Санкт-Петербургский государственный университет

СМИРНОВ Пётр Юрьевич

Выпускная квалификационная работа

Системы доказательств, основанные на диаграммах принятия решений

Уровень образования: магистратура
Направление: 02.04.03 «Математическое обеспечение и администрирование информационных систем»
Основная образовательная программа: ВМ.5665.2018 «Математическое обеспечение и администрирование информационных систем»

Научный руководитель: доцент
факультета математики и
компьютерных наук
к. ф.-м. н.
Ицыксон Дмитрий Михайлович

Рецензент: доцент
Калифорнийский университет
в Сан-Диего (UCSD)
к. ф.-м. н.
Кноп Александр Анатольевич

Санкт-Петербург
2020 г.

Saint Petersburg State University

Petr Smirnov

Graduation qualification thesis

On proof systems based on binary decision diagrams

Master's program
Specialization: 02.04.03 "Software and Administration of Information Systems"
Study program: VM.5665.2018 "Software and Administration of Information Systems"

Thesis advisor: assistant professor
Department of Mathematics and
Computer Science
Ph.D.
Dmitry M. Itsykson

Reviewer: assistant professor
the University of California,
San Diego
Ph.D.
Alexander A. Knop

Saint Petersburg
2020

Оглавление

1. Введение

В этой работе мы изучаем сложность цейтинских формул $\mathrm{T}(G, c)$, построенных на графах $G(V, E)$. Основной рассматриваемой моделью вычислений будет однопроходная ветвящаяся программа (1-BP). Мы описываем структуру 1-BP, решающих задачу $\mathrm{SearchVertex}(G, c)$ поиска вершины с нарушенным условием чётности для невыполнимой $\mathrm{T}(G, c)$. По программе размера S мы строим: (1) 1-BP размера $S^{\mathcal{O}(\log |V|)}$, вычисляющую значение выполнимой $\mathrm{T}(G, c')$; (2) вывод формулы $\neg\mathrm{T}(G, c)$ размера $S \cdot \mathrm{poly}(|\mathrm{T}(G, c)|)$ в системе Фреге константной глубины. Из (1) следует, что если размер регулярного резолюционного опровержения $\mathrm{T}(G, c)$ равен S, то размер 1-BP, вычисляющей значение выполнимой $\mathrm{T}(G, c')$, не больше $S^{\mathcal{O}(\log |V|)}$.

1.1. Пропозициональные доказательства

Булевы функции $f\colon \{0,1\}^n \to \{0,1\}$ (0 и 1 здесь и далее будем отождествлять со значениями лжи и истины соответственно) можно записывать с помощью пропозициональных формул — пропозициональных переменных, соединённых логическими связками (конъюнкцией, дизъюнкцией, отрицанием и другими). Конъюктивная нормальная форма (КНФ) — формула ϕ, записанная в виде $\bigwedge_{i=1}^{m}(\bigvee_{j=1}^{n_i} l_{i,j})$, где $l_{i,j}$ — литерал (пропозициональная переменная или её отрицание), а каждая скобка $(\bigvee_{j=1}^{n_i} l_{i,j})$ называется дизъюнктом ϕ.

Рассмотрим задачу проверки формулы в КНФ на выполнимость — то есть задачу проверки существования набора значений переменных, при подстановке которых формула обращается в истину. По теореме Кука-Левина [7, 16], эта задача является NP-полной.

Пусть мы каким-то образом узнали, выполнима ли формула, и теперь хотим предъявить доказательство того, что ответ действительно такой. Если формула выполнима, достаточно предъявить выполняющий набор переменных.

Будем называть *пропозициональной системой доказательств* полиномиальный алгоритм $V(\phi, w)$, который принимает формулу ϕ в КНФ

и строку w, возвращает 0 или 1 и обладает свойствами:

- Корректность: если $V(\phi, w) = 1$, то ϕ невыполнима; w в этом случае называется доказательством ϕ.

- Полнота: если ϕ невыполнима, то существует строка w такая, что $V(\phi, w) = 1$.

Существует ли пропозициональная система доказательств, в которой для любой невыполнимой формулы ϕ существует доказательство полиномиального от длины ϕ размера? В [6] было показано, что такая система доказательств существует тогда и только тогда, когда классы сложности NP и coNP совпадают. В частности, если такой системы доказательств не существует, то классы P и NP различаются.

Основная программа исследований в теории сложности доказательств заключается в доказательстве суперполиномиальных нижних оценок для как можно более сильных систем доказательств. Для многих интересных из них неизвестны нетривиальные нижние оценки (для систем Фреге, для полуалгебраических систем высокой степени и пр.).

Нижние оценки на системы доказательств позволяют получать сложные примеры для алгоритмов, которые решают задачу выполнимости. Так, алгоритмы расщепления (DPLL) эквивалентны древовидным резолюционным доказательствам, а современные используемые на практике алгоритмы, запоминающие конфликты (CDCL), соответствуют резолюционным доказательствам общего вида.

Резолюционная система доказательств — классическая система, использующая правило резолюции. *Правило резолюции* позволяет из дизъюнктов $x \vee A$ и $\neg x \vee B$ вывести дизъюнкт $A \vee B$. Доказательством ϕ в резолюционной системе является вывод с помощью правила резолюции пустого дизъюнкта (то есть тождественно ложного) из дизъюнктов исходной формулы. Размер доказательства — это количество дизъюнктов в нём.

Доказательство называется древовидным, если любой выведенный дизъюнкт используется затем не более одного раза (чтобы использовать повторно, необходимо вывести его ещё раз). Доказательство на-

зывается регулярным, если на любом пути в графе доказательства не происходит резолюции по одной и той же переменной больше одного раза. Минимальный размер древовидного резолюционного доказательства невыполнимой формулы ϕ обозначим за $S_{\mathrm{T}}(\phi)$, регулярного резолюционного доказательства — за $S_{\mathrm{Reg}}(\phi)$.

1.2. Задача Search_ϕ

В теории сложности доказательств частым объектом изучения является задача поиска невыполненного дизъюнкта. А именно, пусть дана невыполнимая формула ϕ, записанная в КНФ. Тогда для любой постановки значений переменных формулы хотя бы один из её дизъюнктов будет не выполнен. Задача Search_ϕ заключается в том, чтобы по данной подстановке значений выдать номер любого невыполненного дизъюнкта.

Эта задача интересна тем, что существуют связи между доказательствами в некоторых системах доказательств и решением Search_ϕ в соответствующих им моделях вычислений. Так, размер минимального *древовидного* резолюционного опровержения формулы ϕ равен размеру минимального *дерева решений*, решающего задачу Search_ϕ, а размер минимального *регулярного* древовидного опровержения формулы ϕ равен размеру минимальной *однопроходной ветвящейся программы*, решающей задачу Search_ϕ [17].

1.3. Цейтинские формулы

Цейтинские формулы [19] кодируют следующий принцип: в графе чётное число вершин нечётной степени. Цейтинская формула $\mathrm{T}(G, c)$ строится по неориентированному графу $G(V, E)$ и функции пометок $c\colon V \to \{0, 1\}$, каждая переменная формулы соответствует ребру графа. Формула представляет собой конъюнкцию условий чётности в каждой из вершин; условие для вершины v выполнено, если сумма значений переменных x_e по рёбрам e, инцидентным v, равна $c(v)$ по модулю два. Каждое такое условие мы записываем в КНФ, а значит, и вся формула

получается записанной в КНФ. Формула выполнима тогда и только тогда, когда для любой компоненты связности сумма пометок в вершинах четна [20].

Цейтинские формулы представляют интерес в сложности доказательств, так как на них были получены нижние оценки на сложность вывода во многих системах доказательств [20, 2, 18, 11, 5, 10].

Для невыполнимой цейтинской формулы $\mathrm{T}(G, c)$ можно рассмотреть задачу $\mathrm{SearchVertex}(G, c)$ — по данной подстановке найти любую вершину, для которой нарушено условие чётности. Легко видеть, что сложность такой задачи не выше, чем сложность задачи $\mathrm{Search}_{\mathrm{T}(G,c)}$: если мы нашли невыполненный дизъюнкт, то уже точно знаем, в какой вершине есть противоречие.

Изучается также сложность вычисления *выполнимых* цейтинских формул [12, 8, 9] в достаточно слабых моделях вычислений.

Пусть $\mathrm{T}(G, c)$ — невыполнимая цейтинская формула, а $\mathrm{T}(G, c')$ — выполнимая цейтинская формула на том же графе G. В данной работе мы интересуемся тем, как связаны три меры сложности: размер опровержения $\mathrm{T}(G, c)$ (в резолюционной или другой системе доказательств), сложность вычисления $\mathrm{SearchVertex}(G, c)$, сложность вычисления значения $\mathrm{T}(G, c')$.

1.4. Древовидная резолюция и дерево решений

Рассмотрим в качестве системы доказательств древовидную резолюцию, а в качестве модели вычислений — дерево решений.

Дерево решений — это способ представления функций вида $f\colon \{0,1\}^n \to Z$. Оно представляет собой бинарное ориентированное от корня дерево, каждый лист которого помечен элементом из Z, остальные узлы помечены переменными и имеют исходящую степень два, одно из рёбер помечено нулём, а другое — единицей. Вычисление функции заключается в спуске по дереву от корня до листа в соответствии со значениями переменных. Мерой сложности дерева решений является размер, то есть число узлов в нём. Минимальный размер

дерева решений, вычисляющего функцию (отношение) f, обозначим за $\mathrm{Tree}(f)$.

Как упоминалось выше, $S_{\mathrm{T}}(\mathrm{T}(G,c)) = \mathrm{Tree}(\mathrm{Search}_{\mathrm{T}(G,c)})$ и $\mathrm{Tree}(\mathrm{Search}_{\mathrm{T}(G,c)}) \geqslant \mathrm{Tree}(\mathrm{SearchVertex}(G,c))$; однако, в случае дерева решений можно заметить, что сложности задач Search и SearchVertex совпадают: в любом листе мы имеем полную информацию о значениях переменных рёбер, инцидентных опровергнутой вершине, поэтому мы знаем и опровергнутый дизъюнкт. Итак, $S_{\mathrm{T}}(\mathrm{T}(G,c)) = \mathrm{Tree}(\mathrm{Search}_{\mathrm{T}(G,c)}) = \mathrm{Tree}(\mathrm{SearchVertex}(G,c))$.

Однако оказывается, что сложность вычисления выполнимой цейтинской формулы $\mathrm{T}(G,c')$ может быть значительно больше этой величины. Рассмотрим семейство графов P_n, где P_n получен из графа-пути на n рёбрах удвоением рёбер. Размер формулы $\mathrm{T}(P_n,c_n)$ есть $\mathcal{O}(n)$.

Если эта формула выполнима (например, все пометки в вершинах равны нулю), то у неё существует 2^n выполняющих наборов, никакие два из которых не могут приводить в один лист дерева решений (чтобы выдать ответ «1», мы должны спросить все переменные). Значит, $\mathrm{Tree}(\mathrm{T}(P_n,c'_n)) \geqslant 2^n$.

С другой стороны, решая задачу $\mathrm{SearchVertex}(P_n,c_n)$ для невыполнимой $\mathrm{T}(P_n,c_n)$, сначала проведём расщепление по значениям двух рёбер посередине пути, а затем рекурсивно обработаем невыполнимую из двух половин. Рекуррентное соотношение на размер $S(n) \leqslant 4S(n/2)+c$ имеет решение $S(n)=\mathcal{O}(n^2)$, таким образом, $S_{\mathrm{T}}(\mathrm{T}(P_n,c_n)) = \mathrm{Tree}(\mathrm{SearchVertex}(P_n,c_n)) = \mathcal{O}(n^2)$.

1.5. Регулярная резолюция и 1-BP

Перейдём к рассмотрению регулярной резолюционной системы доказательств, а в качестве модели вычислений рассмотрим однопроходную ветвящуюся программу.

Диаграмма принятия решений (ветвящаяся программа, BP) — это способ представления функций вида $f\colon \{0,1\}^n \to Z$. Диаграмма представляет собой ориентированный граф без циклов, каждый сток кото-

рого помечен элементом из Z, остальные узлы помечены переменными и имеют исходящую степень два, одно из рёбер помечено нулём, а другое — единицей. Вычисление функции заключается в спуске по графу от истока (обычно он один) до стока в соответствии со значениями переменных. Мерой сложности диаграммы является размер, то есть число узлов в ней. Ветвящаяся программа называется *однопроходной*, если на любом пути в ней все переменные различны. Будем обозначать как 1-BP(f) минимальный размер 1-BP, вычисляющей функцию (отношение) f.

Пусть дано минимальное регулярное резолюционное опровержение формулы $\mathrm{T}(G, c)$. Как обсуждалось выше, $S_{\mathrm{Reg}}(\mathrm{T}(G, c)) = \text{1-BP}(\mathrm{Search}_{\mathrm{T}(G,c)})$, и, кроме того, $\text{1-BP}(\mathrm{Search}_{\mathrm{T}(G,c)}) \geqslant \text{1-BP}(\mathrm{SearchVertex}(G, c))$.

В данной работе мы задаёмся вопросом, насколько ситуация для модели 1-BP отличается от ситуации для модели дерева решений: можем ли мы оценить сверху сложность вычисления выполнимой цейтинской формулы $\mathrm{T}(G, c')$ с помощью 1-BP, зная размер регулярного резолюционного опровержения $S_{\mathrm{Reg}}(\mathrm{T}(G, c))$? Иными словами, можно ли получать нижние оценки на $S_{\mathrm{Reg}}(\mathrm{T}(G, c))$, доказывая нижние оценки на $\text{1-BP}(\mathrm{T}(G, c'))$?

С другой стороны, мы знаем, что $S_{\mathrm{Reg}}(\mathrm{T}(G, c)) \geqslant \text{1-BP}(\mathrm{SearchVertex}(G, c))$. Можно ли получить верхнюю оценку на $S_{\mathrm{Reg}}(\mathrm{T}(G, c))$, зная $\text{1-BP}(\mathrm{SearchVertex}(G, c))$? К сожалению, можно показать, что полиномиальную верхнюю оценку построить не удастся.

