Сложность пропозициональных доказательств (черновик краткого конспекта*лекций)

Д.М. Ицыксон †

19 октября 2019 г.

Содержание

*Этот текст содержит массу опечаток, ошибок и неточностей. Просьба сообщать о найденных недостатках по электронной почте, в теме письма напишите слово КОНСПЕКТ. Все найденные недостатки будут исправляться в новых версиях конспекта.

†ПОМИ РАН. E-mail: dmitrits@pdmi.ras.ru.

1 Системы доказательств

Пусть Σ — конечное множество, которое мы будем называть алфавитом. Языком мы будем называть произвольное множество конечных слов над алфавитом Σ. Другими словами, язык — это подмножество Σ^*.

Мы сейчас определим понятие системы доказательств для языка L, доказывать мы будем утверждения вида $x \in L$.

Системой доказательств для языка L называется полиномиальный по времени алгоритм Π, который принимает два входа x (строка, для которой доказывается принадлежность к языку L) и w (доказательство) и выдает ответ из множества $\{0, 1\}$. Алгоритм должен обладать следующими свойствами:

- (Корректность) если $\Pi(x, w) = 1$ для некоторых строк $x, w \in \Sigma^*$, то $x \in L$;
- (Полнота) для любой строки $x \in L$ существует строка $w \in \Sigma^*$, что $\Pi(x, w) = 1$.

Система доказательств Π для языка L называется полиномиально ограниченной, если существует такой полином q, что для всех $x \in L$ существует строка w длины не больше, чем $q(|x|)$, что $\Pi(x, w) = 1$.

Класс сложности P состоит из языков, для которых существует полиномиальный по времени алгоритм, проверяющий принадлежность этому языку. Класс сложности NP состоит из языков, для которых существует полиномиально ограниченная система доказательств. Легко понять, что $\mathrm{P} \subseteq \mathrm{NP}$, поскольку для языка из P легко построить систему доказательств, которая не использует доказательство. Неизвестно, является ли это включение строгим.

Между языками определяются сведения. Будем говорить, что язык L сводится за полиномиальное время (или по Карпу) к языку L', если существует такая полиномиально вычислимая функция f, что для всех строк x выполняется $x \in L$ тогда и только тогда, когда $f(x) \in L'$. Обозначаются сведения так: $L \leq_p L'$.

Легко проверить, что сведения обладают свойством рефлексивности, транзитивности и замкнутости относительно принадлежности классу P.

В классе NP лежит специфический язык SAT, который состоит из всех выполнимых пропозициональных формул в конъюнктивной нормальной форме (КНФ). Действительно, доказательством пренадлежности этому языку будет набор значений переменных, которые выполняют эту формулу.

Теорема 1.1 (Кук, Левин)**.** Любой язык из класса NP сводится к SAT за полиномиальное время.

Доказательство этой теоремы мы не приводим. Вы можете его прочитать, например, тут: [AB09].

Класс coNP состоит из языков, дополнение которых лежит в классе NP.

Заметим, что если $\mathrm{P} = \mathrm{NP}$, то и $\mathrm{NP} = \mathrm{coNP} = \mathrm{P}$. Значит, из неравенства классов NP и coNP следует неравенство классов P и NP.

Например, в классе coNP лежит множество невыполнимых формул в КНФ, которое мы обозначаем UNSAT. Действительно, дополнение этого языка состоит из выполнимых формул в КНФ и строк, которые не являются формулами в КНФ.

Из теоремы Кука-Левина легко получается:

Следствие 1.1. Любой язык из класса coNP сводится к UNSAT за полиномиальное время.

Язык UNSAT является основным объектом изучения в теории сложности пропозициональных доказательств. Пропозициональной системой доказательств называется система доказательств для языка UNSAT.

Предложение 1.1. Классы NP и coNP замкнуты относительно полиномиальных по времени сведениям.

Доказательство. Пусть L сводится к L' с помощью функции f и $L' \in$ NP. Пусть Π' — это полиномиально ограниченная система доказательств для L'. Определим $\Pi(x, w) = \Pi'(f(x), w)$. Нетрудно проверить, что Π — это полиномиально ограниченная система доказательств для L. В случае coNP доказательство аналогичное, надо рассматривать полиномиально ограниченную систему доказательств для дополнения языка L'. □

Предложение 1.2 (Кук, Рекхау [CR74])**.** NP = coNP тогда и только тогда, когда для UNSAT есть полиномиально ограниченная система доказательств.

