha \in \{0,1\}^n$ проходит из вершины $v_1$ в вершину $v_2$ (или вершина $v_2$ достижима из $v_1$ на наборе $\alpha$), если в информационном графе существует путь из $v_1$ в $v_2$ такой, что все ребра этого пути проводят набор $\alpha$. Через $\theta_{v_1}(\alpha)$ будем обозначать множество вершин, достижимых из $v_1$ на наборе $\alpha$. Если вершина $v_1$ не указана явно, будем в качестве нее подразумевать корень информационного графа. Так, говорим, что набор проходит в вершину $v$, если он проходит из корня $v_0$ в $v$.

Путь в информационном графе называем *проводимым*, если существует набор такой, что все ребра пути проводят этот набор. Иначе путь называется *не проводимым* (с нулевой проводимостью).

Вершину $v_2$ называем *сильно достижимой* из вершины $v_1$, если существует проводимый путь, соединяющий вершины $v_1$ и $v_2$. Вершина $v$ называется сильно достижимой, если она сильно достижима из корня $v_0$.

Если информационный граф $G$ является ориентированным деревом (то есть он имеет структуру дерева, и все ребра ориентированы от корня к листьям), а также все концевые вершины графа $G$ являются листьями этого дерева, то информационный граф $G$ будем называть *информационным деревом.* Длину минимального проводимого пути из начальной вершины дерева в концевую вершину будем называть *минимальной высотой* этого дерева. Длину максимального проводимого пути из начальной вершины дерева в концевую вершину будем называть *максимальной высотой* этого дерева. Если минимальная и максимальная высота дерева совпадают, то будем называть их просто *высотой.*

Говорим, что информационный граф $G$ с базовым множеством $F = \{f_1, f_2, \dots\}$ реализует булеву функцию (б. ф.) $f(x_1, x_2, \dots, x_n)$, если все предикаты, приписанные ребрам графа $G$ зависят от переменных $x_1, \dots, x_n$, и любой набор $\alpha \in \{0,1\}^n$, на котором функция принимает значение 1, проходит из начальной вершины графа $G$ в одну из концевых вершин, то есть $\theta_{v_0}(\alpha) \cap W \neq \varnothing$, и любой набор $\beta \in \{0,1\}^n$, на котором функция принимает значение 0, не проходит ни в одну из них, то есть $\theta_{v_0}(\beta) \cap W = \varnothing$. Множество графов с базовым множеством $F$, реализующих функцию $f$, обозначаем через $U(f, F)$.

Базовое множество $F$ называется ИГ-полным в классе булевых функций $A$, если для любой булевой функции $f \in A$ существует информационный граф с базовым множеством $F$, реализующий функцию $f$.

Заметим, что множество $F^M$ не является ИГ-полным во всем $P_2$, так как информационными графами с таким базовым множеством можно реализовать только монотонные булевы функции (мон. б. ф.). Однако, оно является ИГ-полным в классе монотонных функций. По этой причине множество $F^M$ называется монотонным базовым множеством. Информационный граф с базовым множеством $F_M$ будем называть *монотонным* информационным графом.

Количество предикатов, вычисленных на наборе $\alpha$ в графе $G$ обозначим через $L(G, \alpha)$. Считается оно следующим образом. Помечаются все вершины графа $G$, в которые проходит набор $\alpha$. Считаем, что в вершине $v$ вычисляются те предикаты, которые приписаны ребрам, выходящим из этой вершины. Общее количество вычисленных предикатов на наборе $\alpha$ равно сумме по всем помеченным вершинам вычисленных в них предикатов, то есть

$$L(G, \alpha) = \sum_{v \in \theta_{v_0}(\alpha)} \psi(v),$$

где $\psi(v)$ — количество ребер, выходящих из $v$ (полустепень исхода вершины $v$).

Количество предикатов, вычисленных на наборе $\alpha$, называется сложностью графа на этом наборе.

Пусть на множестве наборов введено вероятностное пространство $(\{0,1\}^n, \sigma, P)$, где $\sigma$ — множество всех подмножеств булева куба $\{0,1\}^n$, а $P$ — вероятностная мера на этом множестве. Сложностью информационного графа назовем величину

$$L(G) = \sum_{\alpha \in \{0,1\}^n} L(G,\alpha)P(\alpha) = E_\alpha\big(L(G,\alpha)\big),$$

где $P(\alpha)$ — вероятность набора $\alpha$ в данном вероятностном пространстве.

Вероятностью ребра назовем вероятность прохождения набора в вершину, из которой это ребро выходит. Это есть вероятность, с которой данное ребро вычисляется. Эту величину мы также будем называть сложностью ребра.

Далее будем по умолчанию считать, что распределение наборов равномерное, то есть вероятности появления всех наборов равны.

Сложностью б. ф. $f$ называется нижняя грань сложности информационных графов, реализующих эту функцию.

$$L(f,F) = \inf_{G \in U(f,F)} L(G).$$

В [2] было показано, что нижняя грань в определении сложности достигается, то есть в $U(f,F)$ существует граф, сложность которого в точности равна сложности функции $f$. Такие графы будем называть *оптимальными* для функции $f$, и говорить, что они реализуют функцию $f$ оптимально.

Для произвольного класса булевых функций $A$ через $A^{(n)}$ будем обозначать класс функций $f \in A$, зависящих от $n$ переменных. Так, $P_2^{(n)}$ будет обозначать класс всех $n$-местных булевых функций, а $M^{(n)}$ — класс всех монотонных булевых функций, зависящих от $n$ переменных.

Далее, пока не сказано обратное, в качестве базового множества будем подразумевать $F^M$ и второй аргумент в обозначении сложности опускать (вместо $L(f,F^M)$ будем писать $L(f)$).

Функцией Шеннона сложности реализации монотонных булевых функций $n$ переменных монотонными информационными графами назовем максимальную сложность функций из $M^{(n)}$ (обозначаем $L^{Sh}(n)$).

$$L^{Sh}(n) = \max_{f \in M^{(n)}} L(f).$$

Определим также сложность реализации булевых функций с помощью информационных деревьев и функцию Шеннона для древовидных информационных графов.

Сложность реализации б. ф. $f$ деревом определим как

$$L_D(f, F) = \inf_{D \in D(f,F)} L(D),$$

где $D(f, F)$ — множество информационных деревьев с базовым множеством $F$, реализующих функцию $f$.

Как и в случае графов, в [2] было показано, что для любой булевой функции существует оптимальное информационное дерево.

Также в рамках данной работы будем по умолчанию подразумевать $F^M$ в качестве базового множества и опускать второй аргумент в записи сложности (вместо $L_D(f, F^M)$ будем писать $L_D(f)$).

Функцию Шеннона в классе деревьев определим как

$$L_D^{Sh}(n) = \max_{f \in M^{(n)}} L_D(f).$$

Итак, целью данной работы является синтез минимального по сложности монотонного информационного графа (или дерева) для произвольной монотонной булевой функции, а также получение нижней оценки сложности таких графов.

Автором работы получены следующие результаты.

Получены оценки функции Шеннона в классе информационных деревьев и информационных графов.

**Теорема 1.** *При $n \to \infty$ выполнено*

$$L_D^{Sh}(n) \asymp \frac{1}{\sqrt{n}} \left(\frac{3}{2}\right)^n,$$

$$\log_2 L^{Sh}(n) \sim (\log_2 3 - 1)n.$$

Найдены оценки сложности реализации почти всех функций $f(x_1, \ldots, x_n) \in M$ с помощью монотонных информационных деревьев и графов.

**Теорема 2.** *Для почти всех* $f(x_1, \ldots, x_n) \in M$ *при* $n \to \infty$ *выполнено*

$$L_D(f) \asymp \frac{1}{\sqrt{n}} \cdot 2^{\frac{n}{2}},$$
$$\log_2 L(f) \sim \frac{n}{2}.$$

Заметим, что оценки для функции Шеннона и для почти всех монотонных булевых функций отличаются.

Также в работе отдельно рассмотрен класс симметрических пороговых монотонных функций $ST$, определяемый следующим образом: $ST^{(n)} = \{f_h : f_h(a_1, \ldots, a_n) = 1 \Leftrightarrow \sum\limits_{i=1}^{n} a_i \geqslant h,\ h = 0, 1, \ldots, n\}$. Для функций из этого класса получен порядок сложности в классе информационных деревьев и неэкспоненциальная оценка сложности в классе информационных графов.

**Теорема 3.** *Для любой* $f_h(x_1, \ldots, x_n) \in ST$ *при* $n \to \infty$ *выполнено*

$$L_D(f_h) \asymp 2^{-h} \cdot C_n^h,$$
$$n - h \leqslant L(f_h) \leqslant h(n - h + 1)^2.$$

## 2. Нижние оценки сложности

Начнем с получения нижних оценок сложности реализации монотонных булевых функций с помощью информационных деревьев.