Рассмотрим семейство полных графов $G_n = K_{\log n}$ на $\log n$ вершинах. Размер невыполнимой формулы $\mathrm{T}(G_n, c_n)$ будет равен $\mathcal{O}(\log n \cdot 2^{\log n}) = \mathcal{O}(n \log n)$. Вершину с нарушенным условием чётности можно найти, последовательно расщепляясь по значениям всех рёбер, но запоминая на уровне программы только текущую чётность суммы для каждой из вершин; тогда узлов на каждом уровне не более $\mathcal{O}(2^{\log n}) = \mathcal{O}(n)$, а уровней $\mathcal{O}(\log^2 n)$, поэтому $\text{1-BP}(\mathrm{SearchVertex}(G_n, c_n)) \leqslant \mathcal{O}(n \log^2 n)$. С другой стороны, используя теорему Бен-Сассона и Вигдерсона [3] о связи ширины и размера резолюции, несложно понять, что $S_{\mathrm{Reg}}(\mathrm{T}(G_n, c_n)) \geqslant$

$\Omega(2^{\log^2 n}) = \Omega(n^{\log n})$ (подробнее см., например, [4]).

В связи с этим мы ослабляем требования на систему доказательств, для которой мы хотим получить верхнюю оценку по 1-BP(SearchVertex(G, c)). А именно, будем строить доказательство в системе Фреге константной глубины [15].

1.6. Постановка задачи

Итак, в данной работе мы хотим решить две задачи. Пусть $\mathrm{T}(G, c)$ невыполнима, а $\mathrm{T}(G, c')$ выполнима.

1. Получить верхнюю оценку на минимальный размер 1-BP, вычисляющей $\mathrm{T}(G, c')$, по данному размеру регулярного резолюционного опровержения $\mathrm{T}(G, c)$.

2. Получить верхнюю оценку на минимальный размер опровержения формулы $\mathrm{T}(G, c')$ в системе Фреге константной глубины по данному размеру 1-BP, вычисляющей SearchVertex(G, c).

1.7. Результаты и структура работы

Основная часть работы имеет следующую структуру. Пусть G — связный граф, $\mathrm{T}(G, c)$ невыполнима, а $\mathrm{T}(G, c')$ выполнима.

1. В разделе 3 мы описываем структуру минимальной 1-BP, вычисляющей SearchVertex(G, c): в каждом узле s вычисляется SearchVertex(G_s, c_s), где $\mathrm{T}(G_s, c_s)$ — единственная невыполнимая компонента формулы $\mathrm{T}(G, c)$ после подстановки, соответствующей любому пути от истока диаграммы до s.

   Отметим, что аналогичное утверждение уже было известно про минимальные 1-BP, вычисляющие выполнимые цейтинские формулы [8]. Однако, в случае вычисления SearchVertex доказательство получается гораздо более сложным.

2. В разделе 4 по минимальной 1-BP размера S, вычисляющей SearchVertex(G, c), мы строим 1-BP, вычисляющую выполнимую

формулу $\mathrm{T}(G, c')$, и размер полученной программы будет не более $S^{\mathcal{O}(\log|V(G)|)}$. В доказательстве мы пользуемся доказанной теоремой 1 о структурной характеризации: у 1-BP, решающих эти две задачи, схожая структура, что позволяет нам перестроить одну из них в другую (пусть и с увеличением размера).

Тем самым, мы доказываем следующую теорему:

Теорема (Теорема 2)**.** *Пусть* $\mathrm{T}(G, c)$ *— невыполнимая цейтинская формула на графе* $G(V, E)$*. Если существует регулярное резолюционное опровержение* $\mathrm{T}(G, c)$ *размера* S*, то для любой функции пометок* c' *такой, что* $\mathrm{T}(G, c')$ *выполнима, существует* 1-BP*, вычисляющая* $\mathrm{T}(G, c')$*, размер которой* $S^{\mathcal{O}(\log|V|)}$*.*

3. В разделе 5 мы доказываем следующую теорему:

   Теорема (Теорема 4)**.** *Пусть* S *— размер минимальной* 1-BP*, вычисляющей* $\mathrm{SearchVertex}(G, c)$*, где* $G(V, E)$ *— связный граф. Тогда существует вывод формулы* $\neg\mathrm{T}(G, c)$ *в системе Фреге, размер которого не больше* $\mathcal{O}\left(S|\mathrm{T}(G, c)|^4\right)$*, а глубина не больше* $\mathcal{O}(1)$*.*

   В доказательстве мы снова пользуемся теоремой 1 о структурной характеризации минимальной 1-BP, вычисляющей $\mathrm{SearchVertex}(G, c)$. Мы рассматриваем все узлы программы в обратном топологическом порядке и для каждого узла s строим вывод формулы $\neg\mathrm{T}(G_s, c_s)$.

2. Основные определения

Пропозициональная формула ϕ может быть представлена как дерево, внутренние узлы которого помечены операциями, а каждый лист помечен переменной или логической константной. *Размер* формулы $|\phi|$ — число вершин в таком дереве. *Глубина* формулы — максимальное число смен операции на пути от корня к листу, увеличенное на единицу. Например, глубина литерала равняется единице, конъюнкция литералов имеет глубину не более двух, а глубина ДНФ-формулы не больше трёх.

Формула ψ *семантически следует* из множества формул Γ, если любая подстановка, выполняющая все формулы из Γ, также выполняет и формулу ψ.

2.1. Резолюционные системы доказательств

Правило резолюции позволяет из дизъюнктов $x \vee A$ и $\neg x \vee B$ вывести дизъюнкт $A \vee B$.

Резолюционное доказательство формулы ϕ — последовательность дизъюнктов $C_1, \ldots, C_l$ такая, что:

- $C_l = \square$ — пустой дизъюнкт, то есть тождественно ложный;
- каждый C_i получен одним из двух способов:
  - C_i является дизъюнктом ϕ;
  - C_i получен по правилу резолюции из C_j и C_k для некоторых $j, k < i$.

Размер доказательства — количество дизъюнктов в нём (l).

Резолюционное доказательство называется *древовидным*, если каждый выведенный дизъюнкт используется затем не более одного раза. Резолюционное доказательство называется *регулярным*, если в графе доказательства на любом пути по каждой переменной резолюция происходила не больше одного раза.

2.2. Диаграммы принятия решений

Диаграмма принятия решений (или *ветвящаяся программа*) — ориентированный граф без циклов, в каждом стоке записана константа из множества значений Z, в остальных (внутренних) вершинах записана переменная. Из каждой внутренней вершины с переменной x выходит два ребра, помеченные 0 и 1. Если у диаграммы один исток, то она вычисляет функцию $f\colon \{0,1\}^n \to Z$. Чтобы вычислить значение на данной подстановке, необходимо пройти от истока до стока, выбирая рёбра, отвечающие значению переменной в вершине; в стоке будет записан ответ.

Ветвящаяся программа называется *деревом решений*, если граф представляет собой ориентированное от корня дерево. Ветвящаяся программа называется *однопроходной* (1-BP), если на любом пути все переменные различны.

Пусть $\phi(\vec{x})$ — невыполнимая формула в КНФ с дизъюнктами $C_1, \ldots, C_m$. Отношением Search_ϕ будем называть $\{(\vec{x}, i) \mid C_i(\vec{x}) = 0\}$. Вычислить отношение — значит по входу $\vec{x}$ найти i такое, что $(\vec{x}, i) \in \mathrm{Search}_\phi$.

Лемма 1 ([17]). *Пусть ϕ — невыполнимая формула в КНФ. Тогда минимальный размер регулярного резолюционного доказательства формулы ϕ равен минимальному размеру* 1-BP, *которая вычисляет* Search_ϕ.

2.3. Цейтинские формулы

Цейтинская формула $\mathrm{T}(G, c)$ строится по неориентированному графу $G(V, E)$ и функции $c\colon V \to \{0,1\}$: каждому ребру $\{v, u\} \in E$ ставится в соответствие пропозициональная переменная $x_{vu} \equiv x_{uv}$, каждая вершина v задаёт ограничение «сумма на рёбрах, инцидентных v, равна $c(v)$»:

$$c(v) = \bigoplus_{u:\{vu\}\in E(G)} x_{vu},$$

далее каждое ограничение записывается в КНФ, и $\mathrm{T}(G, f)$ — конъюнкция этих ограничений.

Иногда мы будем писать $\mathrm{T}(G, f)$, где $f\colon V' \to \{0,1\}$ и $V' \supseteq V$. Под этим мы будем неявно понимать формулу $\mathrm{T}(G, f|_V)$.

Пусть H — компонента связности G. Будем говорить, что H — *выполнимая компонента* формулы $\mathrm{T}(G, c)$, если формула $\mathrm{T}(H, c)$ выполнима. В противном случае будем говорить, что H — *невыполнимая компонента* формулы $\mathrm{T}(G, c)$.

Пусть $V' \subseteq V$ — подмножество вершин. Мы будем говорить, что ребро e *инцидентно* V', если существует хотя бы одна вершина $v \in V'$ такая, что e инцидентно v.

Лемма 2 ([20])**.** *Цейтинская формула* $\mathrm{T}(G, f)$ *выполнима тогда и только тогда, когда для каждой компоненты связности* S *графа* G *выполняется* $0 = \bigoplus_{v \in S} f(v)$.

Лемма 3 ([14], Lemma 2.3)**.** *Пусть* $G(V, E)$ — *связный граф, а* $c\colon V \to \{0,1\}$ — *функция пометок. Пусть* $U \subsetneq V$*, а* $\Phi = \bigwedge_{v \in U} P_v$ — *конъюнкция условий чётности для всех вершин из* U*. Тогда* Φ *выполнима.*

Лемма 4. *Результат подстановки* $x_e := b$ *(*$b \in \{0,1\}$*) в формулу* $\mathrm{T}(G, c)$ — *цейтинская формула* $\mathrm{T}(G', c')$*, где* $G' = G - e$*, а* c' *отличается от* c *на концах ребра* e *на* b *и совпадает с* c *на остальных вершинах.*

Доказательство. Проверяется непосредственной проверкой. □

Лемма 5. *Пусть* $G(V, E)$ — *связный граф, и пусть* $c_1, c_2\colon V \to \{0,1\}$ — *функции пометок. Если цейтинские формулы* $\mathrm{T}(G, c_1)$ *и* $\mathrm{T}(G, c_2)$ *обе выполнимые или обе невыполнимые, то одну можно получить из другой заменой некоторых переменных на их отрицания.*

Доказательство. Замена x_e на $\neg x_e$ в цейтинской формуле соответствует инвертированию пометок концов ребра e. Поскольку G связен и $\mathrm{T}(G, c_1)$ и $\mathrm{T}(G, c_2)$ обе выполнимые или обе невыполнимые, то, по лемме 2, у функций пометок c_1 и c_2 чётное число различий. Пусть $v_1, v_2, \ldots, v_{2k}$ — вершины, на которых c_1 и c_2 различаются. Пусть p_i — простой путь, соединяющий v_{2i-1} и v_{2i}, для всех $i \in [k]$. Изменим $\mathrm{T}(G, c_1)$ следующим образом: для из путей $p_1, \ldots, p_k$ заменим переменные, соответствующие рёбрам на пути, на их отрицания (если несколько путей

проходят через ребро, мы делаем такую замену для каждого из путей). Получившаяся формула — это $\mathrm{T}(G, c_2)$, поскольку пометки концов путей (то есть вершин $v_1, \ldots, v_{2k}$) изменились, а пометки всех остальных вершин — нет. $\square$

3. Структура 1-BP, вычисляющих $\mathrm{SearchVertex}(G, c)$

3.1. Структурированные ветвящиеся программы

В этом разделе мы хотим показать, что минимальная 1-BP, вычисляющая $\mathrm{SearchVertex}(G, c)$, имеет строгую структуру. А именно, каждый узел 1-BP решает задачу SearchVertex, но для другого графа и функции пометок.

Определение структурированных ветвящихся программ, вычисляющих *выполнимые* цейтинские формулы, в основном результате этого раздела нам не понадобится, однако, нам будет удобно дать его здесь же. Свойства таких программ, описанные в предложении 1, понадобятся нам в разделе 4.

Определение 1. Будем называть ветвящуюся программу D *структурированной* ветвящейся программой, вычисляющей выполнимые цейтинские формулы, если выполняются все следующие свойства:

- У D два стока: один помечен 0 и один помечен 1.
- Существует конечное множество вершин V и отображение μ, определённое на всех узлах D, за исключением 0-стока, переводящее узел s в пару (G_s, c_s), где $G_s(V, E_s)$ — граф на множестве вершин V, а $c_s \colon V \to \{0, 1\}$ — функция пометок такая, что формула $\mathrm{T}(G_s, c_s)$ выполнима. Каждый внутренний узел s помечен переменной x_e, где $e \in E_s$.
- (Свойство стока) $\mu(\text{1-сток}) = (G_\varnothing, \mathbf{0})$, где $G_\varnothing(V, \varnothing)$ — граф без рёбер, а $\mathbf{0}$ — функция, тождественно равная нулю.
- (Локальное свойство) Пусть s — узел, помеченный x_e, а s_i — конец идущего из s ребра с меткой i для $i \in \{0, 1\}$. Пусть функция пометок c_0 равна c_s, а c_1 получена из c_s инвертированием пометок концов e.

– Если e не мост G_s, то $G_{s_0} = G_{s_1} = G_s - e$, $c_{s_0} = c_0$ и $c_{s_1} = c_1$.

– Если e — мост G_s, обозначим за V_A множество вершин компоненты связности $G_s - e$, содержащей один из концов e. Пусть $\gamma = \sum_{v \in V_A} c_s(v)$. (Поскольку $\mathrm{T}(G_s, c_s)$ выполнима, то, по лемме 2, γ не зависит от выбора компоненты V_A.)

Тогда $G_{s_\gamma} = G_s - e$, $c_{s_\gamma} = c_\gamma$, а $s_{1-\gamma}$ — 0-сток.

В предположении, что у D есть единственный исток r, и $\mu(r) = (G, c)$, в предложении 1 мы покажем, что такая D действительно вычисляет $\mathrm{T}(G, c)$.

Определим теперь понятие структурированных ветвящихся программ, вычисляющих SearchVertex.

Определение 2. Пусть $G(V, E)$ — *связный* граф, а формула $\mathrm{T}(G, c)$ невыполнима. Пусть D — ветвящаяся программа с единственным истоком. Будем называть D *структурированной* ветвящейся программой, вычисляющей SearchVertex(G, c), если выполняются все следующие свойства:

- У D ровно $|V|$ стоков, которые помечены различными элементами V.

- Существует отображение ν из узлов D, переводящее каждый узел s в пару (G_s, c_s), где $G_s(V_s, E_s)$ — *связный* подграф графа G, а $c_s \colon V_s \to \{0, 1\}$ функция пометок такая, что формула $\mathrm{T}(G_s, c_s)$ невыполнима. Каждый внутренний узел s помечен переменной x_e для некоторого ребра $e \in E_s$. Отображение ν переводит исток в пару (G, c).

- (Свойство стока) Сток, помеченный v, отображение ν переводит в граф с единственной вершиной v и функцией пометок, равной 1 на v.