Доказательство. Если NP = coNP, то UNSAT $\in$ NP, следовательно по определению класса NP существует полиномиально ограниченная система доказательств для языка NUSAT.

Пусть UNSAT $\in$ NP, поскольку все языки из coNP сводятся к UNSAT, то по предложению 1.1 coNP $\subseteq$ NP. Нетрудно понять, что из UNSAT $\in$ NP следует, что SAT $\in$ coNP. Тогда по теореме Кука-Левина и предложению 1.1 выполняется NP $\subseteq$ coNP. □

Основная программа исследований в теории сложности пропозициональных доказательств состоит в доказательстве суперполиномиальных нижних оценок на длину доказательства во все более и более сильных системах доказательств. Тем самым мы медленно будем приближаться к доказательству неравенства классов NP и coNP.

2 Деревья решений

Мы начнем изучать системы доказательств с самых слабых. Первая система доказательств — это таблица истинности, мы просто выписываем значение формулы при всех значениях переменных. Понятно, что проверка такого доказательства занимает полиномиальное от длины доказательства время. Длина такого доказательства всегда 2^n, где n — это число переменных. Стоит отметить, что нам важна только длина доказательств, само оно не особенно содержательное, поскольку состоит из одних нулей.

Рассмотрим более интересную систему доказательств, деревья решений.

Деревом решений для невыполнимой КНФ формулы ϕ называется бинарное дерево, все вершины которого кроме листьев помечены переменными формулы. Одно из двух исходящих ребер, из вершины, помеченной переменной x помечено подстановкой $x := 0$, другое $x := 1$. Листья дерева помечены дизъюнктами формулы ϕ, причем дизъюнкт в любом листе опровергается подстановкой, которая читается на пути от корня до этого листа.

Такое дерево является доказательством невыполнимости формулы ϕ. Действительно, мы просто перебираем возможные значения переменных и убеждаемся, что во всех случаях мы получаем противоречие с дизъюнктами формулы ϕ. Проверить такое доказательство легко: достаточно проверить что дерево корректное, т.е. для всех листьев надо проверить, что дизъюнкт опровергается подстановкой, которая соответствует пути до этого листа.

Размером дерева удобно считать число листьев, поскольку число вершин не более, чем в два раза превосходит число листьев. Нетрудно видеть, что всегда существует дерево размером 2^n, где n число переменных, достаточно просто перебрать все значения всех переменных. Однако в реальности деревья могут быть значительно короче.

Кроме размера важной характеристикой является глубина дерева (максимальное расстояние от корня до листа).

Предложение 2.3. Размер дерева решений не превосходит 2^d, где d — это его глубина.

Наша ближайшая цель состоит в том, чтобы привести пример формулы (семейства формул), которая имеет полиномиальный размер, но требует экспоненциального дерева решений.

У нас будет несколько базовых примеров, которые мы будем доказывать в разных системах доказательств. Первый такой пример — это принцип Дирихле (pigeonhole principle). Формула $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{m}}$ утверждает, что m кроликов сидят в n клетках так, что каждый кролик сидит в одной клетке, и в каждой клетке сидит не более одного кролика. Формула содержит переменные $p_{i,j}$, где $i \in [m], j \in [n]$, которая означает, что i-й кролик сидит в j-й клетке. Формула содержит два типа дизъюнктов. Дизъюнкты кроликов:

- $p_{i,1} \vee p_{i,2} \vee \cdots \vee p_{i,n}$, для всех $i \in [m]$;

и дизъюнкты для клеток:

- $\neg p_{i,j} \vee \neg p_{k,j}$, для всех $i \neq k \in [m]$ и всех $j \in [n]$;

Нетрудно видеть, что при $m > n$ формула $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{m}}$ является невыполнимой.

Докажем для начала нижнюю оценку на глубину дерева решений для формулы $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{m}}$. Это можно сделать с помощью следующей игры двух игроков. Дана невыполнимая формула ϕ и два игрока: Алиса и Боб, по очереди Алиса выбирает переменную формулы, Боб выбирает значение этой переменной. Игра заканчивается, как только сформированная подстановка противоречит дизъюнкту формулы ϕ. Боб зарабатывает очко за каждый свой ход. Цель Боба заработать как можно больше очков.