Докажем сначала несколько вспомогательных лемм.

Пусть дана некоторая б. ф. $f \in M$. Через $N_i(f)$ будем обозначать число наборов, на которых функция равна $i$.

*Минимальным весом* функции $f$ назовем величину $\min\limits_{\alpha \in N_1(f)} |\alpha|$, то есть минимальный вес набора, на котором функция принимает значение 1.

Говорим, что набор $\alpha = (a_1, \ldots, a_n) \in \{0,1\}^n$ *предшествует или равен* (не больше) набора $\beta = (b_1, \ldots, b_n)$, если $a_i \leqslant b_i,\ i = 1, \ldots, n$ (обозначаем $\alpha \preceq \beta$). Говорим, что набор $\alpha$ строго меньше набора $\beta$, если $\alpha \preceq \beta$ и $\alpha \neq \beta$ (обозначаем $\prec$). Аналогично определяем отношения $\succeq$ и $\succ$.

Нижней единицей функции $f$ называем набор $\alpha$ такой, что $f(\alpha) = 1$, а для любого набора $\beta \prec \alpha$ выполнено $f(\beta) = 0$.

**Лемма 1.** *Если монотонное информационное дерево $D$ реализует мон. б. ф. $f$, то минимальная высота дерева $D$ не меньше минимального веса функции $f$.*

**Доказательство.** Обозначим минимальный вес функции $f$ через $h$.

Предположим, что минимальная высота дерева $D$ строго меньше $h$. Это означает, что существует проводимый путь $\gamma$ из начальной вершины дерева $D$ в концевую, длина которого меньше $h$. Рассмотрим множество предикатов на ребрах этого пути. Обозначим это множество через $S = \{x_{i_1}, \ldots, x_{i_s}\}$. Тогда $|S| < h$.

Рассмотрим набор $\alpha = (a_1, a_2, \ldots, a_n)$ такой, что $a_i = 1$ тогда и только тогда, когда предикат $x_i \in S$. Очевидно, что каждый предикат из множества $S$ равен 1 на этом наборе. Следовательно, каждое ребро пути $\gamma$ проводит набор $\alpha$. Но это означает, что набор $\alpha$ проходит из начальной вершины в концевую, следовательно, значение функции на нем должно быть равно 1. Это противоречит тому, что минимальный вес функции $f$ равен $h$, так как $|\alpha| < h$.

Тем самым, лемма доказана.

**Лемма 2.** *Если монотонное информационное дерево $D$ реализует мон. б. ф. $f$, то в любую концевую вершину дерева $D$ проходит не более одной нижней единицы функции $f$.*

**Доказательство.** Предположим, что это не так. То есть в информационном дереве $D$ существует концевая вершина, в которую проходят два различных набора $\alpha_1$ и $\alpha_2$, оба являющиеся нижними единицами функции $f$.

Заметим, что так как в дереве в каждую концевую вершину ведет только один путь, то множество наборов, проходящих по этому пути является некоторой гранью булева куба $\{0,1\}^n$. Следовательно, если эта грань содержит оба набора $\alpha_1$ и $\alpha_2$, то она также содержит их максимального общего предшественника, то есть максимальный набор $\beta$, для которого выполнено $\beta \preceq \alpha_1$ и $\beta \preceq \alpha_2$. Следовательно, на этом наборе функция $f$ тоже должна принимать значение 1, что невозможно, так как $\alpha_1$ и $\alpha_2$ — нижние единицы, и на всех меньших наборах функция принимает значение 0.

Получили противоречие, лемма доказана.

**Лемма 3.** *Если информационный граф $G$ с базовым множеством $F$ оптимально реализует б. ф. $f$, то в графе $G$ нет не сильно достижимых вершин.*

**Доказательство.** Предположим, что в графе $G$ есть вершина, в которую не проходит ни одного набора. Тогда мы можем удалить эту вершину вместе со всеми ребрами, исходящими из этой вершины, а также ребро, ведущее в эту вершину. Очевидно, что функциональность графа при этом не изменится, так как для каждого набора множество вершин, в которые он проходит, не изменится. Однако, когда мы проделаем такую процедуру для всех не сильно достижимых вершин, мы обязательно удалим хотя бы одно ребро, которое имело сложность, отличную от 0 (то ребро, которое вело из сильно достижимой вершины в не сильно достижимую). Следовательно, сложность полученного графа будет строго меньше сложности исходного графа, хотя он по прежнему будет реализовывать функцию $f$. Это противоречит оптимальности исходного графа $G$. Следовательно, не сильно достижимых вершин в графе $G$ быть не может.

Лемма доказана.

**Следствие 1.** *Сложность любого ребра оптимального информационного графа строго больше 0.*

**Лемма 4.** *Если информационное дерево $D$ с базовым множеством $F$ оптимально реализует б. ф. $f$, то все предикаты, приписанные ребрам произвольного пути в дереве $D$, различны.*

**Доказательство.** Рассмотрим произвольный путь в дереве $D$. Пусть в нем есть два ребра $e_1$ и $e_2$, которым приписаны одинаковые предикаты $f_i$. Пусть ребро $e_2$ (то, которое дальше от начала пути) соединяет вершины $v_1$ и $v_2$. Сделаем следующее преобразование: стянем ребро $e_2$ в точку, то есть поменяем начала всех ребер, выходящих из $v_2$ (если такие есть), с $v_2$ на $v_1$, и удалим ребро $e_2$.

Покажем, что функциональность дерева при этом не изменится. Это напрямую следует из того, что функция проводимости любого пути из начальной вершины не изменится, так как если какой-либо набор проходил через удаленное ребро $e_2$, то он проходил и через ребро $e_1$ в силу древовидности графа. Следовательно, предикат $f_i$, приписанный удаленному ребру, останется в конъюнкции функции проводимости и после удаления ребра $e_2$.

Покажем теперь, что сложность полученного дерева строго меньше исходного. Это следует из того, что сложность всех ребер останется прежней, так как все наборы, проходящие в вершину $v_1$, прошли в нее через ребро $e_1$, а следовательно, предикат $f_i$ на всех таких наборах равен 1, и удаленное ребро $e_2$ ни одного набора не отфильтровывало. Ну и наконец, сложность удаленного ребра строго больше 0 по следствию 1. Получаем противоречие с оптимальностью исходного дерева.

Лемма доказана.

Рассмотрим симметрические пороговые монотонные функции $f_h(x_1, \ldots, x_n)$ следующего вида:

$$f_h(a_1, \ldots, a_n) = 1 \Leftrightarrow \sum_{i=1}^{n} a_i \geqslant h, \; h \leqslant n,$$

то есть функция $f_h$ принимает значение 1 на тех наборах, число единиц в которых не меньше $h$.

Через $ST$ обозначим класс всех функций такого вида, то есть $ST^{(n)} = \{f_h(x_1, \ldots, x_n), \; h = 0, 1, \ldots, n\}$. Для каждого $n$ класс $ST^{(n)}$ содержит ровно $n+1$ функцию (с точностью до переименования переменных).

**Лемма 5.** *Для любой б. ф.* $f_h(x_1, \ldots, x_n) \in ST$ *выполнено*

$$L_D(f_h) \geqslant 2^{-h+1} \cdot C_n^h.$$

**Доказательство.** Рассмотрим произвольное информационное дерево $D$, оптимально реализующее функцию $f_h$. То, что такое дерево существует, показано в [2].

По леммам 1 и 2 имеем, что минимальная высота дерева не меньше $h$, а также в каждую концевую вершину дерева $D$ проходит не более одного набора, являющегося нижней единицей функции $f_h$.

Однако, каждая нижняя единица функции $f_h$ в какую-то концевую вершину дерева проходит. Следовательно, можно заключить, что каждой нижней единице функции $f_h$ можно поставить в соответствие по крайней мере одну концевую вершину, в которую она проходит, так, что разным нижним единицам будет сопоставлены разные вершины. Рассмотрим произвольную такую вершину $w$ и путь из начальной вершины в $w$. По лемме 4 на этом пути всем ребрам приписаны различные предикаты, следовательно, сложность первого ребра пути равна 1, а каждого следующего ровно в раза меньше. Также очевидно, что этот путь не может содержать более $h$ различных предикат, так как тогда по нему бы не проходила нижняя единица функции $f_h$. Поэтому длина этого пути в точности равна $h$. Сложность последнего ребра, тем самым, равна $2^{-h+1}$. Последнее же ребро однозначно соответствует концевой вершине, в которую оно ведет.