- (Локальное свойство) Пусть внутренний узел s помечен переменной x_e, а s_i — конец идущего из s ребра с меткой i для $i \in \{0, 1\}$.

Пусть функция пометок c_0 равна c_s, а c_1 получена из c_s инвертированием пометок концов e.

- Если e не мост G_s, то $G_{s_0} = G_{s_1} = G - e$, $c_{s_0} = c_0$ и $c_{s_1} = c_1$.
- Если e — мост G_s, то $G_s - e$ представляется как дизъюнктное объединение двух связных подграфов: $A(V_A, E_A)$ и $B(V_B, E_B)$. Пусть $\gamma = \sum_{v \in V_A} c_s(v)$. Тогда $G_{s_\gamma} = B$, функция c_{s_γ} равна функции c_γ, суженной на V_B; $G_{s_{1-\gamma}} = A$, и функция $c_{s_{1-\gamma}}$ равна функции $c_{1-\gamma}$, суженной на V_A.

Следующее предложение 1 показывает корректность определения, то есть то, что такая D действительно вычисляет $\mathrm{SearchVertex}(G, c)$.

Предложение 1. *1. Если D — структурированная ветвящаяся программа, вычисляющая выполнимые цейтинские формулы, то:*

a) D является однопроходной;

b) каждый узел s диаграммы D, за исключением 0-стока, вычисляет $\mathrm{T}(G_s, c_s)$, где $(G_s, c_s) = \mu(s)$.

2. Если D — структурированная ветвящаяся программа, вычисляющая $\mathrm{SearchVertex}(G, c)$, то:

a) D является однопроходной;

b) каждый узел s диаграммы D вычисляет $\mathrm{SearchVertex}(G_s, c_s)$, где $(G_s, c_s) = \nu(s)$. В частности, исток D вычисляет $\mathrm{SearchVertex}(G, c)$.

Доказательство. a) Доказательство однопроходности общее для обоих случаев. Рассмотрим путь из узла s во внутренний узел $t \neq s$. Пусть s помечен переменной x_e, а t помечен $x_{e'}$. Из локального свойства следует, что G_t — подграф G_s, и e не принадлежит G_t. Поскольку e' принадлежит G_t, то $e \neq e'$, то есть s и t помечены разными переменными.

b) В обоих случаях проведём доказательство по индукции по номеру узла в обратном топологическом порядке. База индукции: утверждение для стоков следует из свойства стока.

Переход. Пусть внутренний узел s помечен переменной x_e. Если e не мост G_s, то утверждение для s следует из локального свойства для s и предположению индукции для непосредственных последователей s.

Пусть теперь e — мост G_s. Пусть s_0 и s_1 — непосредственные последователи s, где ребро (s, s_i) помечено i для $i \in \{0, 1\}$.

1. Вычисление выполнимой цейтинской формулы. Пусть A — одна из двух компонент связности $G_s - e$, содержащих один из концов e. Пусть V_A — множество вершин A, и пусть $\gamma = \sum_{v \in V_A} c_s(v)$. Пусть σ — выполняющий набор формулы $\mathrm{T}(G_s, c_s)$. Рассмотрим сумму $\sum_{v \in V_A} \sum_{j \in I_s(v)} \sigma(x_j)$, где $I_s(v)$ — множество рёбер G_s, инцидентных вершине v. Поскольку σ выполняет $\mathrm{T}(G_s, c_s)$, эта сумма равна $\sum_{v \in V_A} c_s(v) = \gamma$; с другой стороны, эта сумма равна $\sigma(x_e)$, поскольку $\sigma(x_e)$ встречается в этой сумме один раз, а все остальные переменные — дважды. Следовательно, не существует выполняющих наборов $\mathrm{T}(G_s, c_s)$, присваивающих переменной x_e значение $1 - \gamma$. По предположению индукции, узел s_γ вычисляет $\mathrm{T}(G_{s_\gamma}, c_{s_\gamma})$, где $G_{s_\gamma} = G_s - e$ и c_{s_a} отличается от c_s на γ на концах ребра e, поэтому s вычисляет $\mathrm{T}(G_s, c_s)$.

2. Вычисление SearchVertex. По предположению индукции, s вычисляет $\mathrm{SearchVertex}(G_{s_0}, c_{s_0})$ при $x_e = 0$ и $\mathrm{SearchVertex}(G_{s_1}, c_{s_1})$ при $x_e = 1$. Рассмотрим подстановку σ, и пусть s возвращает вершину v на σ. Обозначим $a \coloneqq \sigma(x_e)$, тогда v — вершина G_{s_a}. Если v не инцидентна e, то σ опровергает условие чётности вершины v в $\mathrm{T}(G_s, c_s)$, поскольку множества инцидентных v рёбер в графе G_s и графе G_{s_a} совпадают. Если v инцидентна e, рассмотрим сумму $\sum_{j \in I_{s_a}(v)} \sigma(x_j)$, где $I_{s_a}(v)$ — множество рёбер в G_{s_a}, инцидентных v. Эта сумма равна $1 + c_{s_a}(v)$, поскольку по индукционному предположению σ опровергает условие чётности вершины v в $\mathrm{T}(G_{s_a}, c_{s_a})$. По локальному свойству, $c_{s_a}(v) = c_s(v) + a$. Следовательно, σ опровергает условие чётности для v в $\mathrm{T}(G_s, c_s)$.

□

3.2. Структурная теорема

Будем говорить, что 1-ВР D *локально минимальная* из удовлетворяющих некоторому свойству P, если для любого внутреннего узла s и любого его непосредственного последователя t выполнено следующее: если все входящие в s рёбра перенаправить в t и удалить s, то получившаяся 1-ВР D' уже не удовлетворяет свойству P.

В этом параграфе мы доказываем следующую теорему.

Теорема 1. *Пусть $G(V, E)$ — связный граф, а функция пометок c такова, что $\mathrm{T}(G, c)$ невыполнима. Пусть D — локально минимальная* 1-ВР*, вычисляющая* $\mathrm{SearchVertex}(G, c)$*. Тогда D является структурированной ветвящейся программой, вычисляющей* $\mathrm{SearchVertex}(G, c)$*.*

Более того, пусть s — произвольный узел D, и пусть $\nu(s) = (H, f)$. Тогда для любой частичной подстановки α, соответствующей какому-нибудь пути от истока D до s, H является единственной невыполнимой компонентой формулы $\mathrm{T}(G, c)|_\alpha$, а f является сужением функции пометок формулы $\mathrm{T}(G, c)|_\alpha$ на множество вершин H.

Пусть D — 1-ВР, вычисляющая $\mathrm{SearchVertex}(G, c)$, где $G(V, E)$ — связный граф, и $\mathrm{T}(G, c)$ невыполнима. Для каждого внутреннего узла s программы D обозначим за $h(s)$ множество пометок тех стоков, которые достижимы из s. Таким образом, $h(s)$ — множество вершин G, которые являются возможными результатами функции, вычисляемой s. Обозначим за $P(s)$ множество частичных подстановок, соответствующих путям из истока D в s.

План доказательства теоремы 1 следующий.

1. Ключевой идеей доказательства является наблюдение за множеством $h(s)$, определённым выше. Предложение 2 показывает, что для каждого узла s все частичные подстановки из $P(s)$ изменяют пометки вершин из $h(s)$ одинаковым образом.

2. В главной технической части доказательства мы показываем, что если D — локально минимальная 1-ВР, то для каждого узла s и для любой $\alpha \in P(s)$ формула $\mathrm{T}(G, c)|_\alpha$ содержит единственную

невыполнимую компоненту связности, а множество вершин этой компоненты есть в точности $h(s)$. Чтобы это показать, мы доказываем несколько промежуточных утверждений:

(a) Предложение 3 показывает, что если D — локально минимальная 1-ВР, то для любого её внутреннего узла s, помеченного переменной x_e, ребро e будет инцидентно множеству $h(s)$.

(b) Предложение 4 показывает, что если D — локально минимальная 1-ВР, то для любого её внутреннего узла s, помеченного переменной x_e, ребро e принадлежит невыполнимой компоненте формулы $\mathrm{T}(G, c)|_\alpha$, которая содержится в $h(s)$ для всех $\alpha \in P(s)$ (по предложению 2, эта компонента не зависит от α).

(c) Предложение 5 показывает, что если D — локально минимальная 1-ВР, то для любого узла s $h(s)$ является множеством всех вершин всех (одной или нескольких) невыполнимых компонент $\mathrm{T}(G, c)|_\alpha$ для всех $\alpha \in P(s)$.

(d) Предложение 6 завершает доказательство этого пункта.

3. Мы доказываем теорему 1, пользуясь результатами пунктов 1 и 2.

Предложение 2. *Пусть s — внутренний узел D. Пусть α_1 и α_2 — частичные подстановки из $P(s)$. Тогда выполняется:*

1. *Для каждого ребра $e \in E$, инцидентного $h(s)$, α_1 присваивает значение переменной x_e тогда и только тогда, когда α_2 присваивает значение этой переменной.*

2. *Для каждой вершины $v \in h(s)$ её пометка в $\mathrm{T}(G, c)|_{\alpha_1}$ равна её пометке в $\mathrm{T}(G, c)|_{\alpha_2}$.*

Доказательство. Рассмотрим вершину $v \in h(s)$ и какой-нибудь достижимый из s сток $t \in D$, помеченный v. Пусть β — частичная подстановка, соответствующая пути из s в t. Заметим, что область определения

β не пересекается с областью определения α_i для каждого $i \in \{1, 2\}$, поскольку D является 1-ВР. Определим $\rho_i = \alpha_i \cup \beta$ для $i \in \{1, 2\}$.

Обе подстановки ρ_1 и ρ_2 опровергают условие чётности вершины v. Значит, для любого инцидентного вершине v ребра e значение x_e присвоено обеими подстановками ρ_1 и ρ_2. Значит, α_1 и α_2 присваивают значения одинаковому подмножеству переменных из $\{x_e \mid e \text{ инцидентно } v\}$, и суммы значений α_1 и α_2 на этих переменных равны по модулю 2. □

Предложение 3. *Пусть D — локально минимальная 1-ВР, вычисляющая* SearchVertex(G, c). *Тогда любой её внутренний узел s помечен ребром, инцидентным $h(s)$.*

Доказательство. Предположим, что для внутреннего узла s, помеченного x_e, утверждение неверно, то есть e соединяет две вершины вне $h(s)$. Пусть t_0 и t_1 — непосредственные последователи s, причём ребро (s, t_i) помечено i.

Рассмотрим D', получаемую из D следующим образом: удалим ребро (s, t_0) и стянем (s, t_1). Получившуюся в результате стягивания вершину обозначим за s' и пометим её той же переменной, которой был помечен узел t_1.

Мы утверждаем, что D' тоже вычисляет SearchVertex(G, c). Рассмотрим полную подстановку β. Пусть $\beta'(x_q) = \beta(x_q)$ для $q \neq e$ и $\beta'(x_e) = 1$.

Если путь в D, соответствующий β, не проходит через s, то в точности такой же путь и с такими же пометками есть в D', поэтому $D'(\beta) = D(\beta)$. Пусть теперь путь в D, соответствующий β, проходит через узел s. В этом случае путь в D, соответствующий β', тоже проходит через s, так как на любом пути от истока до s только s помечена переменной x_e. Тогда $D(\beta') \in h(s)$. Ребро e не инцидентно вершинам из $h(s)$, поэтому e не инцидентно вершине $D(\beta')$. Поскольку условие чётности вершины $D(\beta')$ опровергается β' и e не инцидентно $D(\beta')$, то $D(\beta')$ опровергается и подстановкой β. По построению диаграммы D' выполнено $D(\beta') = D'(\beta)$. Таким образом, β опровергает условие чётности вершины $D'(\beta)$, а значит, D' корректно вычисляет SearchVertex(G, c) —

противоречие с локальной минимальностью D. $\square$

По лемме 4, при подстановке в цейтинскую формулу значения переменной получается снова цейтинская формула. Для произвольной частичной подстановки α из $P(s)$ обозначим за $G_{s,\alpha}$ граф и за $c_{s,\alpha}$ функцию пометок такие, что $\mathrm{T}(G,c)|_\alpha$ является в точности $\mathrm{T}(G_{s,\alpha}, c_{s,\alpha})$.

Заметим, что если для какой-то $\alpha \in P(s)$ C является невыполнимой компонентой $\mathrm{T}(G_{s,\alpha}, c_{s,\alpha})$, и все вершины C лежат в $h(s)$, то по предложению 2 C является невыполнимой компонентой для всех подстановок из $P(s)$. Обозначим за $U(s)$ множество невыполнимых компонент $\mathrm{T}(G_{s,\alpha}, c_{s,\alpha})$, содержащихся в $h(s)$, где $\alpha \in P(s)$. Как было замечено, $U(s)$ не зависит от выбора α.

Определение 3. Пусть $\alpha \in P(s)$. Пусть $H(V_H, E_H)$ компонента связности $G_{s,\alpha}$, содержащая хотя бы одну вершину из $h(s)$. Тогда выполняется один из трёх случаев (три типа компоненты H для узла s и частичной подстановки α):

(1) $V_H \subseteq h(s)$ и H — невыполнимая компонента $\mathrm{T}(G_{s,\alpha}, c_{s,\alpha})$. Иными словами, $H \in U(s)$;

(2) $V_H \subseteq h(s)$ и H — выполнимая компонента $\mathrm{T}(G_{s,\alpha}, c_{s,\alpha})$;

(3) $V_H \not\subseteq h(s)$.

Предложение 4. *Пусть D —* локально минимальная 1-ВР*, вычисляющая* SearchVertex(G,c)*. Тогда любой её внутренний узел s помечен переменной x_e такой, что e соединяет две вершины компоненты связности из $U(s)$.*

Доказательство. По предложению 3, для любого узла l диаграммы D, если l помечен x_e, то e инцидентно $h(l)$.

Рассуждаем от противного. Выберем самый глубокий (то есть самый далёкий от истока) узел s, в котором доказываемое утверждение неверно. Пусть s помечен x_e. Обозначим за t_0 и t_1 его непосредственных последователей, причём ребро (s,t_i) помечено i для каждого $i \in \{0,1\}$.

Пусть α — частичная подстановка, соответствующая пути от истока до s.

Пусть $C(V_C, E_C)$ — компонента связности $G_{s,\alpha}$, содержащая ребро e. Поскольку в узле s утверждение не выполнено, либо C — выполнимая компонента $G_{s,\alpha}$, либо C имеет хотя бы одну вершину вне $h(s)$.