Лемма 2.1. Если в указанной игре для невыполнимой формулы ϕ у Боба есть стратегия, которая позволяет ему заработать как минимум t очков, то глубина дерева решений формулы ϕ как минимум t.

Доказательство. Пусть есть дерево решений для формулы ϕ глубины меньше t, тогда Алиса будет делать запросы согласно этому дереву и противоречие будет найдено за менее, чем t шагов, следовательно Боб заработает менее t очков, противоречие. □

С помощью этой игры очень легко доказать, что глубина дерева решений для формулы $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{m}}$ не меньше, чем n. Действительно, Боб может на все вопросы отвечать 0. Заметим, что ни один из клеточных дизъюнктов не может опровергнуться, значит опровергнулся кроличий дизъюнкт, следовательно Боб давал как минимум n ответов и заработал как минимум n очков. Эту оценку можно улучшить до $\Omega(n^2)$, выбрав более хитрую стратегию, это остается в качестве упражнения.

Однако из нижней оценки на глубину не следует нижняя оценка на размер дерева решений. Чтобы доказать оценку на размер, мы модифицируем игру. Теперь первого игрока зовут

Прувер, а второго Делэйер. Прувер также выбирает значение переменной, а Делэер либо отвечает $*$, либо выбирает значение переменной из $\{0,1\}$. В случае ответа $*$ Прувер выбирает значение сам. Игра заканчивается, когда текущая подстановка опровергает дизъюнкт формулы ϕ. На этот раз Делэер получает очки не за все ответы, а только за ответ $*$. В этом виде игра была предложена в работе Пудлака и Иммальяццо 2000 года [PI00].

Лемма 2.2. Если в игре Прувера и Делэера для невыполнимой формулы ϕ у Делэера есть стратегия, которая позволяет ему заработать как минимум t очков, то размер любого дерева решений формулы ϕ как минимум 2^t.

Доказательство. Рассмотрим какое-нибудь дерево решений T и рассмотрим стратегию Прувера на этом дереве: он спрашивает значение переменной (сначала в корне дерева), если Делэер говорит значение, то Прувер переходит по соответствующему ребру дерева, если Делэер возвращает $*$, то Прувер выбирает значение случайным образом и перемещается в дереве согласно этому выбору. Игра с вероятностью 1 заканчивается в листе дерева. Поскольку на пути до каждого листа Делэер заработал как минимум t очков, то вероятность оказаться в этом листе не более 2^{-t}. Следовательно число листов как минимум 2^t. □

Теперь можно доказать оценку на размер дерева.

Предложение 2.4. Размер любого дерева решений формулы $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{m}}$ при $m > n$ не меньше, чем 2^n.

Доказательство. Достаточно описать стратегию для Делэера, которая гарантирует ему заработать как минимум n очков. Стратегия очень простая: если Прувер спрашивает про переменную $p_{i,j}$ и при этом в клетке j уже кто-то сидит, то Делэер отвечает 0, в противном случае он отвечает $*$. Поскольку Делэер следит за тем, чтобы в одну клетку не посадили двух кроликов, то игра может закончиться только тем, что нарушился кроличий дизъюнкт $p_{k,1} \vee p_{k,2} \vee \cdots \vee p_{k,n}$. Покажем, что для каждой клетки Делэер заработал хотя бы одно очко. Действительно, кто подставил $p_{k,j} = 0$. Если это был Прувер, то это был ответ $*$ и очко для Делэера. Если это был Делэер, то он это мог сделать только потому, что в клетке j уже кто-то сидит. Но туда кого-то посадить мог только Прувер, поскольку Делэер не дает ответы 1 сам. Следовательно, за этот ответ Делэер заработал очко. Итого, Делэер заработал как минимум n очков. □

Хорошо, нижнюю оценку мы доказали. А насколько она точная? Тривиальная верхняя оценка для формулы $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{n+1}}$ — это $2^{O(n^2)}$, поскольку число переменных $n(n+1)$. Следующая лемма показывает, что верхнюю оценку можно улучшить.

Лемма 2.3. Существует дерево решений для формулы $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{m}}$ при $m > n$ размера $2^{O(n \log n)}$.