Зная, что число нижних единиц пороговой функции $f_h$ равно $C_n^h$, получаем оценку сложности дерева $D$:

$$L(D) \geqslant 2^{-h+1} \cdot C_n^h.$$

Отсюда следует утверждение леммы.

Рассмотрим поведение этой оценки как функции от $h$.

$$\frac{2^{-h+1} \cdot C_n^h}{2^{-h+2} \cdot C_n^{h-1}} = \frac{(h-1)!(n-h+1)!}{2(h)!(n-h)!} = \frac{n-h+1}{2h},$$

то есть $\frac{2^{-h+1} \cdot C_n^h}{2^{-h+2} \cdot C_n^{h-1}} \geqslant 1$ при $h \leqslant \frac{n+1}{3}$.

Следовательно, максимум сложности достигается при $h = [\frac{n+1}{3}]$, при этом она равна $2^{-[\frac{n+1}{3}]+1} \cdot C_n^{[\frac{n+1}{3}]}$, что асимптотически равно $\frac{3}{\sqrt{\pi n}} \left(\frac{3}{2}\right)^n$ при $n \to \infty$.

Получаем, что справедливо следующее утверждение:

**Следствие 2.** *При $n \to \infty$ выполнено*

$$L_D^{Sh}(n) \gtrsim \frac{3}{\sqrt{\pi n}} \left(\frac{3}{2}\right)^n.$$

Перейдем теперь к случаю информационных графов.

Докажем некоторые утверждения о структуре оптимальных информационных графов для монотонных функций.

Вопрос существования оптимального графа рассмотрен в [2], и установлено, что он всегда существует для равномерного распределения наборов.

**Лемма 6.** *Если монотонный информационный граф $G$ оптимально реализует мон. б. ф. $f$, тогда для любого ребра $h$ графа $G$ существует путь из начальной вершины в одну из концевых, содержащий ребро $h$ и проводящий хотя бы один набор, являющийся нижней единицей функции $f$.*

**Доказательство.** Для каждой нижней единицы $\alpha$ функции $f$ рассмотрим путь, по которому она проходит в концевую вершину. Такой путь обязательно существует, так как граф реализует функцию $f$, но может быть не единственным. Фиксируем один из таких путей (произвольный) и обозначим его через $p_\alpha$. Тогда очевидно, что подграф $G'$ графа $G$, образованный объединением всех выделенных путей $p_\alpha$, также будет реализовывать функцию $f$, так как любой набор, на котором функция принимает значение 1, больше либо равен некоторой нижней единице и, следовательно, пройдет по одному из этих путей.

Если граф $G'$ совпадает с $G$, то утверждение выполнено, так как граф $G'$ составлен в точности из путей, проводящих нижние единицы. Если же $G \neq G'$, то граф $G'$ есть некоторый собственный подграф графа $G$, и по следствию 1 его сложность строго меньше сложности графа $G$, что противоречит оптимальности $G$.

Лемма доказана.

$i$-м слоем булева куба $\{0,1\}^n$ будем называть множество наборов веса $i$, то есть наборов с $i$ единицами.

**Лемма 7.** *Если монотонный информационный граф $G$ оптимально реализует мон. б. ф. $f$, у которой все нижние единицы расположены на слоях не выше $i$-го, то сложность любого ребра графа $G$ не меньше $2^{-i}$.*

**Доказательство.** Рассмотрим произвольное ребро $h$ и вершину $v$, из которой оно выходит. По лемме 6 существует путь из корня в концевую вершину, проходящий через ребро $h$ и проводящий набор $\alpha$, являющийся нижней единицей функции $f$. Очевидно, что набор $\alpha$ проходит в вершину $v$. Тогда в нее проходят и все наборы, не меньшие $\alpha$, а их не меньше $2^{n-i}$, так как в $\alpha$ не больше $i$ единиц (по условию). Отсюда следует, что сложность любого ребра, выходящего из вершины $v$ (в том числе и $h$), не меньше $2^{-n} \cdot 2^{n-i} = 2^{-i}$.

Лемма доказана.

**Замечание.** Для деревьев в утверждении леммы 7 можно заменить $2^{-i}$ на $2^{-i+1}$ (см. доказательство леммы 5).

**Лемма 8.** *При $n \to \infty$ выполнено*

$$L^{Sh}(n) \gtrsim \frac{1}{2(\log_2 3 - \frac{2}{3})n} \cdot \frac{3}{\sqrt{\pi n}} \cdot \left(\frac{3}{2}\right)^n .$$

**Доказательство.** Будем использовать мощностные соображения для получения нижней оценки.

Пусть $N(n,k)$ — число неизоморфных двуполюсных контактных схем, которые имеют не более $k$ ребер, не содержат изолированных вершин, кроме, может быть, полюсов, и содержат контакты лишь из множества $\{x_1, \ldots, x_n\}$. В [3] показано, что их число может быть оценено сверху как

$$N(n,k) \leqslant (c_1 \cdot nk)^k,$$

где $c_1$ — некоторая константа.

Обозначим через $N_o(n,k)$ аналогичное число ориентированных контактных схем.

При переходе к ориентированным контактным схемам надо учесть две возможные ориентации каждого ребра. Тем самым число таких схем возрастет не более чем в $2^k$ раз. Следовательно, оценка числа ориентированных контактных схем (а вместе с ними и числа монотонных информационных графов, так как в структурном смысле это одно и то же) останется верна, но для другой константы $c_2 = 2c_1$.

$$N_o(n,k) \leqslant (c_2 \cdot nk)^k. \tag{1}$$

Рассмотрим множество всех булевых монотонных функций от $n$ переменных, нижние единицы которых расположены не выше $i$-го слоя булева куба. Будем обозначать такое множество через $M_i^{(n)}$. Очевидно, что $|M_i^{(n)}| \geqslant 2^{C_n^i}$, так как множество $M_i^{(n)}$ содержит все функции, у которых нижние единицы в точности на $i$-м слое.

Рассмотрим некоторую функцию $l_i(n)$, и пусть $\alpha_i(n)$ — доля функций $f(x_1, \dots, x_n)$ из $M_i^{(n)}$, сложность которых не больше $l_i(n)$.

Каждой такой функции сопоставим оптимальный граф, который ее реализует. Сложность каждого такого графа будет не более $l_i(n)$, причем разным функциям будут соответствовать неизоморфные графы. По лемме 7 имеем, что количество ребер в каждом таком графе будет не больше $2^i \cdot l_i(n)$.

Имея оценку количества неизоморфных ориентированных графов с $k$ ребрами (1), выпишем оценку числа функций из $M_i^{(n)}$ сложности не больше $l_i(n)$:

$$\alpha_i(n) \cdot |M_i^{(n)}| \leqslant (c_2 \cdot n \cdot 2^i l_i(n))^{2^i l_i(n)}.$$

Пусть теперь $l_i(n) = (1-\varepsilon)\frac{2^{-i}C_n^i}{n \cdot d(i,n)}$, где $d(i,n)$ — некоторый поправочный коэффициент, такой, что $d(i,n) \asymp 1$, $n \to \infty$, который мы определим далее. Оценим $\log_2 \alpha_i(n)$.

$$\log_2 \alpha_i(n) \leqslant (1-\varepsilon)\frac{C_n^i}{n \cdot d(i,n)} \log_2\Big(c_2(1-\varepsilon)\frac{C_n^i}{d(i,n)}\Big) - C_n^i.$$

Используя асимптотику $C_n^i$, имеем

$$C_n^i = \frac{n!}{i!(n-i!)} \sim \frac{\sqrt{2\pi n}(\frac{n}{e})^n}{\sqrt{2\pi i}(\frac{i}{e})^i \cdot \sqrt{2\pi(n-i)}(\frac{n-i}{e})^{n-i}} =$$
$$= \sqrt{\frac{n}{2\pi i(n-i)}} \cdot \frac{n^n}{i^i(n-i)^{n-i}} = \sqrt{\frac{1}{2\pi n I(1-I)}} \cdot \frac{1}{I^{In}(1-I)^{n(1-I)}} =$$
$$= \frac{1}{\sqrt{2\pi n}} \cdot \frac{1}{\sqrt{I(1-I)}} \cdot \frac{1}{(I^I(1-I)^{1-I})^n}, \ n \to \infty,$$

где $I = \frac{i}{n}$.

Рассмотрим $i(n)$ такое, что $i(n) \asymp n$. Тогда $I \asymp 1$.