Рассмотрим частичную подстановку θ, выполняющую условия чётности всех вершин из $V_C \cap h(s)$, но не присваивающую значения переменным, соответствующим рёбрам $G_{s,\alpha}$ за пределами C. Если C выполнима, то θ существует по определению; если C имеет хотя бы одну вершину вне $h(s)$, то такая θ существует по лемме 3).

Утверждение 1. Рассмотрим любой путь $\tau = (\tau_1, \ldots, \tau_m)$ в D из $t_{\theta(x_e)}$ в сток. Тогда для любой пометки $x_{e'}$ узла из τ, ребро e' не инцидентно C.

Доказательство. Рассуждаем от противного, пусть существует узел из τ, для которого условие нарушено. Пусть $i \in [m]$ — минимальный индекс, для которого узел τ_i помечен $x_{e'}$, и ребро e' инцидентно C.

Поскольку τ_i последователь s, $h(\tau_i) \subseteq h(s)$. По предложению 3, ребро e' инцидентно $h(\tau_i)$. Если e' не содержится в невыполнимой компоненте внутри $h(\tau_i)$, то мы получаем противоречие с тем, что s было самым глубоким узлом, нарушающим условие предложения 4.

Пусть тогда e' содержится в невыполнимой компоненте $C' \subseteq h(\tau_i)$. Как обсуждалось выше, структура такой компоненты не зависит от выбранного пути от истока до τ_i, поэтому мы можем выбрать путь, который согласован с α на пути от истока до s, а затем идёт через $\tau_1, \ldots, \tau_i$. Пусть μ — частичная подстановка, соответствующая такому пути. μ расширяет α, поэтому граф $G_{\tau_i,\mu}$ является подграфом $G_{s,\alpha}$. C — компонента связности $G_{s,\alpha}$. C и C' имеют общее ребро e', поэтому C' является подграфом C. Подстановка θ выполняет цейтинскую формулу, соответствующую компоненте связности C и функции пометок $c_{s,\alpha}$. Более того, у областей определения θ и μ есть только одна общая переменная. Эта переменная — x_e, и μ согласовано с θ на x_e по построению. Следовательно, μ можно достроить до полной подстановки, согласован-

ной с θ. Но эта полная подстановка выполняет условия чётности всех вершин из C', что противоречит невыполнимости C' в узле τ_i. □

Дальнейшее рассуждение похоже на доказательство утверждения 3. Рассмотрим диаграмму D', которая получается из D удалением ребра $(s, t_{1-\theta(e)})$ и стягиванием ребра $(s, t_{\theta(e)})$ (t_i определено выше для $i \in \{0, 1\}$). Поскольку D локально минимальная 1-ВР, вычисляющая SearchVertex(G, c), существует полная подстановка β такая, что условие чётности вершины $D'(\beta)$ не опровергнуто подстановкой β. Путь в D', соответствующий β, проходит через s, поскольку в противном случае $D'(\beta) = D(\beta)$, а условие чётности этой вершины опровергнуто β.

Пусть $\beta'(x_q) = \beta(x_q)$ для $q \neq e$, $\beta'(x_e) = \theta(x_e)$ и $v = D'(\beta)$.

Как и в доказательстве предложения 3, $D(\beta') = D'(\beta) = v$, и по корректности D, β' нарушает условие чётности v. С другой стороны, β не выполняет условие чётности v по выбору β. Поскольку β' и β отчитаются только на x_e, ребро e инцидентно v.

Так как v инцидентно e, а e соединяет две вершины из C, то $v \in C$. По утверждению 1, часть подстановки β', соответствующая пути от $t_{\theta(x_e)}$ до стока, не подставляет значения инцидентным C рёбрам, и поскольку β' нарушает условие чётности v, мы получаем, что v является листом в G_s. Но значение $\beta'(x_e)$ было выбрано в соответствии с подстановкой θ, выполняющей условия чётности всех вершин в $V_C \cap h(s) \ni v$, поэтому β' выполняет условие чётности вершины v — противоречие. □

Предложение 5. *Пусть* D локально минимальная 1-ВР*, вычисляющая* SearchVertex(G, c)*. Пусть* s *— узел* D*, а частичная подстановка* $\alpha \in P(s)$*. Тогда каждая вершина* v *из* $h(s)$ *принадлежит невыполнимой компоненте* $\mathrm{T}(G_{s,\alpha}, c_{s,\alpha})$*, лежащей в* $h(s)$*.*

Доказательство. Докажем утверждение по индукции по расстоянию d от s до самого далёкого стока, достижимого из s.

База индукции: $d = 0$, то есть s — сток. $h(s)$ состоит из единственной вершины v, условие чётности v нарушено подстановкой α, тогда компонента $\{v\}$ невыполнима.

Переход. Рассуждаем от противного, пусть v — вершина из $h(s)$, лежащая в компоненте связности $C(V_C, E_C)$ типа (2) или (3) (см. определение 3) для узла s и подстановки $\alpha \in P(s)$. Пусть t_0 и t_1 — непосредственные последователи s. Заметим, что $h(s) = h(t_0) \cup h(t_1)$ по определению h, поэтому существует $i \in \{0, 1\}$ такое, что $v \in h(t_i)$. Рассмотрим $\beta_i \in P(t_i)$, продолжающее α.

Пусть s помечено переменной x_e; по предложению 4 ребро e содержится в невыполнимой компоненте $\mathrm{T}(G, c)|_\alpha$ из $U(s)$, поэтому e не содержится в компоненте C. Если $V_C \setminus h(t_i) \neq \varnothing$, то v принадлежит компоненте связности типа (3) для узла t_i и подстановки β_i. Если $V_C \subseteq h(t_i)$, то, поскольку e не содержится в C, компонента связности C совпадает с соответствующей компонентой связности $\mathrm{T}(G, c)|_{\beta_i}$, содержащейся в $h(t_i)$, более того, пометки вершин C совпадают в формулах $\mathrm{T}(G, c)|_{\beta_i}$ и $\mathrm{T}(G, c)|_\alpha$. Таким образом, в этом случае v содержится в компоненте типа (2) для узла t_i и подстановки β_i.

Но все вершины $h(t_i)$ содержатся в компонентах типа (1) для узла t_i и подстановки β_i по индукционному предположению. Противоречие, следовательно все вершины $h(s)$ содержатся в невыполнимых компонентах. □

Предложение 6. *Пусть D — локально минимальная* 1-BP*, вычисляющая* SearchVertex(G, c)*, а граф G связен. Пусть s — узел D. Тогда $U(s)$ состоит из единственной компоненты связности с множеством вершин $h(s)$. Более того, для каждой подстановки $\alpha \in P(s)$, компонента из $U(s)$ является единственной невыполнимой компонентой $\mathrm{T}(G, c)|_\alpha$.*

Доказательство. Докажем утверждение по индукции по расстоянию d от истока до s.

База индукции: $d = 0$, то есть s исток D. Поскольку G связен, он состоит из единственной невыполнимой компоненты $\mathrm{T}(G, c)$. По лемме 3, для любой вершины $v \in V$ существует полная подстановка, нарушающая условие чётности v, но выполняющее условия чётности остальных вершин. Следовательно, $h(s) = V$.

Переход. Пусть α — частичная подстановка из $P(s)$. Пусть r — непо-

средственный предшественник s в пути, соответствующем α. Пусть частичная подстановка $\beta \in P(r)$ согласована с α. По предположению индукции, $U(r)$ состоит из одной компоненты связности $C(V_C, H_C)$ формулы $\mathrm{T}(G,c)|_\beta$ с множеством вершин $h(r)$. Пусть узел r помечен переменной x_e. По предложению 4, ребро e содержится в C. Рассмотрим два случая.

Если e не мост C, то при подстановке $x_e := \alpha(x_e)$ в $\mathrm{T}(G,c)|_\beta$ полученная формула (являющаяся $\mathrm{T}(G,c)|_\alpha$) имеет единственную невыполнимую компоненту $C - e$. По лемме 3, каждая вершина V_C может быть единственной вершиной с нарушенным условием чётности формулы $\mathrm{T}(G,c)|_\alpha$. Таким образом, $h(s) = h(r) = V_C$.

Пусть теперь e — мост C. Пусть A и B — компоненты связности $C-e$. В результате подстановки $x_e := \alpha(x_e)$ в $\mathrm{T}(G,c)|_\beta$ полученная формула (являющаяся $\mathrm{T}(G,c)|_\alpha$) имеет компоненты связности A и B вместо C. По лемме 2, в точности одна из компонент A и B невыполнима. Не умаляя общности, предположим, что A — невыполнимая компонента $\mathrm{T}(G,c)|_\alpha$. По лемме 3, каждая вершина A может быть единственной вершиной с нарушенным условием чётности, поэтому $h(s)$ содержит все вершины компоненты A. $h(s)$ не содержит вершин из B, поскольку по предложению 5 $h(s)$ содержит вершины только невыполнимых компонент. A — единственная невыполнимая компонента $\mathrm{T}(G,c)|_\alpha$, поскольку подстановка значения в x_e в формулу $\mathrm{T}(G,c)|_\beta$ не влияет на отличные от C компоненты. Таким образом, индукционный переход доказан. □

Доказательство теоремы 1. Рассмотрим произвольный узел s диаграммы D. По предложению 6, для каждой частичной подстановки $\alpha \in P(s)$ формула $\mathrm{T}(G_{s,\alpha}, c_{s,\alpha})$, являющаяся результатом подстановки α в $\mathrm{T}(G,c)$, имеет единственную невыполнимую компоненту $H \in U(s)$ с множеством вершин $h(s)$. Более того, по предложению 2, H и сужение $c_{s,\alpha}$ на вершины H не зависят от выбора α. Обозначим за f это сужение.

Зафиксируем $\alpha \in P(s)$ и рассмотрим произвольный путь из s в сток D. Пусть θ — частичная подстановка, соответствующая этому пути, а полная подстановка γ — объединение подстановок θ и α. Пусть $v = D(\gamma)$.

Тогда γ нарушает условие чётности в вершине v формулы $\mathrm{T}(G, c)$. Значит, θ нарушает условие чётности вершины v формулы $\mathrm{T}(H, f)$. Следовательно, узел s диаграммы D вычисляет $\mathrm{SearchVertex}(H, f)$.

Проверим, что D является структурированной ветвящейся программой. Определим $\nu(s) = (H, f)$; H — связный подграф G, а $\mathrm{T}(H, f)$ невыполнима. Если s помечена переменной x_e, то, по предложению 4, e — ребро H. Свойства стоков, очевидно, выполнены. Локальные свойства проверяются непосредственно, поскольку нам известно, что́ вычисляет каждый узел D. □

4. Построение 1-BP для $\mathrm{T}(G, c')$ по 1-BP для $\mathrm{SearchVertex}(G, c)$

Теорема 2. *Пусть* $\mathrm{T}(G, c)$ *— невыполнимая цейтинская формула. Если существует регулярное резолюционное опровержение* $\mathrm{T}(G, c)$ *размера* S*, то для любой функции пометок* c' *такой, что* $\mathrm{T}(G, c')$ *выполнима, существует* 1-BP*, вычисляющая* $\mathrm{T}(G, c')$*, размер которой* $S^{\mathcal{O}(\log n)}$*, где* n *— число вершин в графе* G*.*

По лемме 1 размер регулярного резолюционного опровержения $\mathrm{T}(G, c)$ равен размеру минимальной 1-BP, вычисляющей $\mathrm{Search}_{\mathrm{T}(G,c)}$. Размер последней, в свою очередь, не меньше размера минимальной 1-BP, вычисляющей $\mathrm{SearchVertex}(G, c)$. Поэтому для доказательства теоремы 2 достаточно доказать:

Теорема 3. *Пусть* $G(V, E)$ *— связный граф, формула* $\mathrm{T}(G, c')$ *выполнима, а* $\mathrm{T}(G, c)$ *невыполнима. Пусть существует* 1-BP*, вычисляющая* $\mathrm{SearchVertex}(G, c)$*, размер которой равен* S*. Тогда существует* 1-BP*, вычисляющая* $\mathrm{T}(G, c')$*, размер которой не более* $S^{\mathcal{O}(\log |V|)}$*.*

Опишем идею доказательства теоремы 3. По теореме 1, мы можем считать, что 1-BP, вычисляющая $\mathrm{SearchVertex}(G, c)$, является структурированной. Поскольку определения структурированных ветвящихся программ для вычисления SearchVertex и для вычисления выполнимых цейтинских формул похожи, мы сможем преобразовать первую программу во вторую по индукции по обратному топологическому порядку. Однако, есть одно существенное различие между структурированными программами — локальное свойство для случая, когда ребро e является мостом G_s. В этом случае мы не можем просто перестроить программу, мы вынуждены дублировать некоторые вершины несколько раз, отсюда и возникает увеличение в размере.

Кроме того, по лемме 5, конкретный выбор функции пометок не важен, важна только выполнимость формулы.

Доказательство теоремы 3. Пусть D — минимальная по размеру

1-BP, вычисляющая SearchVertex(G, c), и пусть S — её размер. По лемме 3, каждая вершина G может быть единственной вершиной с нарушенным условием чётности в $\mathrm{T}(G, c)$, поэтому D содержит хотя бы $|V|$ стоков и, следовательно, $S \geqslant |V|$. По теореме 1, D — структурированная ветвящаяся программа, вычисляющая SearchVertex(G, c).

По пункту (1) предложения 1 и по лемме 5, достаточно построить структурированную ветвящуюся программу, вычисляющую выполнимую формулу $\mathrm{T}(G, c')$, размер которой будет не более $S^{\mathcal{O}(\log |V|)}$.

Если $V_H \subseteq V$, то для графа $H(V_H, E_H)$ обозначим за $\widehat{H}(V, E_H)$ граф, который получается из H добавлением изолированных вершин $V \setminus V_H$. Для функции пометок $c_H \colon V_H \to \{0, 1\}$ обозначим за $\widehat{c_H} \colon V \to \{0, 1\}$ функцию пометок, продолжающую c_H до V нулями. Для вершины $w \in V$ обозначим $\mathbf{1}_w \colon V \to \{0, 1\}$ функцию пометок, которая равна 1 только на вершине w.

Занумеруем узлы D в обратном топологическом порядке: $u_1, u_2, \ldots, u_S$ — то есть каждое ребро D направлено от узла с большим номером к узлу с меньшим. Пусть $\nu(u_i) = (G_i(V_i, E_i), c_i)$ для всех $i \in [S]$. Для k от 0 до S мы шаг за шагом строим структурированную ветвящуюся программу $D^{(k)}$, вычисляющую выполнимые цейтинские формулы, причём для каждого $i \in [k]$, для каждой функции пометок $c_i' \colon V_i \to \{0, 1\}$ отличающейся от c_i ровно в одной вершине из V_i, существует узел s программы $D^{(k)}$, для которого $\mu(s) = (\widehat{G_i}, \widehat{c_i'})$.