Доказательство. Поскольку формула $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{m}}$ при $m > n$ содержит все дизъюнкты $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{n+1}}$, то достаточно доказывать только для $m = n + 1$.

Пусть минимальное дерево для формулы $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{n+1}}$ имеет размер T_n. Оценим T_n через T_{n-1}, для этого рассмотрим дерево, которое сначала перебирает значение переменной $p_{n+1,1}$, в ветви со значением 0 спрашивается значение $p_{n+1,2}$ и т.д. В этом дереве есть ветвь, которая соответствует подстановке $p_{n+1,1} = p_{n+1,2} = \cdots = p_{n+1,n} = 0$, эта ветвь опровергает дизъюнкт, который соответствует $(n+1)$-му кролику. У нас есть еще n ветвей, в которых мы $n+1$ кролика куда-то посадили. В каждой из этих ветвей надо расщепиться по всем другим кроликам сидящим в

той же клетке. Каждый раз, значению 1 будет соответствовать немедленное противоречию с клеточным дизъюнктом. Останутся n вершин, в каждой из которой $(n+1)$ кролик где-то сидит и больше никто в ней не сидит. Т.е. n раз надо опровергнуть формулу $\mathrm{PHP}^{\mathrm{n}}_{\mathrm{n}-1}$ (возможно только клетки в ней иначе переименованы). Таким образом мы можем оценить $T_n \le nT_{n-1}+n^2+1$. При этом $T_1 \le 2$. Докажем индукцией по n, что $T_n \le 2^{2n\log n}$. База очевидна, индукционный переход: $T_n \le nT_{n-1}+n^2+1 \le 2^{2(n-1)\log(n-1)+\log n}+n^2+1 \le 2^{(2n-1)\log n}+2^{3\log n} \le 2^{2n\log n}$ при $n \ge 2$. □

Оказывается, что нижнюю оценку тоже можно улучшить до $2^{\Omega(n\log n)}$. Впервые это было доказано Данчевым и Риисом в 2001 году [DR01]. Мы приведем доказательство, использующее асимметриченые игры Прувера и Делэера, предложенные Бейесдорфом, Галези и Лауриа в 2010 году [BGL10]. Асимметричная игра Прувера и Делэера отличается от обычной тем, что вместе с ответом $*$ делэер сообщает два положительных вещественных числа (a_0, a_1), что $\frac{1}{a_0}+\frac{1}{a_1}=1$. При этом, если Прувер выбирает 0, то Делэер получает $\log a_0$ очков, а если Прувер выбирает 1, то $\log a_1$ очков. В случае обычной игры $a_0 = a_1 = 2$. Оказывается, что лемма 2.2 выполняется и для асимметричной игры.

Лемма 2.4. Если в асимметричной игре Прувера и Делэера для невыполнимой формулы ϕ у Делэера есть стратегия, которая позволяет ему заработать как минимум t очков, то размер любого дерева решений формулы ϕ как минимум 2^t.

Доказательство. Рассмотрим какое-нибудь дерево решений T и рассмотрим стратегию Прувера на этом дереве: он спрашивает значение переменной (сначала в корне дерева), если Делэер говорит значение, то Прувер переходит по соответствующему ребру дерева, если Делэер возвращает $*$ и два числа (a_0, a_1), то Прувер выбирает значение случайным образом: 0 с вероятностью $\frac{1}{a_0}$ и 1 с вероятностью $\frac{1}{a_1}$ и переходит по соответствующему ребру дерева. Игра с вероятностью 1 заканчивается в листе дерева. Рассмотрим какой-нибудь лист дерева, в котором закончилась игра, пусть на пути до этого листа Делэер k раз выдавал $*$ с числами $(a_0^{(1)}, a_1^{(1)}), (a_0^{(2)}, a_1^{(2)}), \ldots, (a_0^{(k)}, a_1^{(k)})$ и лист соответствует выборам $i_1, i_2, \ldots, i_k \in \{0,1\}$. Поскольку на пути до каждого листа Делэер заработал как минимум t очков, то вероятность оказаться в этом листе равна $\frac{1}{\prod_{j=1}^{k} a_{i_j}^{(j)}} = 2^{-\sum_{j=1}^{k}\log a_{i_j}^{(j)}} \le 2^{-t}$. Следовательно число листов как минимум 2^t. □

Теперь мы готовы доказать асимптотически точную нижнюю оценку.