Тогда

$$\log_2 C_n^i = -\frac{1}{2}\log_2 n - (I\log_2 I + (1-I)\log_2(1-I))n + o(\log_2 n),\ n \to \infty.$$

Подставляя это в оценку для $\alpha_i(n)$, получаем

$$\log_2 \alpha_i(n) \leqslant (1-\varepsilon)\frac{C_n^i(-(I\log_2 I + (1-I)\log_2(1-I)) + O(\frac{\log_2 n}{n}))}{d(i,n)} - C_n^i.$$

Теперь, если в качестве $d(i,n)$ мы возьмем $-(I\log_2 I + (1-I)\log_2(1-I))$, то получим, что при $i \asymp n$ выполнено $d(i,n) \asymp 1$, следовательно, верно неравенство

$$\log_2 \alpha_i(n) \leqslant -\varepsilon C_n^i + o(C_n^i),\ n \to \infty. \tag{2}$$

Это означает, что доля функций из $M_i^{(n)}$ со сложностью меньшей $l_i(n)$ стремится к нулю при $n \to \infty$. В частности, это дает оценку для функции Шеннона: при $i = [\frac{n+1}{3}]$ получаем, что для любого $\varepsilon > 0$ найдется достаточно большое $n$, что в $M_i^{(n)}$ есть функция $f$ со сложностью, большей $(1-\varepsilon)\frac{2^{-i}C_n^i}{n\cdot d(i,n)} \sim (1-\varepsilon)\frac{1}{2(\log_2 3-\frac{2}{3})n}\cdot\frac{3}{\sqrt{\pi n}}\cdot(\frac{3}{2})^n,\ n\to\infty$. Это дает следующую асимптотическую оценку функции Шеннона:

$$L^{Sh}(n) \gtrsim \frac{1}{2(\log_2 3 - \frac{2}{3})n}\cdot\frac{3}{\sqrt{\pi n}}\cdot\left(\frac{3}{2}\right)^n,\ n \to \infty. \tag{3}$$

Тем самым, лемма доказана.

## 3. Реализация пороговых функций

Теперь установим верхнюю оценку сложности реализации пороговых функций, рассмотренных в разделе 2.

### 3.1. Случай графов

Предложим метод реализации пороговой функций с помощью информационных графов.

**Лемма 9.** *Для любой* $f_h(x_1, \dots, x_n) \in ST$ *выполнено*

$$n - h \leqslant L(f_h) \leqslant h(n - h + 1)^2.$$

**Доказательство.** Пусть дана пороговая функция $f_h(x_1, x_2, \dots, x_n) \in ST$. Предложим сначала конструкцию, в которой кроме переменных некоторым ребрам будут приписаны функции, тождественно равные 1, то есть просто проводящие контакты. Далее из полученной конструкции путем замены ребер с 1 на некоторые пучки ребер получим граф в нашем базисе.

Итак, фиксируем начальную и концевую вершину графа и построим $n - h + 1$ непересекающихся цепочек длины $h$, соединяющих эти вершины. Обозначим их $c_1, c_2, \dots, c_{n-h+1}$. Каждая цепочка $c_i$ состоит из $h$ ребер и может быть представлена как $v_{i0}e_{i1}v_{i1}e_{i2}\dots e_{ih}v_{ih}$, где $e_{ij}$ — суть ребра этой цепочки, а $v_{ij}$ — вершины цепочки. Причем $v_{i0}$ и $v_{ih}$ — соответственно корень и концевая вершина графа, следовательно, они совпадают для всех $i$.

Теперь зададим способ, которым переменные будут приписываться ребрам каждой цепочки.

Каждому ребру $e_{ij}$ припишем переменную $x_{i+j-1}$. Так, первому ребру первой цепочки будет приписана переменная $x_{1-1+1} = x_1$, а последнему ребру последней цепочки будет приписана переменная $x_{n-h+1+h-1} = x_n$.

Далее, из каждой промежуточной вершины $v_{ij}$, $i = 1, \dots, n - h$, $j = 1, \dots, h - 1$ выпустим ребро с пометкой 1 в соответствующую ей вершину $v_{i+1j}$ в следующей параллельной цепочке.

Покажем, что построенный граф реализует функцию $f_h$. Действительно, любой путь из корня в концевую вершину состоит из $h$ ребер, которым приписаны переменные $x_i$, и ребер с отметкой 1, причем все переменные на ребрах различны, так как их индексы обязательно возрастают с перемещением от корня к концевой вершине и от первой цепочки к последней. С другой стороны, любой набор переменных длины $h$ мы можем упорядочить по возрастанию индексов и последовательно выделить цепочку, по которой он проходит. Таким образом, все нижние единицы функции, реализуемой построенным графом, имеют в точности $h$ единичных компонент, следовательно это и есть функция $f_h(x_1, \dots, x_n)$.

Теперь сделаем замену ребер, которым приписана 1 на пучки ребер, которыми приписаны некоторые переменные, так, чтобы функциональность графа не изменилась. Сначала заменим каждое такое ребро пучком из $n$ ребер, которым приписаны все переменные $x_1, \dots, x_n$. Функциональность графа при этом, очевидно, не изменится. Однако мы сейчас покажем, что такая замена избыточна, и можно обойтись меньшим числом ребер.

Понятно, что если мы будем выкидывать ребра из каких-то пучков, то проводимость графа только уменьшится, следовательно, нужно следить лишь, чтобы все необходимые наборы проходили по этим пучкам. Рассмотрим, например, пучок ребер, ведущий из вершины $v_{11}$ в вершину $v_{21}$. Пусть какой-то набор проходит по этому пучку из начальной вершины в концевую. Но тогда он обязательно должен был бы пройти и по ребру $e_{11}$, чтобы попасть в вершину $v_{11}$. Следовательно, первая его компонента обязательно равна 1. Отсюда следует, что если мы выкинем из этого пучка все ребра за исключением одного с пометкой $x_1$, проводимость всего графа не изменится. Аналогично можно заключить, что в любом пучке, исходящем из цепочки $c_1$ можно выкинуть все ребра, за исключением одного с пометкой $x_1$.

Проводя подобные рассуждения, придем к выводу, что для $c_i$-й цепочки достаточно оставить ребра $x_1, \dots, x_i$, выходящие из нее. Но, с другой стороны, так как схема у нас центрально-симметричная, для каждой такой цепочки также достаточно оставить ребра $x_{i+h}, \dots, x_n$.

Далее можно выбрать для каждого пучка одно из этих множеств так, чтобы суммарная сложность графа была минимальной. В любом случае, сложность графа будет не больше, чем число ребер в этом графе, а их не больше $(n-h+1)h + (h-1) \cdot \left]\frac{n-h}{2}\right[^2$.

Это дает верхнюю оценку сложности функции

$$L(f_h) \leqslant h(n-h+1)^2.$$

Осталось заметить, что нижняя оценка следует из леммы 5 работы [2] и того, что $\zeta_1(f_h) = h$ и $\zeta_0(f_h) = n-h$ (функции $\zeta_1$ и $\zeta_0$ определены в [2]).

Лемма доказана.

**Замечание.** Лемма 9 говорит о том, что верхняя оценка сложности реализации некоторых пороговых функций (например

$f_{[\frac{n}{2}]}(x_1, \ldots, x_n))$ с помощью монотонных информационных графов меньше нижней оценки сложности их реализации с помощью деревьев (см. утверждение леммы 5). Это означает, что с помощью только деревьев оптимальной сложности достичь нельзя.

### 3.2. Случай деревьев

**Лемма 10.** *Для любой б. ф.* $f_h(x_1, \ldots, x_n) \in ST$ *при* $h \leqslant [\frac{n}{2}]$ *выполнено*

$$L_D(f_h) \leqslant 10 \cdot 2^{-h+1} C_n^h.$$

**Доказательство.** Рассмотрим разложение функции $f_h$ в следующем виде:

$$f_h(x_1, x_2, \ldots, x_n) =$$
$$= (x_1 \vee x_2) f_{h-1}(x_3, \ldots, x_n) \vee x_1 x_2 f_{h-2}(x_3, \ldots, x_n) \vee f_h(x_3, \ldots, x_n).$$

Любое разложение функции в базисе $\vee, \&$ задает способ построить граф или дерево из подграфов или поддеревьев, соответствующих подфункциям в разложении, а именно, дизъюнкция соответствует параллельному соединению подграфов или поддеревьев, а конъюнкция последовательному соединению графов или декартову произведению деревьев, то есть к каждой концевой вершине первого дерева пристраивается второе дерево.