База индукции: для $k = 0$ программа $D^{(0)}$ состоит из 0-стока и 1-стока, и $\mu(\text{1-сток}) = (G_\varnothing(V, \varnothing), \mathbf{0})$.

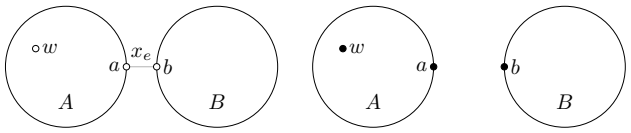
Переход. Пусть программа $D^{(k-1)}$ уже построена. Покажем, как добавить к ней несколько узлов и определить μ на них, чтобы полученная таким образом программа $D^{(k)}$ была структурированной ветвящейся программой, вычисляющей выполнимые цейтинские формулы, и удовлетворяла условию для u_k.

Если u_k — сток, помеченный вершиной v, то граф G_k состоит из единственной вершины v и $c_k(v) = 1$. В этом случае мы не добавляем в $D^{(k-1)}$ новых узлов (то есть $D^{(k)} = D^{(k-1)}$), поскольку 1-сток удовлетворяет условию для u_k.

Теперь предположим, что u_k — внутренняя вершина, помеченная переменной x_e. Пусть из u_k выходит помеченное i ребро (u_k, u_{k_i}) для $i \in \{0,1\}$. Для каждой вершины w графа G_k добавим узел s_w в $D^{(k)}$, пометим его переменной x_e и расширим μ так, что $\mu(s_w) = (\widehat{G_k}, \widehat{c_k} + \mathbf{1}_w)$.

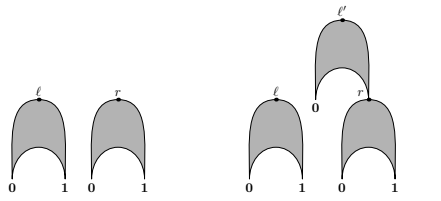
Рассмотрим два случая.

e не мост G_k. По локальному свойству D, графы G_{k_0} и G_{k_1} оба равны $G_k - e$. Пусть w — вершина G_k, тогда она принадлежит графам G_{k_0} и G_{k_1}. По предположению индукции для k_0 и k_1, существуют узлы s_w^0 и s_w^1 в $D^{(k-1)}$, что $\mu(s_w^0) = (\widehat{G_{k_0}}, \widehat{c_{k_0}} + \mathbf{1}_w)$ и $\mu(s_w^1) = (\widehat{G_{k_1}}, \widehat{c_{k_1}} + \mathbf{1}_w)$. Добавим в $D^{(k)}$ два ребра: из нового узла s_w в (s_w, s_w^i), помеченное i для $i \in \{0,1\}$. Заметим, что по локальному свойству для D, c_{k_0} совпадает с c_k, а c_{k_1} отличается от c_k только в концах e, поэтому то же верно для $\widehat{c_{k_0}}, \widehat{c_{k_1}}$ и $\widehat{c_k}$ с инвертированным значением в вершине w. Поэтому локальное свойство выполнено для узла s_w в $D^{(k)}$.



граф u_k граф s_w^δ граф $s_b^{1-\delta}$

Рис. 1: Графы узлов (для $\delta = 1$); закрашены чёрным вершины, у которых пометка отличается от значения c_k.



до копирования после копирования

Рис. 2: Копирование.

e — мост G_k. В этом случае $G_k - e$ представляется в виде дизъюнктного объединения двух связных подграфов G_k: $A(V_A, E_A)$ и $B(V_B, E_B)$. Пусть w — вершина G_k; не умаляя общности, пусть A содержит w. Пусть $a \in A$, $b \in B$ — концы ребра e.

Положим $\delta = \sum_{v \in V_A} c_k(v)$. Проведём ребро из s_w в 0-сток, помеченное δ, и ребро из s_w в некоторый узел ℓ', помеченное $1-\delta$. Для узла ℓ' должно выполняться $\mu(\ell') = (\widehat{G_k - e}, \widehat{c_k} + \mathbf{1}_w + (1-\delta)(\mathbf{1}_a + \mathbf{1}_b))$. Локальное свойство для s_w будет выполнено, однако, мы должны объяснить, как получить

такой узел ℓ'. (Отметим, что γ в определении локального свойства равна $\sum_{v \in V_A} (\widehat{c_k} + \mathbf{1}_w)(v) = 1 - \delta$.)

По локальному свойству для D, $G_{k_{1-\delta}} = A$ и $G_{k_\delta} = B$. По предположению индукции, $D^{(k-1)}$ содержит узел $s_w^{1-\delta}$ такой, что $\mu(s_w^{1-\delta}) = (\widehat{A}, \widehat{c_{k_{1-\delta}}} + \mathbf{1}_w)$, и узел s_b^δ такой, что $\mu(s_b^\delta) = (\widehat{B}, \widehat{c_{k_\delta}} + \mathbf{1}_b)$ (см. рис. 1).

Предложение 7. *Пусть V_H и V_F — два непересекающихся подмножества V, $h\colon V_H \to \{0,1\}$ и $f\colon V_F \to \{0,1\}$ — две функции пометок, и $H(V_H, E_H)$ и $F(V_F, E_F)$ — два графа, причём формулы $\mathrm{T}(H,h)$ и $\mathrm{T}(F,f)$ выполнимы. Пусть D_H и D_F вместе с отображениями μ_H и μ_F — структурированные ветвящиеся программы, вычисляющие выполнимые цейтинские формулы $\mathrm{T}(\widehat{H}, \widehat{h})$ и $\mathrm{T}(\widehat{F}, \widehat{f})$, и множества узлов D_H и D_F не пересекаются.*

Определим ветвящуюся программу $D_{H \cup F}$: перенаправим рёбра D_H, идущие в 1-сток, в исток программы D_F; кроме того, удалим 1-сток D_H и объединим два 0-стока в один 0-сток. Определим отображение $\mu_{H \cup F}$ (для всех узлов $D_{H \cup F}$, кроме 0-стока): если s — узел D_F, то положим $\mu_{H \cup F}(s) = \mu_F(s)$. Если s — узел D_H и $\mu_H(s) = (H_s, c_s)$, то положим $\mu_{H \cup F}(s) = (H_s \cup F, c_s + \widehat{t})$, где $H_s \cup F$ — граф на множестве вершин V, множество рёбер которого — объединение рёбер H_s и E_F. Тогда $D_{H \cup F}$ вместе с отображением $\mu_{H \cup F}$ структурированная ветвящаяся программа, вычисляющая выполнимую цейтинскую формулу $\mathrm{T}(\widehat{H \cup F}, \widehat{h} + \widehat{f})$.

Доказательство. Доказываем непосредственной проверкой локального свойства. □

Предложение 7 объясняет, как создать узел, который отображение μ переведёт в $(\widehat{A \cup B}, \widehat{c_{k_{1-\delta}}} + \mathbf{1}_w + \widehat{c_{k_\delta}} + \mathbf{1}_b)$. Заметим, что, по локальному свойству для D, $\widehat{c_{k_{1-\delta}}} + \widehat{c_{k_\delta}} = \widehat{c_k} + (1-\delta)\mathbf{1}_a + \delta \mathbf{1}_b$, следовательно $\widehat{c_{k_{1-\delta}}} + \mathbf{1}_w + \widehat{c_{k_\delta}} + \mathbf{1}_b = \widehat{c_k} + \mathbf{1}_w + (1-\delta)(\mathbf{1}_a + \mathbf{1}_b)$ и, следовательно, этот узел можно использовать в качестве ℓ'. Однако, конструкция в предложении 7 не может быть использована напрямую, поскольку в ней мы изменяем значения μ для некоторых узлов, и это может нарушить условия на $D^{(k)}$. Чтобы этого избежать, мы будем копировать узлы.

Если $|V_A| \leqslant |V_B|$, положим $\ell = s_w^{1-\delta}$ и $r = s_b^{\delta}$, иначе положим $\ell = s_b^{\delta}$ и $r = s_w^{1-\delta}$. Скопируем подпрограмму ℓ (то есть всех последователей ℓ, кроме стоков) вместе со значениями μ на этих узлах, и добавим копию к $D^{(k)}$. Обозначим копию ℓ за ℓ'. Применим предложение 7, где в качестве H возьмём подпрограмму ℓ', а в качестве F — подпрограмму r (см. рис. 2). Как было описано выше, ℓ' удовлетворяет необходимым условиям.

Доказав индукционное утверждение, мы получаем, что $D^{(S)}$ — структурированная ветвящаяся программа, вычисляющая выполнимые цейтинские формулы, и содержащая узел s, вычисляющий $\mathrm{T}(G, c')$, где c' отличается от c в одной вершине. Осталось оценить количество узлов в $D^{(S)}$. Заметим, что если есть два узла s и t с одинаковыми значениями μ, то t можно удалить, перенаправив все входящие в него рёбра в s. Поэтому мы будем считать, что все значения μ различны, и оценим количество возможных различных значений.

Утверждение 2. Пусть s — узел $D^{(S)}$ и $\mu(s) = (Q, q)$. Рассмотрим все компоненты связности Q размера хотя бы два в порядке возрастания размера: $C_1(V_{C_1}, E_{C_1}), C_2(V_{C_2}, E_{C_2}), \ldots, C_m(V_{C_m}, E_{C_m})$, и $2 \leqslant |V_{C_1}| \leqslant |V_{C_2}| \leqslant \ldots \leqslant |V_{C_m}|$. Тогда:

1. $|V_{C_i}| \geqslant |V_{C_1}| + \cdots + |V_{C_{i-1}}|$ для каждого $i \in [m]$.

2. Для каждого $i \in [m]$ существует узел p программы D такой, что $\nu(p) = (C_i, h)$, где h отличается от q ровно в одной вершине из V_{C_i}.

Доказательство. Докажем утверждение по индукции по k для всех узлов программы $D^{(k)}$.

$D^{(0)}$ состоит только из стоков, поэтому утверждение верно для $l = 0$.

Пусть узел s был добавлен как новый узел $s = s_w$, и выполняется $\mu(s_w) = (\widehat{G_k}, \widehat{c_k} + \mathbf{1}_w)$. Заметим, что $\nu(u_k) = (G_k, c_k)$, поэтому $\widehat{G_k}$ может иметь максимум одну компоненту размера хотя бы два — G_k. Кроме того, $\widehat{c_k} + \mathbf{1}_w$ отличается от c_k на V только в вершине w. Поэтому u_k подходит в качестве p.

Теперь рассмотрим случай, когда узел s получен в результате копирования и применения предложения 7.

Пусть предложение 7 было применено для подпрограмм узлов ℓ и r диаграммы $D^{(k-1)}$. Пусть $\mu(\ell) = (\widehat{H}, \widehat{h})$, $\mu(r) = (\widehat{F}, \widehat{f})$, где графы $H(V_H, E_H)$ и $F(V_F, E_F)$ связны, $h\colon V_H \to \{0,1\}$, $f\colon V_F \to \{0,1\}$ и $V_H \cap V_F = \varnothing$. Рассмотрим узел t подпрограммы ℓ. Поскольку ℓ лежит в $D^{(k-1)}$, t также лежит в $D^{(k-1)}$. Пусть t' — копия t, созданная на этом шаге; проверим условие для $s = t'$.

По локальному свойству для $D^{(k-1)}$, $\mu(t) = (\widehat{H_t}, h_t)$, где H_t — подграф H. Новый узел t' имеет $\mu(t') = (H_{t'}, h_{t'})$, где $H_{t'}$ получено из $\widehat{H_t}$ добавлением всех рёбер новой компоненты связности $F(V_F, E_F)$ (все вершины V_F изолированы в $\widehat{H_t}$). Поскольку $|V_F| \geqslant |V_H|$ и H_t — подграф H, то $|V_F|$ не меньше суммарного размера всех компонент связности H_t с хотя бы двумя вершинами. Функция пометок $h_{t'}$ отличается от h_t только на V_F и совпадает с f на V_F. Следовательно, утверждение для t' следует из индукционного предположения для узлов r и t. □

Рассмотрим узел s программы $D^{(S)}$, пусть $\mu(s) = (H, f)$. Если v изолирована в H, то $f(v) = 0$. Предположим, что H содержит m компонент связности с хотя бы двумя вершинами. По первому пункту утверждения 2, $m \leqslant \log|V|$. Рассмотрим компоненту связности $C(V_C, E_C)$ графа H с хотя бы двумя вершинами. По второму пункту утверждения 2, существует не более $S|V|$ различных значений пары $(C, f|_{V_C})$. Следовательно, количество различных значений $\mu(s)$ не более $\sum_{m=0}^{\lfloor \log|V| \rfloor} (S|V|)^m \leqslant \log|V| \cdot (S|V|)^{\log|V|} = S^{\mathcal{O}(\log|V|)}$ (воспользовались тем, что $S \geqslant |V|$).

□

5. Построение вывода $\neg\mathrm{T}(G, c)$ в системе Фреге константной глубины по 1-BP для $\mathrm{SearchVertex}(G, c)$

5.1. Системы Фреге

В этом разделе рассматриваются пропозициональные формулы в базисе, состоящем из бинарной конъюнкции, бинарной дизъюнкции, унарного отрицания и логических констант 0 (ложь) и 1 (истина).

Конкретная система Фреге определяется конечным множеством правил вывода. *Правило вывода* П имеет вид $\frac{\phi_1,\ldots,\phi_k}{\phi}$, где $\phi_1, \ldots, \phi_k$ и ϕ — пропозициональные формулы, а $k \geqslant 0$. Правило вывода должно быть корректным, то есть ϕ должно семантически следовать из $\{\phi_1, \ldots, \phi_k\}$. Если σ — подстановка, отображающая переменные в формулы, то мы говорим, что формула $\phi[\sigma]$ получается из $\phi_1[\sigma], \ldots, \phi_k[\sigma]$ применением правила П.

Пусть Г — множество пропозициональных формул. Будем говорить, что ϕ *выводится из* Г, если существует последовательность формул $\psi_1, \psi_2, \ldots, \psi_s$ такая, что $\psi_s = \phi$, и каждая ψ_i либо принадлежит Г, либо может быть получена из формул с меньшими номерами применением какого-то из правил вывода. Формула τ *выводима*, если она может быть выведена из пустого множества формул.

Система правил вывода называется системой Фреге, если она импликационно полна: если формула ϕ семантически следует из множества формул Г, то ϕ должна выводиться из Г.