Заметим также, что дерево $D_h^n(x_1, \ldots, x_n)$ для функции $f_h(x_1, \ldots, x_n)$ можно построить из дерева $D_{h-1}^n(x_1, \ldots, x_n)$ для $f_{h-1}(x_1, \ldots, x_n)$, присоединив к некоторым концевым вершинам последнего дерева ребра, которые будут вести в вершины, концевые для дерева $D_h^n$. А именно, если в дереве $D_{h-1}^n$ есть путь с предикатами $x_1, \ldots, x_{h-1}$ в концевую вершину этого дерева, то выпустив из этой вершины ребра с предикатами $x_h, x_{h+1}, \ldots, x_n$, мы получим, что они как раз образуют концевые вершины для дерева $D_h^n$ (получившиеся пути пропускают в точности наборы, соответствующие нижним единицам функции $f_h$). Очевидно также, что из всех концевых вершин дерева $D_{h-1}^n$ можно провести ровно по одному ребру для каждой нижней единицы функции $f_h$, то есть всего $C_n^h$ ребер.

Таким образом, мы можем реализовать дерево для функции $(x_1 \vee x_2) f_{h-1}(x_3, \dots, x_n) \vee f_h(x_3, \dots, x_n)$, построив дерево $D^{n-2}_{h-1}(x_3, \dots, x_n)$ и выпустив из всех его концевых вершин по два ребра с предикатами $x_1, x_2$, чтобы реализовать первую конъюнкцию, и $C^h_{n-2}$ ребер с предикатами $x_3,\ x_4,\ \dots\ ,\ x_{n-1}$ или $x_n$, необходимых для реализации функции $f_h(x_3, \dots, x_n)$.

Функцию $x_1 x_2 f_{h-2}(x_3, \dots, x_n)$ реализуем просто деревом $D^{n-2}_{h-2}(x_3, \dots, x_n)$, построенном на последовательно соединенных ребрах с предикатами $x_1$ и $x_2$.

Дерево для всей функции $f_h(x_1, x_2, \dots, x_n)$ получается, таким образом, путем параллельного объединения двух только что построенных деревьев.

Таким образом, мы задали алгоритм построения дерева $D^n_h(x_1, \dots, x_n)$ из деревьев $D^{n-2}_{h-1}(x_3, \dots, x_n)$ и $D^{n-2}_{h-2}(x_3, \dots, x_n)$.

Осталось определить, как будут выглядеть деревья $D^n_h$ для $h = 0,\ 1$, тогда остальные деревья будут строиться из них ($n$ не может быть меньше $h$).

Очевидно, что оптимальным деревом $D^n_0(x_1, \dots, x_n)$ для функции $f_0(x_1, \dots, x_n) = 0$ будет пустое дерево со сложностью 0, а для функции $f_1(x_1, \dots, x_n) = x_1 \vee \dots \vee x_n$ — дерево $D^n_1(x_1, \dots, x_n)$, состоящее из $n$ параллельных ребер с предикатами $x_1, x_2, \dots, x_n$, ведущих в концевые вершины. Сложность последнего дерева равна $n$.

По построению дерева $D^n_h(x_1, \dots, x_n)$, его сложность удовлетворяет рекуррентному соотношению

$$L(D^n_h) = L(D^{n-2}_{h-1}) + \underbrace{2^{-h+1}(2C^{h-1}_{n-2} + C^h_{n-2})}_{\text{ребра из концевых вершин } D^{n-2}_{h-1}} + \underbrace{1.5 + 0.25 \cdot L(D^{n-2}_{h-2})}_{x_1 x_2 D^{n-2}_{h-2}(x_3,\dots,x_n)}.$$

Покажем, что для некоторого $c(n)$ выполняется

$$L(D^n_h) \leqslant c(n) \cdot 2^{-h+1} C^h_n. \tag{4}$$

Очевидно, что при $h = 0, 1$ и любом $n$ оценка выполняется при $c(n) \geqslant 1$. Пусть $c(0) = c(1) = 1$.

Пусть теперь $h \geqslant 2$ (а следовательно и $n \geqslant 2$), и для всех $n_1, h_1 < n, h$ функция $c(n_1)$ определена так, что выполнена оценка (4).

Используя рекуррентное соотношение для $L(D_h^n)$ получаем

$$L(D_h^n) \leqslant c(n-2)\cdot 2^{-h+2}C_{n-2}^{h-1} + 2^{-h+1}(2C_{n-2}^{h-1} + C_{n-2}^{h}) + \\ + 1.5 + 0.25\cdot c(n-2)\cdot 2^{-h+3}C_{n-2}^{h-2} = \\ = 2^{-h+1}(2\cdot c(n-2)\cdot C_{n-2}^{h-1} + 2C_{n-2}^{h-1} + C_{n-2}^{h} + 0.25\cdot c(n-2)\cdot 4C_{n-2}^{h-2}) + 1.5 = \\ = 2^{-h+1}C_n^h \tfrac{(2\cdot c(n-2)+2)\cdot h(n-h)+(n-h)(n-h-1)+c(n-2)\cdot h(h-1)}{n(n-1)} + 1.5 = \\ = 2^{-h+1}C_n^h\left(\tfrac{c(n-2)\cdot h(2n-h-1)+(n-h)(n+h-1)}{n(n-1)} + \tfrac{1.5\cdot 2^{h-1}}{C_n^h}\right).$$

Пусть $d(n) := \max\limits_{h\leqslant\frac{n}{2}} \frac{c(n-2)\cdot h(2n-h-1)+(n-h)(n+h-1)}{n(n-1)}$, а $c(n) = d(n) + \max\limits_{h\leqslant\frac{n}{2}} \frac{1.5\cdot 2^{h-1}}{C_n^h}$, $n \geqslant 2$. Тогда выполнено $L(D_h^n) \leqslant c(n)\cdot 2^{-h+1}C_n^h$, и осталось лишь оценить $c(n)$.

При $c(n) \geqslant 1$ максимум в определении $d(n)$ достигается в точке $[\frac{n}{2}]$. Действительно, производная по $h$ числителя равна $c(n-2)(2n-2h-1) - 2h + 1$ и не убывает при $h \leqslant \frac{c(n-2)n}{1+c(n-2)} + \frac{1-c(n-2)}{2(1+c(n-2))}$, (следовательно и при всех $h \leqslant \frac{n}{2}$).

Тогда

$$d(n) = \frac{c(n-2)\cdot[\frac{n}{2}](2n-[\frac{n}{2}]-1) + (n-[\frac{n}{2}])(n+[\frac{n}{2}]-1)}{n(n-1)}.$$

Если $n$ — четное, то получим

$$d(n) = \frac{c(n-2)\cdot(n^2-\frac{n^2}{4}-\frac{n}{2}) + (\frac{3n^2}{4}-\frac{n}{2})}{n(n-1)} = \\ = \frac{3}{4}(c(n-2)+1)\left(1-\frac{1}{3(n-1)}\right).$$

Если $n$ — нечетное, то получим

$$d(n) = \frac{c(n-2)\cdot\frac{n-1}{2}(2n-\frac{n-1}{2}-1) + \frac{n+1}{2}(\frac{3n-1}{2}-1)}{n(n-1)} = \\ = \frac{3}{4}\left(c(n-2)(1-\frac{1}{3n}) + 1 + \frac{1}{n}\right).$$

Таким образом, видим, что $c(n) \geqslant d(n) \geqslant 1$, и $c(n) \to 3,\ n \to \infty$.

Очевидно, что при $h \leqslant \frac{n}{2}$ имеем $\frac{1.5 \cdot 2^{h-1}}{C_n^h} \leqslant 1$. Причем если $c(n-2) \leqslant 10$, то $d(n) \leqslant 9$, и следовательно, $c(n) \leqslant 10$.

Значит $\forall n\ c(n) \leqslant 10$, так как $c(0) = c(1) = 1 < 10$.

Это означает, что

$$L(D_h^n) \leqslant 10 \cdot 2^{-h+1} C_n^h.$$

Но сложность функции по определению не больше чем сложность дерева, которое ее реализует. Таким образом, лемма доказана.

Объединяя результат этой леммы с утверждением лемм 5 и 9 получаем утверждение теоремы 3.

## 4. Реализация произвольной монотонной булевой функции

Для произвольной монотонной б. ф. $f$ необходимо построить монотонный информационный граф, который ее реализует и обладает как можно меньшей сложностью.

Начнем с построения информационного дерева, реализующего функцию $f$. Причем будем строить его путем выделения некоторых вершин и ребер булева куба (рассматриваем булев куб как граф $B_n$). Можно считать, что каждому ребру булева куба приписана переменная, которая различает наборы, связанные этим ребром. Тогда ребру информационного дерева будет приписан соответствующий предикат.

Корнем дерева будет являться вершина булева куба, соответствующая набору $(0, 0, \dots, 0)$.

Рассмотрим следующий алгоритм выделения вершин и ребер булева куба. Будем просматривать куб сверху вниз по уровням (то есть от набора $(1, 1, \dots, 1)$ до набора $(0, 0, \dots, 0)$). На каждом уровне будем выделять множество вершин куба которые войдут в информационное дерево, и в которые надо провести цепочку из начальной вершины дерева. Вместе с выделением такого множества будем описывать, какими ребрами нужно соединить эти вершины между собой, чтобы в итоге получилось дерево достаточно маленькой сложности.