Глубина вывода — это максимальная из глубин формул, которые в нём встречаются. *Длина вывода* — количество формул в нём. *Размер вывода* — число встречающихся в нём различных подформул. Если в вывод размера S всюду вместо переменной x подставить формулу ϕ, получится новый вывод размера не более $S + |\phi|$.

5.2. Вспомогательные утверждения

В этом подразделе доказаны вспомогательные утверждения, которые понадобятся для построения вывода в системе Фреге.

Мы не фиксируем конкретную систему Фреге, однако будет удобнее расширить систему несколькими корректными правилами вывода. (Ниже в предложении 8 мы покажем, почему это не умаляет общность.)

Будем называть правило вывода $\frac{\phi}{\psi}$ системы Фреге *двусторонним*, если в этой системе также есть правило $\frac{\psi}{\phi}$. Вместе оба этих правила мы будем обозначать как $\frac{\phi}{=\psi}$. Заметим, что из корректности правил следует, что ϕ эквивалентна ψ, то есть для любой подстановки переменных ϕ и ψ имеют одно и то же значение.

Будем использовать систему Фреге со следующими дополнительными правилами:

1. Коммутативность дизъюнкции: $\frac{p \vee q}{=q \vee p}$.
2. Коммутативность конъюнкции: $\frac{p \wedge q}{=q \wedge p}$.
3. Ассоциативность дизъюнкции: $\frac{a \vee (b \vee c)}{=(a \vee b) \vee c}$.
4. Ассоциативность конъюнкции: $\frac{a \wedge (b \wedge c)}{=(a \wedge b) \wedge c}$.
5. Дистрибутивность: $\frac{(a \vee b) \wedge c}{=(a \wedge c) \vee (b \wedge c)}$.
6. Законы де Моргана: $\frac{\neg(a \vee b)}{=\neg a \wedge \neg b}$, $\frac{\neg(a \wedge b)}{=\neg a \vee \neg b}$.
7. Добавление или удаление двойного отрицания: $\frac{a}{=\neg\neg a}$.
8. Ослабление: $\frac{a}{a \vee b}$, $\frac{a \wedge b}{a}$.
9. Ветвление: $\frac{x, y}{(z \wedge x) \vee (\neg z \wedge y)}$.

Лемма 6. *Для любой системы Фреге $\mathcal{F}$ существует функция $C_{\mathcal{F}} : \mathbb{N} \to \mathbb{N}$ такая, что для любого k, любого множества формул Γ и любой формулы ϕ таких, что общий размер $\Gamma \cup \{\phi\}$ не превышает k, если ϕ семантически следует из Γ, то существует вывод ϕ из Γ в системе $\mathcal{F}$ размера не более $C_{\mathcal{F}}(k)$.*

Доказательство. Существует лишь конечное число пар (Γ, ψ) таких, что общий размер не превышает k. Для каждой такой пары, если из Γ семантически следует ϕ, то по импликационной полноте $\mathcal{F}$ существует вывод ϕ из Γ. Определим $C_{\mathcal{F}}(k)$ как максимум размеров таких выводов по всем парам. □

Предложение 8. *Пусть $\mathcal{F}'$ — система Фреге, полученная из другой системы Фреге $\mathcal{F}$ добавлением нескольких корректных правил вывода. Тогда существует константа C такая, что для каждый $\mathcal{F}'$-вывод D' может быть преобразован в $\mathcal{F}$-вывод D так, что размер D не более чем в C раз больше размера D', а глубина D не более чем на C больше глубины D'.*

Доказательство. Пусть общий размер всех формул в любом правиле вывода системы $\mathcal{F}'$ ограничен константой k. По лемме 6, для заключения любого правила существует его вывод из предпосылок в системе $\mathcal{F}$, размер которого не более $C_{\mathcal{F}}(k)$, а значит и глубина не более $C_{\mathcal{F}}(k)$.

Каждое применение правила системы $\mathcal{F}'$ можно заменить на соответствующий вывод. Размер всего вывода увеличивается не более чем в $C_{\mathcal{F}}(k)$ раз, а глубина увеличивается не более, чем на $C_{\mathcal{F}}(k)$. □

При применении двустороннего правила формула заменяется на эквивалентную ей. Следующее предложение показывает, что двусторонние правила можно применять не только ко всей формуле, но и к её подформулам, если вся формула имеет константную глубину.

Предложение 9. *Пусть $\mathcal{F}$ — система Фреге, Π — двустороннее правило вывода $\mathcal{F}$, а ϕ' — формула, получаемая из формулы ϕ одним применением правила Π. Пусть Ψ — формула константной глубины без четырёх вложенных отрицаний подряд, содержащая ϕ как подформулу (мы фиксируем одно вхождение ϕ в Ψ), а Ψ' — формула, получаемая из Ψ заменой вхождения ϕ на ϕ'. Тогда существует вывод Ψ' из Ψ в системе $\mathcal{F}$, размер которого не превышает $\mathcal{O}(|\Psi|^2 + |\Psi'|^2)$, а глубина не более $\mathcal{O}(1)$.*

Доказательство.

Утверждение 3. Пусть $\sigma(x, \bar{y})$ — формула размера s, где x — переменная, а $\bar{y}$ — вектор из нескольких переменных. Пусть $\frac{\tau_1}{\tau_2}$ — двустороннее правило вывода системы Фреге $\mathcal{F}$, а суммарный размер формул τ_1 и τ_2 равен c. Пусть ϕ_2 получается из ϕ_1 применением этого правила. Тогда существует вывод формулы $\sigma(\phi_2, \bar{y})$ в системе $\mathcal{F}$ из формулы $\sigma(\phi_1, \bar{y})$ такой, что его длина не более $C_{\mathcal{F}}(s + c)$, а любая формула в нём имеет размер не более $C_{\mathcal{F}}(s+c) \cdot \max\{|\phi_1|, |\phi_2|\}$ и глубину не более максимальной из глубин формул ϕ_1 и ϕ_2, увеличенной на $C_{\mathcal{F}}(s + c)$.

Доказательство. Поскольку τ_1 эквивалентна τ_2, $\sigma(\tau_1, \bar{y})$ эквивалентна $\sigma(\tau_2, \bar{y})$. По лемме 6, существует вывод в системе $\mathcal{F}$ формулы $\sigma(\tau_2, \bar{y})$ из $\sigma(\tau_1, \bar{y})$ размера не более $C_{\mathcal{F}}(s+c)$. Изменим этот вывод: применим подстановку, которая превращает τ_1 в ϕ_1, а τ_2 в ϕ_2 (такая существует, так как ϕ_2 получена из ϕ_1 применением правила $\frac{\tau_1}{\tau_2}$). Таким образом, размер каждой формулы вывода увеличился не более, чем в $\max\{|\phi_1|, |\phi_2|\}$ раз, а глубина увеличилась не более, чем на максимальную из глубин формул ϕ_1 и ϕ_2 плюс $C_{\mathcal{F}}(s + c)$. □

Пусть d — глубина вхождения ϕ в формулу Ψ. Рассмотрим последовательность логических операций $o_1, o_2, \ldots, o_h$, встречающихся на пути от корня Ψ до вхождения ϕ, причём если встречаются несколько конъюнкций подряд, то мы оставляем только одну из них; аналогично поступаем с дизъюнкциями (но не с отрицаниями). По условиям предложения, в этой последовательности может идти не более трёх отрицаний подряд, поэтому $h \leqslant 3d$, и значит, h — константа.

Пусть k — число конъюнкций и дизъюнкций в этой последовательности. Определим понятие *оболочки* последовательности $o_1, o_2, \ldots, o_h$ (иногда мы будем говорить об оболочке вхождения ϕ в Ψ, имея в виду оболочку соответствующей последовательности) следующий образом. Неформально говоря, оболочка — это формула, полученная из Ψ заменой ϕ на новую переменную z и перестановками такими, чтобы все вершины на пути то корня Ψ до z были левыми операндами операций на пути, а также заменой вторых операндов этих операций на новые переменные x_i для $i \in [k]$ (у отрицаний вторых операндов нет).

Например, для $\Psi = (y_1 \wedge y_2) \vee \neg(y_3 \wedge \phi)$ мы имеем $h = 3$, $o_1 = \vee, o_2 = \neg, o_3 = \wedge$, и оболочкой последовательности o_1, o_2, o_3 является формула $\neg(z \wedge x_1) \vee x_2$.

Формально определим оболочку по индукции. Если $h = 0$, оболочкой будет формула z. Пусть $h > 0$. Если o_1 — это отрицание, и $\sigma(z, x_1, x_2, \ldots, x_k)$ — оболочка для $o_2, o_3, \ldots, o_h$, то $\neg\sigma(z, x_1, x_2, \ldots, x_k)$ будет оболочкой для $o_1, \ldots, o_h$. Если o_1 — конъюнкция или дизъюнкция, и $\sigma(z, x_1, x_2, \ldots, x_{k-1})$ оболочка для $o_2, o_3, \ldots, o_h$, то $o_1(\sigma(z, x_1, \ldots, x_{k-1}), x_k)$ будет оболочкой для $o_1, \ldots, o_h$. (То есть в случае конъюнкции оболочкой будет $\sigma(z, x_1, \ldots, x_{k-1}) \wedge x_k$, а в случае дизъюнкции — $\sigma(z, x_1, \ldots, x_{k-1}) \vee x_k$.)

Пусть $\sigma(z, x_1, \ldots, x_k)$ — оболочка вхождения ϕ в Ψ. Обозначим за Ψ_z формулу, которая получается из Ψ при замене вхождения ϕ новой переменной z. Мы покажем, что $\sigma(z, \phi_1, \ldots, \phi_k)$ выводится из Ψ_z, где ϕ_i — формула, которую мы заменяем на x_i в нашем неформальном определении оболочки. Кроме того, мы покажем, что Ψ_z выводится из $\sigma(z, \phi_1, \ldots, \phi_k)$. Эти выводы будут иметь размер $\mathcal{O}(|\Psi_z|^2)$ и глубину $\mathcal{O}(1)$.

Итоговый вывод будет устроен следующим образом: мы подставляем ϕ вместо z в вывод $\sigma(z, \phi_1, \ldots, \phi_k)$ из Ψ_z и получаем вывод $\sigma(\phi, \phi_1, \ldots, \phi_k)$ из Ψ размера $\mathcal{O}(|\Psi_z|^2 + |\phi|) = \mathcal{O}(|\Psi|^2)$ и глубины $\mathcal{O}(1)$, затем по утверждению 3 получаем вывод $\sigma(\phi', x_1, \ldots, x_k)$ из $\sigma(\phi, x_1, \ldots, x_k)$ размера $\mathcal{O}(|\phi| + |\phi'|)$, и подставляем ϕ_i вместо x_i для всех $i \in [k]$, получая вывод $\sigma(\phi', \phi_1, \ldots, \phi_k)$ из $\sigma(\phi, \phi_1, \ldots, \phi_k)$ размера $\mathcal{O}(|\Psi| + |\Psi'|)$ и глубины $\mathcal{O}(1)$. Затем мы рассматриваем вывод Ψ_z из $\sigma(z, \phi_1, \ldots, \phi_k)$ и подставляем ϕ' вместо z, получая вывод Φ' из $\sigma(\phi', \phi_1, \ldots, \phi_k)$ размера $\mathcal{O}(|\Psi'|^2)$ и глубины $\mathcal{O}(1)$. Наконец, мы получаем требуемый вывод размера $\mathcal{O}(|\Psi|^2 + |\Psi'|^2)$ и глубины $\mathcal{O}(1)$.

Осталось доказать следующее утверждение:

Утверждение 4. Существует вывод $\sigma(z, \phi_k, \phi_{k-1}, \ldots, \phi_1)$ из Ψ_z и вывод Ψ_z из $\sigma(z, \phi_k, \phi_{k-1}, \ldots, \phi_1)$ для некоторых формул $\phi_1, \ldots, \phi_{k-1}, \phi_k$ такие, что размер каждого из этих выводов есть $\mathcal{O}(|\Psi_z|^2)$, а глубина $\mathcal{O}(1)$.

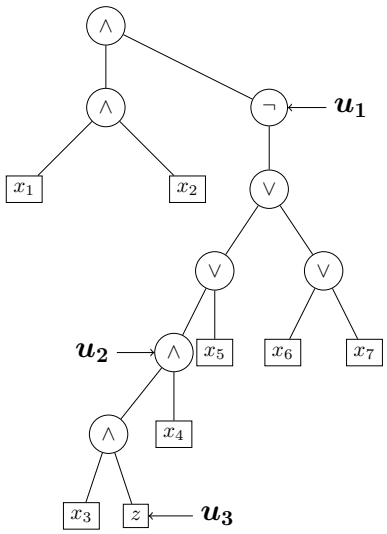
Доказательство. Пусть формула F содержит только бинарные дизъюнкции (или конъюнкции) и представляется как $\bigvee_{i\in I} x_i$ (или $\bigwedge_{i\in I} x_i$). Будем называть x_j *самым левым главным операндом*, если F имеет вид $x_j \vee F'$ (или $x_j \wedge F'$); при этом F' будем называть *остатком*. Например, x является самым левым главным операндом для формулы $x \vee ((y \vee r) \vee t)$, а формула $((y \vee r) \vee t)$ — её остаток. У формулы $(x \vee y) \vee r$ нет самого левого главного операнда.

Рассмотрим формулу Ψ_z (как дерево) и путь от её корня до вхождения z. На этом пути выделим k вершин $u_1, u_2, \ldots, u_k$, соответствующих операндам конъюнкций или дизъюнкций таким, что для каждой выделенной вершины операция в ней отличается от операции в непосредственном родителе (u_k считаем равным вершине, соответствующей z). Для каждого i от 1 до k мы будем менять порядок применения дизъюнкции или конъюнкции таким образом, чтобы u_i стало самым левым главным операндом, а за ϕ_i обозначим соответствующий остаток. Мы выполняем эти действия последовательно: сначала для u_1, затем для u_2, и так далее, и, наконец, для u_k. См. рис. 3.

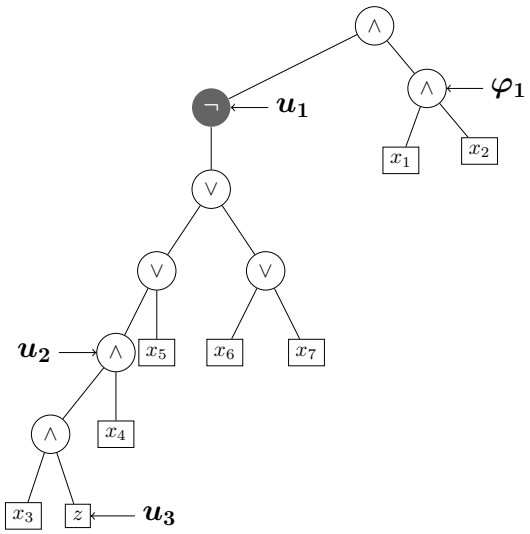
Пусть i' — минимальное число из $\{0, 1, \ldots, h\}$ такое, что $o_1, \ldots, o_{i'}$ — префикс последовательности $o_1, o_2, \ldots, o_h$, который соответствует операциям от корня до родителя u_i, кроме последней операции (возможно, $i' = 0$). Чтобы сделать u_i самым левым главным операндом дизъюнкции или конъюнкции, мы используем правила ассоциативности и коммутативности этой операции. Эти правила двунаправленные, поэтому их можно применять для подформул по утверждению 3, где σ будет оболочкой $o_1, \ldots, o_{i'}$. Размер вывода увеличивается в константу раз при каждом применении утверждения 3.

Пусть u_i операнд дизъюнкции (или конъюнкции) из t_i операндов. Чтобы сделать u_i самым левым главным операндом, мы применим $\mathcal{O}(t_i)$ правил, размер каждой формулы в этом выводе $\mathcal{O}(|\Psi_z|)$, поэтому даже с учётом применения утверждения 3, размер вывода для обработки u_i будет $\mathcal{O}(|\Psi_z| t_i)$, а его глубина $\mathcal{O}(1)$. Таким образом, общий размер вывода будет $\mathcal{O}(|\Psi_z|(t_1 + t_2 + \cdots + t_k)) = \mathcal{O}(|\Psi_z|^2)$, а его глубина $\mathcal{O}(1)$.

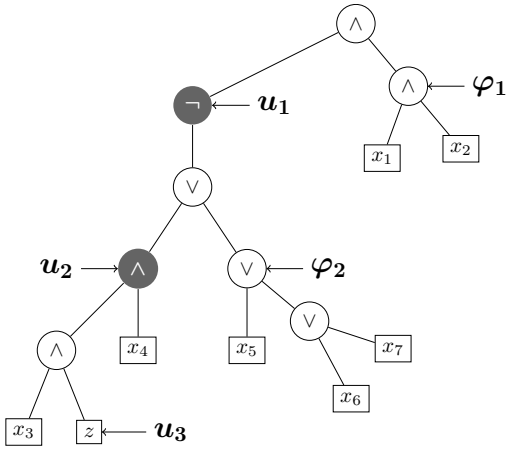
Обратный вывод строится аналогично с таким же размером и глу-



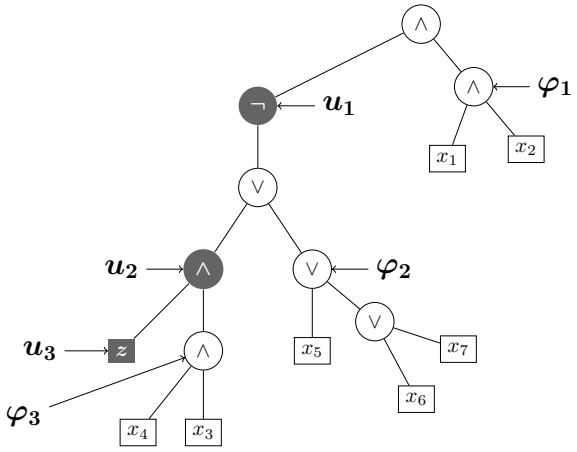
Исходная формула



После первого шага



После второго шага



После третьего шага

Рис. 3: Вывод $\sigma(z, \phi_k, \phi_{k-1}, \ldots, \phi_1)$.

биной, так как были использованы только двусторонние правила. □

□

Мы считаем, что запись $\bigvee_{i=1}^{n} x_i$ левоассоциативна и обозначает $x_1 \vee (x_2 \vee (x_3 \vee \ldots x_n) \ldots)$; аналогичное соглашение и для конъюнкции $\bigwedge$.

Предложение 10. *Пусть* $\phi = \bigvee_{i=1}^{n} x_i$, σ *— перестановка* $\{1, \ldots, n\}$. *Тогда существует вывод* $\phi' = \bigvee_{i=1}^{n} x_{\sigma(i)}$ *из* ϕ *размера не более* $\mathcal{O}(n^4)$ *и глубины* $\mathcal{O}(1)$.

Доказательство. Получим необходимую перестановку, используя $\mathcal{O}(n^2)$ элементарных транспозиций. Это может быть сделано с помощью

правил коммутативности и ассоциативности дизъюнкции по предложению 9 (таким образом, каждое применение имеет увеличивает размер на $\mathcal{O}(n^2)$). □

5.3. Построение вывода

Теорема 4. *Пусть D — минимальная по размеру* 1-BP*, вычисляющая* SearchVertex(G, c)*, где $G(V, E)$ — связный граф. Тогда существует вывод формулы $\neg\mathrm{T}(G, c)$ в системе Фреге, размер которого не больше $\mathcal{O}\big(\big(|D||\mathrm{T}(G, c)|^4\big)$, а глубина не больше $\mathcal{O}(1)$, где $\mathrm{T}(G, c)$ представлена в КНФ.*

Доказательство. По предложению 8 можно считать, что в данной нам системе Фреге есть правила 1-9 из раздела 5.2.

Для пропозициональной переменной x введём обозначения $x^0 = \neg x$, $x^1 = x$. Мы предполагаем, что на все рёбра G линейно упорядочены, и все вершины G тоже линейно упорядочены. Для любого подграфа G порядок на его рёбрах индуцируется порядком на рёбрах во всём G, и аналогично с вершинами. Эти порядки всегда будут использоваться, если явно не оговорено иное.

Пусть $H(V_H, E_H)$ — подграф G, а $f : V_H \to \{0, 1\}$ — функция пометок. Для $v \in V_H$ обозначим множество инцидентных v рёбер в H как $I_H(v)$.

Обозначим

$$\mathrm{err}(H, v, f) = \bigvee_{\substack{m \in \{0,1\}^{I_H(v)} \\ \sum_{e \in I_H(v)} m_e \neq f(v) \bmod 2}} \bigwedge_{e \in I_H(v)} x_e^{m_e},$$

то есть $\mathrm{err}(H, v, f)$ — это формула, истинная тогда и только тогда, когда подстановка значений переменных не удовлетворяет условие чётности Цейтинской формулы $\mathrm{T}(H, f)$ в вершине v.

Для $U \subseteq V_H$ обозначим $\mathrm{Err}(H, U, f) = \bigvee_{u \in U} \mathrm{err}(H, u, f)$, то есть $\mathrm{Err}(H, U, f)$ — формула, утверждающая, что условие чётности нарушено хотя бы в одной вершине из U. Если v — изолированная вершина в H, то $\mathrm{err}(H, v, f)$ равно логической константе 1, если $f(v) = 1$, и логической

константе 0 в противном случае.

План нашего доказательства состоит в том, чтобы сначала вывести формулу $\mathrm{Err}(G, V, c)$, а затем вывести из неё формулу $\neg\mathrm{T}(G, c)$.

Не умаляя общности, предположим, что D — минимальная по размеру 1-BP, вычисляющая $\mathrm{SearchVertex}(G, c)$. По теореме 1 каждый узел s программы D вычисляет $\mathrm{SearchVertex}(G_s, c_s)$, где $G_s(V_s, E_s)$ — связный граф, и для каждой частичной подстановки α, соответствующей пути от истока к s, $\mathrm{T}(G, c)|_\alpha$ имеет единственную невыполнимую компоненту G_s с функцией пометок c_s.

Мы собираемся вывести $\mathrm{Err}(G_s, V_s, c_s)$ для всех узлов s программы D. Если r — исток D, то $\mathrm{Err}(G_r, V_r, c_r)$ совпадает с $\mathrm{Err}(G, V, c)$.

Если узел s — сток, то G_s содержит только одну вершину, поэтому $\mathrm{Err}(G_s, V_s, c_s)$ есть логическая константа 1, для которой есть вывод константного размера и константной глубины.

Пусть теперь s не сток, а s_0 и s_1 — непосредственные последователи s, причём ребро (s, s_i) помечено i для $i \in \{0, 1\}$.

Обозначим $M = |\mathrm{T}(G, c)|$.

Утверждение 5. Существует вывод $\mathrm{Err}(G_s, V_s, c_s)$ из $\mathrm{Err}(G_{s_0}, V_{s_0}, c_{s_0})$ и $\mathrm{Err}(G_{s_1}, V_{s_1}, c_{s_1})$ размера не более $\mathcal{O}(M^4)$ и глубины $\mathcal{O}(1)$.

Доказательство. Пусть узел s помечен переменной x_e, где $e \in E_s$, и пусть e соединяет вершины u и v. Для упрощения обозначений положим $G_i(V_i, E_i) = G_{s_i}$, $c_i = c_{s_i}$ и $\phi_i = \mathrm{Err}(G_i, V_i, c_i)$ для $i \in \{0, 1\}$. Если e — мост, то не умаляя общности будем считать, что $u \in V_0$, $v \in V_1$; если e не мост, то мы будем считать, что v стоит раньше чем u в фиксированном порядке на вершинах.

Выведем две формулы ψ_0 и ψ_1 из ϕ_0 и ϕ_1:

$$\psi_i = \mathrm{Err}(G_i, V_i \cap \{v, u\}, c_i) \vee \mathrm{Err}(G_i, V_i \setminus \{v, u\}, c_i), \quad \text{для } i \in \{0, 1\}.$$

ψ_i можно получить из ϕ_i перестановкой элементов дизъюнкции. По предложению 10 существует соответствующий вывод глубины $\mathcal{O}(1)$ и размера $\mathcal{O}(M^4)$.

Выведем формулу $\psi = (x_e \wedge \psi_1) \vee (\neg x_e \wedge \psi_0)$, используя правило ветвления (правило 9) из ψ_0 и ψ_1. Размер такого вывода $\mathcal{O}(M)$, а глубина $O(1)$.

Раскрывая определение ψ_i, мы получаем:

$$\psi = \Big(x_e \wedge \big(\mathrm{Err}(G_1, V_1 \cap \{v,u\}, c_1) \vee \mathrm{Err}(G_1, V_1 \setminus \{v,u\}, c_1)\big)\Big) \vee \\ \Big(\neg x_e \wedge \big(\mathrm{Err}(G_0, V_0 \cap \{v,u\}, c_0) \vee \mathrm{Err}(G_0, V_0 \setminus \{v,u\}, c_0)\big)\Big).$$

Теперь применим правило дистрибутивности к подформулам по предложению 9, а затем правило коммутативности дизъюнкции, тем самым получая:

$$\Big(x_e \wedge \mathrm{Err}(G_1, V_1 \cap \{v,u\}, c_1)\Big) \vee \Big(\neg x_e \wedge \mathrm{Err}(G_0, V_0 \cap \{v,u\}, c_0)\Big) \vee \\ \Big(x_e \wedge \mathrm{Err}(G_1, V_1 \setminus \{v,u\}, c_1)\Big) \vee \Big(\neg x_e \wedge \mathrm{Err}(G_0, V_0 \setminus \{v,u\}, c_0)\Big).$$

Размер такого вывода не больше $\mathcal{O}(M^2)$, а глубина равна $\mathcal{O}(1)$.

Заметим, что пропозициональная формула $A \vee B \vee (x_e \wedge C) \vee (\neg x_e \wedge D)$ семантически влечёт формулу $A \vee B \vee (x_e \wedge C) \vee (\neg x_e \wedge D)$, поэтому существует вывод константного размера и глубины второй формулы из первой. В этот вывод мы подставляем вместо A, B, C и D соответствующие подформулы выведенной выше формулы и получаем:

$$\Big(x_e \wedge \mathrm{Err}(G_1, V_1 \cap \{v,u\}, c_1)\Big) \vee \Big(\neg x_e \wedge \mathrm{Err}(G_0, V_0 \cap \{v,u\}, c_0)\Big) \vee \\ \mathrm{Err}(G_1, V_1 \setminus \{v,u\}, c_1) \vee \mathrm{Err}(G_0, V_0 \setminus \{v,u\}, c_0).$$

Размер такого вывода $\mathcal{O}(M)$, а глубина $\mathcal{O}(1)$.

Заметим, что c_0 и c_1 отличаются от c_s только в вершинах u и v. Поэтому во второй части формулы мы можем заменить c_0 и c_1 на c_s. Таким образом, сейчас выведена следующая формула:

$$\Big(x_e \wedge \mathrm{Err}(G_1, V_1 \cap \{v,u\}, c_1)\Big) \vee \Big(\neg x_e \wedge \mathrm{Err}(G_0, V_0 \cap \{v,u\}, c_0)\Big) \vee \\ \mathrm{Err}(G_1, V_1 \setminus \{v,u\}, c_s) \vee \mathrm{Err}(G_0, V_0 \setminus \{v,u\}, c_s). \quad (1)$$

Рассмотрим два случая.

1. e не мост в графе G_s. Тогда $G_s = G_0 = G_1$, $V_0 = V_1 = V_s$, $\{v, u\} \subseteq V_s$ и $\mathrm{Err}(G_1, V_1) \setminus \{v, u\}, c_s) = \mathrm{Err}(G_0, V_0 \setminus \{v, u\}, c_s)$, здесь равенство мы понимаем как равенство формул, поскольку вершины перечисляются в одном и том же фиксированном порядке.

Выведем формулу, которая получается из (1) удалением четвёртого операнда дизъюнкции. Заметим, что он совпадает с третьим операндом. Поскольку формула $A \vee B \vee C \vee C$ семантически влечёт $A \vee B \vee C$, существует вывод константного размера и глубины второй формулы из первой, в который вместо A, B и C мы подставляем соответствующие подформулы формулы (1). Кроме того, раскроем определение Err в первых двух операндах дизъюнкции, получая тем самым:

$$\Big(x_e \wedge \big(\mathrm{err}(G_1, v, c_1) \vee \mathrm{err}(G_1, u, c_1)\big)\Big) \vee \Big(\neg x_e \wedge \big(\mathrm{err}(G_0, v, c_0) \vee \mathrm{err}(G_0, u, c_0)\big)\Big) \vee \mathrm{Err}(G_s, V_s \setminus \{v, u\}, c_s).$$

Размер этого вывода равен $\mathcal{O}(M)$, а его глубина $\mathcal{O}(1)$.