Для каждого слоя $i$ через $\Lambda_i$ обозначим множество вершин куба, в которые будет проведена цепочка из корня дерева.

Обозначим через $A$ рабочее множество вершин, с которым будет работать алгоритм. Шаг алгоритма будет состоять в определении множества $\Lambda_i$ для $i$-го слоя. На первом шаге определится множество $\Lambda_n$, на последнем — множество $\Lambda_0$.

Итак, сначала рассмотрим множество всех нижних единиц функции $f$. Через $\lambda_i$ обозначим множество тех нижних единиц функции $f$, которые лежат на $i$-м слое булева куба.

Во-первых, для каждого слоя $i$ в $\Lambda_i$ войдут все нижние единицы этого слоя, то есть $\lambda_i \subseteq \Lambda_i$. Для верхнего слоя $n$ больше в $\Lambda_n$ ничего входить не будет (заметим, что он может оказаться и пустым).

Далее, пусть для какого-то слоя $i+1$ мы определили $\Lambda_{i+1}$.

Занесем его в множество $A$ и выполним шаг алгоритма для слоя $i$. Изначально в $\Lambda_i$ находятся все нижние единицы слоя $i$. Далее будем брать по 4 вершины из $A$ таких, что они имеют общую смежную вершину на слое $i$. Для каждой такой четверки будем заносить общую смежную вершину в $\Lambda_i$, а сами вершины удалять из рабочего множества. Пусть мы проделали данную операцию для всех возможных четверок из рабочего множества, и больше ее нельзя применить ни к какой оставшейся четверке вершин. Тогда для всех оставшихся в рабочем множестве вершин мы выбираем произвольную смежную вершину на слое $i$ и заносим ее в $\Lambda_i$, а саму вершину удаляем из рабочего множества.

В результате применения двух таких операций рабочее множество станет пустым, а $\Lambda_i$ будет содержать те вершины, которые войдут в дерево, реализующее функцию $f$.

Дерево, реализующее функцию $f$, тем самым, строится следующим очевидным образом. Мы располагаем множества $\Lambda_0$ (корень дерева), $\Lambda_1, \ldots, \Lambda_n$ последовательно на разных уровнях от корня и соединяем вершины на соседних слоях ребрами согласно тому, как формировались множества $\Lambda_i$. А именно, если вершина $v$ на слое $i$ была образована в результате первой операции, то она соединяется ребрами с четырьмя вершинами на $(i+1)$-м слое, которые ее образовали. Если она была образована одной вершиной, то она соединяется ребром с этой вершиной.

Все вершины, входящие в $\lambda_i, i = 0, \ldots, n$ выделяем как концевые вершины дерева. Очевидно, что в результате мы получим дерево такое, что каждой нижней единице функции $f$ соответствует вершина этого дерева, и к этой вершине идет ровно одна цепочка, проводящая все наборы, большие либо равные этой нижней единицы.

Таким образом, дерево в точности реализует функцию $f$.

Оценим теперь сложность такого дерева.

Во-первых, оценим число вершин на каждом слое $i+1$ булева куба $B_n$, к которым могла быть применена операция 2 нашего алгоритма. Множество таких вершин $K_i$ должно обладать свойством, что никакие 4 его вершины не имеют смежной вершины на $i$-м слое булева куба. Известно, что каждая вершина $(i+1)$-го слоя булева куба $B_n$ соединена с $i+1$ вершинами $i$-го слоя, а каждая вершина $i$-го слоя соединена с $n-i$ вершинами $(i+1)$-го слоя. Тогда для любой вершины $i$-го слоя есть как минимум $n-i-3$ вершины на $(i+1)$-м слое, которые не принадлежат $K_i$. Так как на $i$-м слое $S(i) = C_n^i$ вершин, то всего получим $S(i)\cdot(n-i-3)$ вершин, не принадлежащих $K_i$. Однако, некоторые вершины могли быть посчитаны несколько раз, но не более, чем $i+1$. Следовательно, количество вершин на $i+1$-м слое, не принадлежащих $K_i$ как минимум $S(i)\frac{n-i-3}{i+1}$. Мощность множества $K_i$, тем самым, не более, чем $S(i+1)-S(i)\frac{n-i-3}{i+1} = S(i)(\frac{n-i}{i+1}-\frac{n-i-3}{i+1}) = S(i)\frac{3}{i+1}$.

Для дерева подсчет сложности не представляет труда, если известны количества вершин на каждом слое, а именно, сложность дерева в нашем случае будет равна

$$L(D) = \sum_{i=1}^{n} 2^{-i+1}|\Lambda_i|. \tag{5}$$

Таким образом, нам нужно оценить лишь мощности множеств $\Lambda_i$.

Каждое множество $\Lambda_i$ является объединением трех множеств: $\lambda_i$, $A_i^1$, которое состоит из вершин, порожденных операцией 1 алгоритма построения дерева, и $A_i^2$, которое состоит из вершин, порожденных операцией 2 алгоритма.

Для $A_i^2$ оценка следует из только что доказанного утверждения для множества $K_i$, а именно, $|A_i^2| \leqslant S(i)\frac{3}{i+1}$ (каждая вершина из $K_i$ порождает ровно одну вершину в $A_i^2$). Множество $A_i^1$ оценим сверху как $\frac{|\Lambda_{i+1}|}{4}$.

Таким образом, имеем систему неравенств:

$$\begin{aligned}
&|\Lambda_n| \leqslant |\lambda_n|,\\
&|\Lambda_{n-1}| \leqslant |\lambda_{n-1}| + \tfrac{3}{n}S(n-1) + \tfrac{|\Lambda_n|}{4},\\
&\ldots\\
&|\Lambda_i| \leqslant |\lambda_i| + \tfrac{3}{i+1}S(i) + \tfrac{|\Lambda_{i+1}|}{4},\\
&\ldots\\
&|\Lambda_1| \leqslant |\lambda_1| + \tfrac{3}{2}S(1) + \tfrac{|\Lambda_2|}{4}.
\end{aligned}$$

Домножая на соответствующие коэффициенты, получаем:

$$\begin{aligned}
&2^{-n+1}|\Lambda_n| \leqslant 2^{-n+1}|\lambda_n|,\\
&2^{-n+2}|\Lambda_{n-1}| \leqslant 2^{-n+2}|\lambda_{n-1}| + 2^{-n+2}\tfrac{3}{n}S(n-1) + \tfrac{2^{-n+1}|\Lambda_n|}{2},\\
&\ldots\\
&2^{-i+1}|\Lambda_i| \leqslant 2^{-i+1}|\lambda_i| + 2^{-i+1}\tfrac{3}{i+1}S(i) + \tfrac{2^{-i}|\Lambda_{i+1}|}{2},\\
&\ldots\\
&|\Lambda_1| \leqslant |\lambda_1| + \tfrac{3}{2}S(1) + \tfrac{2^{-1}|\Lambda_2|}{2}.
\end{aligned}$$

Суммируя все неравенства, получаем

$$\sum_{i=1}^{n} 2^{-i+1}|\Lambda_i| \leqslant \sum_{i=1}^{n} 2^{-i+1}|\lambda_i| + \sum_{i=1}^{n-1} 2^{-i+1}\frac{3}{i+1}S(i) + \frac{1}{2}\sum_{i=1}^{n} 2^{-i+1}|\Lambda_i|$$

Переносим самую правую сумму в левую часть и подставляем полученную оценку в (5). Получаем

$$L(D) \leqslant \sum_{i=1}^{n} 2\cdot 2^{-i+1}|\lambda_i| + \sum_{i=1}^{n-1} 2\cdot 2^{-i+1}\frac{3}{i+1}S(i). \tag{6}$$

Второе слагаемое не зависит от конкретной функции и может быть оценено следующим образом:

$$\sum_{i=1}^{n-1} 2\cdot 2^{-i+1}\frac{3}{i+1}S(i) = 12\sum_{i=1}^{n-1} 2^{-i}\frac{n!}{(i+1)i!(n-i)!} = \frac{12}{n+1}\sum_{i=1}^{n-1} 2^{-i}C_{n+1}^{i+1}.$$

Через $R_n(i)$ обозначим величину $2^{-i+1}C_n^i$, которая равна сложности всех ребер, входящих на $i$-й слой булева куба. Именно эта величина являлась нижней оценкой сложности реализации пороговой функции $f_i$ с помощью деревьев. $R_n = \max\limits_{i=1,\ldots,n} R_n(i)$ — максимальная сложность слоев булева куба. В разделе 2 показано, что $R_n = R_n([\frac{n+1}{3}])$ является нижней оценкой функции Шеннона в классе деревьев.

Продолжим оценку второго слагаемого из неравенства (6).

$$\sum_{i=1}^{n-1} 2\cdot 2^{-i+1}\frac{3}{i+1}S(i)\leqslant\frac{12}{n+1}\sum_{i=1}^{n-1}R_{n+1}(i+1)\leqslant 12R_{n+1}\sim 18R_n,\ n\to\infty.$$

Для оценки первого слагаемого установим ряд утверждений.

Введем понятие тени точки и множества.

Пусть дано некоторое множество $S$ вершин булева куба $\{0,1\}^n$. *Тенью* $T(S)$ множества $S$ назовем множество таких наборов из $\{0,1\}^n$, что для каждого из них имеется предшествующий либо равный набор из $S$. То есть $T(S) = \{v \mid \exists w \in S : w \preceq v\}$.

Пусть $v$ — произвольная вершина $k$-го слоя из $\{0,1\}^n$. Множество вершин $w$ $(k+r)$-го слоя из $\{0,1\}^n$, где $r \geqslant 1$ и $k+r \leqslant n$, таких, что $w \succeq v$, назовём *$r$-тенью* вершины $v$.

Пусть $S = \{v_1, \ldots, v_m\}$ — некоторое множество вершин из $\{0,1\}^n$, в каждой из которых имеется не более $n-r$ единиц. При каждом $j = 1, 2, \ldots, r$ обозначим через $T^j(S)$ множество вершин из $\{0,1\}^n$, каждая из которых принадлежит $j$-тени некоторой вершины из $S$. Множество $T^j(S)$ назовём *$j$-тенью* множества $S$.

**Лемма 11.** *Пусть $M$ — некоторое множество вершин $i$-го слоя булева куба $\{0,1\}^n$, тогда $\frac{|M|}{S(i)} \leqslant \frac{|T^1(M)|}{S(i+1)}$. Другими словами, доля вершин, занимаемая множеством $M$ на слое булева куба не уменьшается при переходе к 1-тени этого множества.*

**Доказательство.** Известно, что каждая вершина $(i+1)$-го слоя соединена с $i+1$ вершинами $i$-го слоя, а каждая вершина $i$-го слоя соединена с $n-i$ вершинами $(i+1)$-го слоя. Тогда любая вершина из множества $M$ соединена в точности с $n-i$ вершинами из множества $T^1(M)$ на $(i+1)$-м слое. Однако, некоторые вершины из $T^1(M)$ могут быть соединены с несколькими вершинами из $M$, но не более, чем с $i+1$. Следовательно, количество вершин в $T^1(M)$ не меньше, чем $\frac{|M|\cdot(n-i)}{i+1}$. Между $S(i)$ и $S(i+1)$ можно записать очевидное соотношение $(i+1)S(i+1) = (n-i)S(i)$. Отсюда следует, что

$$\frac{|T^1(M)|}{S(i+1)} \geqslant \frac{|M|\cdot(n-i)}{(i+1)S(i+1)} = \frac{|M|\cdot(n-i)}{(n-i)S(i)} = \frac{|M|}{S(i)}.$$

Что и требовалось доказать.

**Лемма 12.** *Пусть дана монотонная функция $f$, и $\lambda_i$ — множество ее нижних единиц на $i$-м слое булева куба, тогда выполнено неравенство*

$$\sum_{i=0}^{n} \frac{|\lambda_i|}{S(i)} \leqslant 1.$$

**Доказательство.** Обозначим через $\mu_i$ множество вершин $i$-го слоя булева куба, принадлежащих тени множества $\lambda_0 \cup \lambda_1 \cup \dots \cup \lambda_i$. Очевидно, что $\mu_0 = \lambda_0$ и $\mu_i = T^1(\mu_{i-1}) \sqcup \lambda_i,\ i > 0$. Это следует из того, что никакая нижняя единица не может принадлежать тени другой нижней единицы, и того, что $T(S) = T^1(S) \cup T^1(T^1(S)) \cup \dots \cup T^1(T^1(\dots(T^1(S))\dots))$.

Рассмотрим теперь величины $\frac{|\mu_i|}{S(i)}$. Для $i=0$ имеем $\frac{|\mu_0|}{S(0)} = \frac{|\lambda_0|}{S(0)}$, а для $i > 0$ имеем $\frac{|\mu_i|}{S(i)} = \frac{|T^1(\mu_{i-1})|+|\lambda_i|}{S(i)} = \frac{|T^1(\mu_{i-1})|}{S(i)} + \frac{|\lambda_i|}{S(i)}$. Используя лемму 11 имеем

$$\frac{|\mu_i|}{S(i)} = \frac{|T^1(\mu_{i-1})|}{S(i)} + \frac{|\lambda_i|}{S(i)} \geqslant \frac{|\mu_{i-1}|}{S(i-1)} + \frac{|\lambda_i|}{S(i)}.$$

Очевидно, что доля вершин, принадлежащих тени нижних единиц на каждом слое не больше 1. В том числе это выполнено для $n$-го слоя. Используя только что полученную оценку для $\frac{|\mu_i|}{S(i)}$ получаем

$$1 \geqslant \frac{|\mu_n|}{S(n)} \geqslant \frac{|\mu_{n-1}|}{S(n-1)} + \frac{|\lambda_n|}{S(n)} \geqslant \dots \geqslant \frac{|\lambda_0|}{S(0)} + \dots + \frac{|\lambda_n|}{S(n)}.$$

Лемма доказана.

Заметим, что равенство в этом неравенстве достигается на любой пороговой функции $f_h \in ST$, так как все нижние единицы таких функций расположены на одном слое булева куба.

Итак, вернемся к оценке первого слагаемого из неравенства (6). Очевидно, выполнено

$$\sum_{i=1}^{n} 2 \cdot 2^{-i+1}|\lambda_i| = \sum_{i=1}^{n} 2 \cdot 2^{-i+1} S(i)\frac{|\lambda_i|}{S(i)} = \\ = 2 \cdot \sum_{i=1}^{n} R_n(i)\frac{|\lambda_i|}{S(i)} \leqslant 2 \cdot R_n \sum_{i=1}^{n} \frac{|\lambda_i|}{S(i)} \leqslant 2R_n.$$

Объединяя с оценкой для второго слагаемого, получаем

$$L(D) \leqslant 2R_n + 12R_{n+1} \sim 20R_n,\ n \to \infty.$$

Мы получили верхнюю оценку на сложность реализации произвольной мон. б. ф. $f$ с помощью деревьев, таким образом, справедлива следующая лемма.

**Лемма 13.** *Для любой мон. б. ф.* $f(x_1, \ldots, x_n) \in M$ *выполнено*

$$L_D(f) \leqslant 2R_n + 12R_{n+1}.$$

Объединяя этот результат с нижней оценкой функции Шеннона в классе графов (лемма 8) и в классе деревьев (следствие 2) получаем утверждение теоремы 1.

## 5. Оценки сложности для почти всех монотонных булевых функций

В этом разделе мы рассмотрим оценки сложности реализации почти всех монотонных булевых функций с помощью монотонных информационных графов.

Основой этих оценок будет служить результат Коршунова А. Д. о конструкции монотонных булевых функций.

**Теорема 4 ([4]).** *Почти каждая монотонная б. ф.* $f(x_1, \ldots, x_n)$ *имеет сокращенную д.н.ф.* $D_c(f)$ *такую, что при* $n \to \infty$

- $D_c(f)$ *состоит из* $r$ *конъюнкций, где* $r \sim \frac{1}{2} C_n^{[\frac{n}{2}]}$,
- *при четных* $n$ *в каждую конъюнкцию из* $D_c(f)$ *входит не менее* $\frac{n}{2} - 1$ *и не более* $\frac{n}{2} + 1$ *переменных,*
- *при нечетных* $n$ *в каждую конъюнкцию из* $D_c(f)$ *входит не менее* $\frac{n-3}{2}$ *и не более* $\frac{n+3}{2}$ *переменных.*

Эту теорему можно также переформулировать другими словами, а именно, что почти все монотонные булевы функции имеют асимптотически $\frac{1}{2} C_n^{[\frac{n}{2}]}$ нижних единиц и все эти единицы расположены на слоях с $[\frac{n}{2}] - 1$ по $]\frac{n}{2}[+1$.

Далее будем использовать идею из раздела 2, состоящую в том, что сложность информационного дерева, реализующего монотонную б. ф. $f$, не меньше суммарной сложности ребер, входящих в концевые вершины, а тех в свою очередь не меньше числа нижних единиц функции $f$.