Теперь применим правило дистрибутивности к подформулам (пользуясь предложением 9) и внесём x_e^0 и x_e^1 в операнды дизъюнкции:

$$\left(\bigvee_{\substack{m \in \{0,1\}^{I_1(v)} \\ \sum_{f \in I_1(v)} m_f \neq c_1(v)}} \left(x_e^1 \wedge \bigwedge_{f \in I_1(v)} x_f^{m_f}\right)\right) \vee \left(\bigvee_{\substack{m \in \{0,1\}^{I_1(u)} \\ \sum_{f \in I_1(u)} m_f \neq c_1(u)}} \left(x_e^1 \wedge \bigwedge_{f \in I_1(u)} x_f^{m_f}\right)\right) \vee$$
$$\left(\bigvee_{\substack{m \in \{0,1\}^{I_0(v)} \\ \sum_{f \in I_0(v)} m_f \neq c_0(v)}} \left(x_e^0 \wedge \bigwedge_{f \in I_0(v)} x_f^{m_f}\right)\right) \vee \left(\bigvee_{\substack{m \in \{0,1\}^{I_0(u)} \\ \sum_{f \in I_0(u)} m_f \neq c_0(u)}} \left(x_e^0 \wedge \bigwedge_{f \in I_0(u)} x_f^{m_f}\right)\right) \vee$$
$$\mathrm{Err}(G_s, V_s \setminus \{v, u\}, c_s).$$

Здесь $I_i(w)$ обозначает $I_{G_i}(w)$. Размер такого вывода есть $\mathcal{O}(M^3)$, а глубина $\mathcal{O}(1)$.

Отметим, что $c_0(w) = c_s(w)$, $c_1(w) = 1 - c_s(w)$ для $w \in \{u, v\}$. Перейдём в графу G_s и пронесём x_e и $\neg x_e$ внутрь конъюнкций на место в соответствии с порядком на рёбрах E_s, пользуясь правилами ассоциа-

тивности и коммутативности конъюнкции (а также используя предложение 9).

$$\left(\bigvee_{\substack{m\in\{0,1\}^{Is(v)}\\ \sum_{f\in I_s(v)} m_f\neq c_s(v)\\ m_e=1}}\bigwedge_{f\in I_s(v)} x_f^{m_f}\right)\vee\left(\bigvee_{\substack{m\in\{0,1\}^{Is(u)}\\ \sum_{f\in I_s(u)} m_f\neq c_s(u)\\ m_e=1}}\bigwedge_{f\in I_s(u)} x_f^{m_f}\right)\vee$$
$$\left(\bigvee_{\substack{m\in\{0,1\}^{Is(v)}\\ \sum_{f\in I_s(v)} m_f\neq c_s(v)\\ m_e=0}}\bigwedge_{f\in I_s(v)} x_f^{m_f}\right)\vee\left(\bigvee_{\substack{m\in\{0,1\}^{Is(u)}\\ \sum_{f\in I_s(u)} m_f\neq c_s(u)\\ m_e=0}}\bigwedge_{f\in I_s(u)} x_f^{m_f}\right)\vee$$
$$\mathrm{Err}(G_s, V_s\setminus\{v,u\}, c_s).$$

Размер такого вывода будет не более $\mathcal{O}(M^3)$, а глубина $\mathcal{O}(1)$.

Таким образом, с точностью до порядка операндов дизъюнкции, мы получили $\mathrm{err}(G_s, v, c_s)\vee\mathrm{err}(G_s, u, c_s)\vee\mathrm{Err}(G_s, V_s\setminus\{v,u\}, c_s)$. Переставляя конъюнкты ещё раз, мы наконец получаем требуемую формулу $\mathrm{Err}(G_s, V_s, c_s)=\phi_s$. Согласно предложению 10, такая перестановка может быть выполнена выводом размера $\mathcal{O}(M^4)$ и глубины $\mathcal{O}(1)$.

2. e — мост графа G_s. Тогда $V_1\cap V_0=\varnothing$; формула (1) имеет следующий вид:

$$\Big(x_e\wedge\mathrm{err}(G_1, v, c_1)\Big)\vee\Big(\neg x_e\wedge\mathrm{err}(G_0, u, c_0)\Big)\vee$$
$$\mathrm{Err}(G_1, V_1\setminus\{v,u\}, c_s)\vee\mathrm{Err}(G_0, V_0\setminus\{v,u\}, c_s).$$

Заметим, что для каждого $i\in\{0,1\}$ в формуле $\mathrm{Err}(G_i, V_i\setminus\{v,u\}, c_s)$ можно заменить G_i на G_s, поскольку для любой вершины из $V_i\setminus\{v,u\}$ множество инцидентных ей рёбер одинаково в G_i и в G_s. Таким образом, последняя формула может быть записана в следующем виде:

$$\Big(x_e \wedge \mathrm{err}(G_1, v, c_1)\Big) \vee \Big(\neg x_e \wedge \mathrm{err}(G_0, u, c_0)\Big) \vee$$
$$\mathrm{Err}(G_s, V_1 \setminus \{v, u\}, c_s) \vee \mathrm{Err}(G_s, V_0 \setminus \{v, u\}, c_s).$$

Раскроем определение err в первых двух операндах дизъюнкции. Далее применим правило дистрибутивности к подформулам (пользуясь предложением 9) и пронесём x_e^0 и x_e^1 внутрь дизъюнкции:

$$\left(\bigvee_{\substack{m \in \{0,1\}^{I_1(v)} \\ \sum_{f \in I_1(v)} m_f \neq c_1(v)}} \left(x_e^1 \wedge \bigwedge_{f \in I_1(v)} x_f^{m_f}\right)\right) \vee \left(\bigvee_{\substack{m \in \{0,1\}^{I_0(u)} \\ \sum_{f \in I_0(u)} m_f \neq c_0(u)}} \left(x_e^0 \wedge \bigwedge_{f \in I_0(u)} x_f^{m_f}\right)\right) \vee$$
$$\mathrm{Err}(G_s, V_1 \setminus \{v, u\}, c_s) \vee \mathrm{Err}(G_s, V_0 \setminus \{v, u\}, c_s).$$

Размер такого вывода равен $\mathcal{O}(M^3)$, а его глубина $\mathcal{O}(1)$.

Воспользуемся теперь равенствами $c_1(v) = 1 - c_s(v)$ и $c_0(u) = c_s(u)$. Перейдём в графу G_s и пронесём x_e и $\neg x_e$ внутрь конъюнкций на место в соответствии с порядком на рёбрах E_s, пользуясь правилами ассоциативности и коммутативности конъюнкции (а также используя предложение 9):

$$\left(\bigvee_{\substack{m \in \{0,1\}^{I_s(v)} \\ \sum_{f \in I_s(v)} m_f \neq c_s(v) \\ m_e = 1}} \bigwedge_{f \in I_s(v)} x_f^{m_f}\right) \vee \left(\bigvee_{\substack{m \in \{0,1\}^{I_s(u)} \\ \sum_{f \in I_s(u)} m_f \neq c_s(u) \\ m_e = 0}} \bigwedge_{f \in I_s(u)} x_f^{m_f}\right) \vee$$
$$\mathrm{Err}(G_s, V_1 \setminus \{v, u\}, c_s) \vee \mathrm{Err}(G_s, V_0 \setminus \{v, u\}, c_s).$$

Размер такого вывода равен $\mathcal{O}(M^3)$, а глубина $\mathcal{O}(1)$.

Заметим, что первый операнд — подформула формулы $\mathrm{err}(G_s, v, c_s)$, а второй — формулы $\mathrm{err}(G_s, u, c_s)$. Таким образом, вся формула, с точ-

ностью до порядка операндов дизъюнкции, является подформулой

$$\mathrm{err}(G_s, v, c_s) \vee \mathrm{err}(G_s, u, c_s) \vee \mathrm{Err}(G_s, V_1 \setminus \{v, u\}, c_s) \vee \mathrm{Err}(G_s, V_0 \setminus \{v, u\}, c_s),$$

то есть с точностью до порядка операндов дизъюнкции, является подформулой формулы $\mathrm{Err}(G_s, V_s, c_s)$.

Применим тогда правило ослабления (правило 8) и добавим недостающие конъюнкты. Затем переставим операнды дизъюнкции по предложению 10 и выведем $\mathrm{Err}(G_s, V_s, c_s)$. Размер такого вывода будет не больше $\mathcal{O}(M^4)$, а глубина $\mathcal{O}(1)$.

Итоговый размер вывода в обоих случаях будет не более $\mathcal{O}(M^4)$, а глубина не более $\mathcal{O}(1)$. □

Таким образом, если r — исток D, то существует вывод формулы $\mathrm{Err}(G_r, V_r, c_r)$, совпадающей с $\mathrm{Err}(G, V, c)$, размер которого не более $\mathcal{O}(M^4|D|)$, а глубина $\mathcal{O}(1)$.

Теперь выведем $\neg\mathrm{T}(G, c)$ из $\mathrm{Err}(G, V, c)$. Сначала применим правило двойного отрицания и получим $\neg\neg\mathrm{Err}(G, V, c)$. Затем пронесём второе отрицание по законам де Моргана (правило 6) и пользуясь предложением 9. Размер такого вывода будет не более $\mathcal{O}(M^3)$, а глубина $\mathcal{O}(1)$.

Итоговый размер всего вывода формулы $\neg\mathrm{T}(G, c)$ не больше $\mathcal{O}(|D|M^4)$, а его глубина равна $\mathcal{O}(1)$. □

Заключение

Таким образом, мы решили поставленные задачи: построили верхнюю оценку на 1-BP($\mathrm{T}(G, c')$) по 1-BP(SearchVertex(G, c)), а значит и по $S_{\mathrm{Reg}}(\mathrm{T}(G, c))$; построили верхнюю оценку на минимальный размер вывода $\neg\mathrm{T}(G, c)$ в системе Фреге константной глубины по 1-BP(SearchVertex(G, c)). В частности, мы получили два способа доказательства нижних оценок на сложность SearchVertex(G, c) для 1-BP: достаточно получить нижнюю оценку на размер 1-BP(SearchVertex(G, c)), либо на размер вывода $\neg\mathrm{T}(G, c)$ в системе Фреге константной глубины.

Структурная теорема и теорема о построении 1-BP для $\mathrm{T}(G, c')$ по 1-BP для SearchVertex(G, c) включены в статью [13]. Во второй части этой статьи для всех графов G доказывается нижняя оценка 1-BP$(\mathrm{T}(G, c')) \geqslant 2^{\Omega(\mathrm{tw}(G))}$, где $\mathrm{tw}(G)$ — древесная ширина графа G. Вместе с теоремой о перестроении, это даёт нижнюю оценку на минимальный размер регулярного резолюционного доказательства цейтинской формулы для любого графа G: $S_{\mathrm{Reg}}(\mathrm{T}(G, c)) \geqslant 2^{\Omega(\mathrm{tw}(G)/\log|V|)}$. Для графов константной степени эта оценка оказывается почти точной: для них известна верхняя оценка $S_{\mathrm{Reg}}(\mathrm{T}(G, c)) \leqslant 2^{\mathcal{O}(\mathrm{tw}(G))}\mathrm{poly}(|V|)$ [1].

Кроме того, в статье [13] строится семейство выполнимых цейтинских формул $\mathrm{T}(G_n, c'_n)$ на графах $G_n(V_n, E_n)$ константной степени, для которых 1-BP$(\mathrm{T}(G_n, c'_n)) \geqslant 2^{\Omega(\mathrm{tw}(G_n)\log|V_n|)}$. При этом по [1] нам известно, что 1-BP(SearchVertex$(G_n, c_n)) \leqslant S_{\mathrm{Reg}}(\mathrm{T}(G_n, c_n)) \leqslant 2^{\mathcal{O}(\mathrm{tw}(G_n))}\mathrm{poly}(|V_n|)$. Из этого следует, что при перестроении 1-BP для SearchVertex(G, c) в 1-BP для $\mathrm{T}(G, c')$ нельзя избавиться от показателя степени $\log|V|$.

Список литературы

[1] Michael Alekhnovich and Alexander A. Razborov. Satisfiability, branch-width and tseitin tautologies. *Comput. Complex.*, 20(4):649–678, 2011.

[2] Eli Ben-Sasson. Size space tradeoffs for resolution. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 457–464, New York, NY, USA, 2002. ACM.

[3] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001.

[4] Sam Buss, Dmitry Itsykson, Alexander Knop, and Dmitry Sokolov. Reordering rule makes OBDD proof systems stronger. In *Computational Complexity Conference*, volume 102 of *LIPIcs*, pages 16:1–16:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[5] Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. In *STOC*, pages 547–556. ACM, 1999.

[6] Stephen Cook and Robert Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, STOC '74, pages 135–148, New York, NY, USA, 1974. ACM.

[7] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

[8] Ludmila Glinskih and Dmitry Itsykson. Satisfiable tseitin formulas are hard for nondeterministic read-once branching programs. In *MFCS*, volume 83 of *LIPIcs*, pages 26:1–26:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[9] Ludmila Glinskih and Dmitry Itsykson. On tseitin formulas, read-once branching programs and treewidth. In *CSR*, volume 11532 of *Lecture Notes in Computer Science*, pages 143–155. Springer, 2019.

[10] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theor. Comput. Sci.*, 259(1-2):613–622, 2001.

[11] Johan Håstad. On small-depth frege proofs for tseitin for grids. In *FOCS*, pages 97–108. IEEE Computer Society, 2017.

[12] Dmitry Itsykson, Alexander Knop, Andrei E. Romashchenko, and Dmitry Sokolov. On obdd-based algorithms and proof systems that dynamically change order of variables. In *STACS*, volume 66 of *LIPIcs*, pages 43:1–43:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[13] Dmitry Itsykson, Artur Riazanov, Danil Sagunov, and Petr Smirnov. Almost tight lower bounds on regular resolution refutations of tseitin formulas for all constant-degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:178, 2019.

[14] Arist Kojevnikov and Dmitry Itsykson. Lower bounds of static lovász-schrijver calculus proofs for tseitin tautologies. In *ICALP (1)*, volume 4051 of *Lecture Notes in Computer Science*, pages 323–334. Springer, 2006.

[15] Jan Krajícek. Lower bounds to the size of constant-depth propositional proofs. *J. Symb. Log.*, 59(1):73–86, 1994.

[16] L.A. Levin. Universal problems of full search. *Probl. Peredachi Inf.*, 9(3):115–116, 1973.

[17] László Lovász, Moni Naor, Ilan Newman, and Avi Wigderson. Search problems in the decision tree model. *SIAM J. Discret. Math.*, 8(1):119–132, 1995.

[18] Toniann Pitassi, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. Poly-logarithmic frege depth lower bounds via an expander switching lemma. In *STOC*, pages 644–657. ACM, 2016.

[19] G.S. Tseitin. On the complexity of derivation in the propositional calculus. In *Studies in Constructive Mathematics and Mathematical Logic Part II*. A. O. Slisenko, editor, a968.

[20] Alasdair Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, January 1987.