**Лемма 14.** *Для почти всех мон. б. ф.* $f(x_1, \ldots, x_n) \in M$ *выполнено*

$$L_D(f) \gtrsim \frac{2^{[\frac{n}{2}]}}{2\sqrt{\pi n}}, n \to \infty.$$

**Доказательство.** Возьмем произвольную функцию $f$, удовлетворяющую свойствам в утверждении теоремы 4.

Для нее существует информационное дерево $D$, которое реализует эту функцию оптимально.

Из леммы 2 следует, что каждой нижней единице функции $f$ соответствует по крайней мере одна концевая вершина дерева $D$, причем для разных нижних единиц эти вершины разные. Так как все нижние единицы функции $f$ расположены не выше $]\frac{n}{2}[+1$-го слоя, то из леммы 7 следует, что сложность любого ребра в дереве $D$ не меньше $2^{-]\frac{n}{2}[}$, в том числе всех ребер, ведущих в концевые вершины.

Поэтому сложность дерева $D$ оценивается снизу как число нижних единиц функции $f$ умноженное на минимальную сложность ребра. Имеем

$$L(D) \gtrsim \frac{1}{2} C_n^{[\frac{n}{2}]} \cdot 2^{-]\frac{n}{2}[} \sim \frac{2^n}{\sqrt{\pi n}} \cdot 2^{-]\frac{n}{2}[-1} \sim \frac{2^{[\frac{n}{2}]-1}}{\sqrt{\pi n}},\ n \to \infty.$$

Данное неравенство доказывает лемму.

**Лемма 15.** *Для почти всех мон. б. ф.* $f(x_1, \dots, x_n) \in M$ *выполнено*

$$L(f) \gtrsim \frac{2^{[\frac{n}{2}]}}{2n\sqrt{\pi n}},\ n \to \infty.$$

**Доказательство.** Для получения нижней оценки в классе графов воспользуемся мощностной оценкой, полученной в разделе 2.

Рассмотрим множество функций $M^{(n)}_{]\frac{n}{2}[+1}$, то есть тех функций, у которых нижние единицы расположены на слоях не выше $]\frac{n}{2}[+1$. Теорема 4 утверждает, что $M^{(n)}_{]\frac{n}{2}[+1}$ содержит почти все монотонные функции от $n$ переменных. С другой стороны, из неравенства (2) леммы 8 следует, что в этом множестве доля функций, сложность которых не больше $(1-\varepsilon)\frac{2^{-]\frac{n}{2}[-1} C_n^{]\frac{n}{2}[+1}}{n \cdot d(]\frac{n}{2}[+1, n)}$ стремится к 0 для любого $\varepsilon > 0$.

Отсюда можно заключить, что для почти всех монотонных функций $f$ справедлива следующая оценка на их сложность:

$$L(f) \gtrsim \frac{2^{-]\frac{n}{2}[-1} C_n^{]\frac{n}{2}[+1}}{n \cdot d(]\frac{n}{2}[+1, n)} \sim \frac{2^{[\frac{n}{2}]-1}}{n\sqrt{\pi n}},\ n \to \infty.$$

Лемма доказана.

Теперь приведем верхнюю оценку для сложности реализации монотонных функций с помощью монотонных информационных деревьев.

**Лемма 16.** *Для почти всех мон. б. ф.* $f(x_1, \dots, x_n) \in M$ *выполнено*

$$L_D(f) \lesssim 42 \cdot 2^{-[\frac{n}{2}]} \cdot C_n^{[\frac{n}{2}]},\ n \to \infty.$$

**Доказательство.** Рассмотрим пороговую функцию $f_{[\frac{n}{2}]-1} \in ST^{(n)}$.

По теореме 4 почти все монотонные булевы функции $f \in M^{(n)}$ меньше нее, то есть если функция $f$ принимает на каком-то наборе значение 1, то функция $f_{[\frac{n}{2}]-1}$ тоже принимает на нем значение 1.

Рассмотрим произвольную такую функцию $f$ и построим дерево $D$, которое ее реализует.

Раз $f$ меньше $f_{[\frac{n}{2}]-1}$, значит для любой нижней единицы $\alpha$ функции $f$ найдется нижняя единица $\beta$ функции $f_{[\frac{n}{2}]-1}$ такая, что $\alpha \succeq \beta$.

Пусть $D^n_{[\frac{n}{2}]-1}$ — дерево, реализующее функцию $f_{[\frac{n}{2}]-1}$. Из леммы 2 следует, что тогда в этом дереве разные концевые вершины соответствуют разным нижним единицам функции $f_{[\frac{n}{2}]-1}$. Дерево $D$ должно обладать таким же свойством относительно функции $f$.

Дерево $D$ будем получать путем достраивания дерева $D^n_{[\frac{n}{2}]-1}$.

Для каждой нижней единицы $\beta$ функции $f_{[\frac{n}{2}]-1}$ через $D^n_{[\frac{n}{2}]-1}(\beta)$ обозначим одну из концевых вершин дерева $D^n_{[\frac{n}{2}]-1}$, соответствующих этой нижней единице (то есть концевую вершину, в которую проходит нижняя единица $\beta$).

Пусть $\alpha = (a_1, \ldots, a_n)$ — нижняя единица функции $f$. Пусть $\beta = (b_1, \ldots, b_n)$ — некоторая нижняя единица функции $f_{[\frac{n}{2}]-1}$ такая, что $\alpha \succeq \beta$. Пусть они различаются в компонентах $x_{i_1}, \ldots, x_{i_s}$. Это значит, что $a_{i_j} = 1, b_{i_j} = 0,\ j = 1, \ldots, s$. Пристроим к вершине $D^n_{[\frac{n}{2}]-1}(\beta)$ цепочку ребер с предикатами $x_{i_1}, \ldots, x_{i_s}$ и отметим конечную вершину этой цепочки как концевую вершину дерева $D$. Тогда понятно, что в эту концевую вершину пройдут в точности те наборы, которые не меньше набора $\alpha$.

Теперь, если мы проделаем данную операцию для каждой нижней единицы функции $f$, то получим, что дерево $D$ и итоге будет реализовывать функцию $f$.

Теперь считаем, что дерево $D^n_{[\frac{n}{2}]-1}$ в точности то, которое использовалось в доказательстве леммы 10. Оно реализует функцию $f_{[\frac{n}{2}]-1}$.

Тогда, во-первых, его сложность удовлетворяет неравенству

$$L(D^n_{[\frac{n}{2}]-1}) \leqslant 10 \cdot 2^{-[\frac{n}{2}]+2} \cdot C_n^{[\frac{n}{2}]-1},$$

а во-вторых, вероятность прохождения набора в концевую вершину дерева $D^n_{[\frac{n}{2}]-1}$ равна $2^{-[\frac{n}{2}]+1}$.

Далее, сложность каждой цепочки из концевых вершин дерева $D^n_{[\frac{n}{2}]-1}$ к концевым вершинам дерева $D$ не более $2^{-[\frac{n}{2}]+1}\cdot 2$ (на самом деле их сложность равна $2^{-[\frac{n}{2}]+1}(2-2^{-l+1})$, где $l$ — длина цепочки).

Количество нижних единиц функции $f$ (а значит и число концевых вершин дерева $D$) асимптотически равно $\frac{1}{2}C_n^{[\frac{n}{2}]}$.

Суммируя все вышесказанное получаем оценку сложности функции $f$:

$$L_D(f) \lesssim 10\cdot 2^{-[\frac{n}{2}]+2}\cdot C_n^{[\frac{n}{2}]-1} + 2^{-[\frac{n}{2}]+1}\cdot 2\cdot \frac{1}{2}C_n^{[\frac{n}{2}]},\ n\to\infty.$$

Приводя подобные окончательно получаем

$$L_D(f) \lesssim 42\cdot 2^{-[\frac{n}{2}]}\cdot C_n^{[\frac{n}{2}]},\ n\to\infty.$$

Лемма доказана.

Объединяя с нижними оценками из лемм 14 и 15, получаем утверждение теоремы 2.

Автор выражает благодарность Гасанову Эльяру Эльдаровичу за постановку задачи и помощь в проведении исследования, а также Кудрявцеву Валерию Борисовичу за внимание, проявленное к работе.

## Список литературы

[1] Гасанов Э. Э., Кудрявцев В. Б. Теория хранения и поиска информации. М.: Физматлит, 2002.

[2] Шуткин Ю. С. О реализации булевых функций информационными графами // Дискретная математика. 2008. 20: 4. С. 31–41.

[3] Лупанов О. Б. Асимптотические оценки сложности управляющих систем. М.: Изд-во МГУ, 1984.

[4] Коршунов А. Д. О числе монотонных булевых функций // Проблемы кибернетики. 1981. Т. 38.