Теорема 2.1. Размер любого дерева решений формулы $\mathrm{PHP}^{\mathrm{m}}_{\mathrm{n}}$ при $m > n$ не меньше, чем $2^{\Omega(n\log n)}$.

Доказательство. Мы опишем стратегию для Делэера в асимметричной игре. Числа (a_0, a_1) будут одинаковыми для всех ходов, а чему они равны мы выберем позже.

Также как и раньше Делэер не будет позволять сажать кролика в клетку, в которой уже кто-то сидит, но теперь будут выполняться и другие ограничения. Пусть α — это текущая подстановка, которая соответствует ответам Прувера, ответы Делэера мы сюда не включаем. Обозначим через $J_i(\alpha)$ множество клеток, которые явно запрещены кролику i и при этом в этих клетках никто не сидит. Более формально $J_i(\alpha) = \{j \in [n] \mid \alpha(p_{i,j}) = 0 \text{ и } \alpha(p_{i',j}) \ne 1 \text{ при } i' \in [m]\}$.

Стратегия Делэера будет такой. Пусть Прувер спрашивает значение переменной $p_{i,j}$:

- Он отвечает 0, если в j-й клетке уже кто-то сидит или i-й кролик уже где-то сидит.
- Он отвечает 1, если $|J_i(\alpha)| \geq n/2$, j-я клетка свободна и i-й кролик еще нигде не сидит.
- Он отвечает $*$ в противном случае.

Ответ 1 быстрее приводит к противоречию, поэтому мы выберем $a_1 > 2$. Рассмотрим конец игры. Как и раньше, противоречие не может быть в дизъюнкте, который соответствует клетке, поскольку Делэер не дает посадить кролика в клетку, в которой уже кто-то сидит.

Значит, противоречие случилось в дизъюнкте, который соответствовал кролику: $p_{i,1} \vee p_{i,2} \vee \cdots \vee p_{i,n}$. Раз такое противоречие произошло, то оказалось так, что Делэер не смог ответить 1 в тот момент, когда случилось $|J_i(\alpha)| \geq n/2$. Это могло произойти только в том случае, если в клетке, про которую был запрос уже кто-то сидит. Значит, есть момент, в который есть как минимум $\frac{n}{2} - 1$ занятая клетка и, поскольку Делэер не дает пруверу сажать двух кроликов в одну клетку, то эти клетки заняты разными кроликами.

Мы рассмотрим эти $\frac{n}{2} - 1$ клеток и кроликов в них. И для каждой клетки посмотрим, кто сажал в нее кролика. В первом случае этого кролика сажал Прувер, это значит, что за это Делэер получил $\log a_1$ очков. Во втором случае этого кролика сажал Делэер. А это значит, что в тот момент $|J_i(\alpha)| \geq n/2$, и все эти нули мог ставить только Прувер, поскольку нули Делэер ставит только по причине того, что либо клетка уже кем-то занята, либо кролик где-то сидит. Значит, за эту клетку Делэер получил как минимум $\frac{n}{2} \log a_0$ очков. Итого, мы получили, что Делэер заработает как минимум $(\frac{n}{2} - 1) \min\{\log a_1, \frac{n}{2} \log a_0\}$ очков. Осталось подобрать нужные a_1 и a_0.

Мы хотим, чтобы $\log a_1 = \Omega(\log n)$ и при этом $\log a_0 = \Omega(\log n/n)$. $a_0 = a_1/(a_1 - 1) = 1 + 1/(a_1 - 1) \geq e^{1/(a_1-1)}$. Выберем $1/(a_1 - 1) = \log n/n$, т.е. $a_1 = n/\log n + 1$, тогда $\log a_0 \geq \Omega(\log n/n)$, а $\log a_1 = \Omega(\log n)$. □

3 Метод резолюций

3.1 Ветвящиеся программы

Если мы хотим усилить деревья решений, то естественной идеей является такая: в деревьях решений могут быть вершины с одинаковыми поддеревьями, такие вершины можно склеивать. Т.е. вместо деревьев решений у нас будут ветвящиеся программы.

Ветвящаяся программа для формулы ϕ в КНФ называется ориентированный граф без циклов с одним входом (вершиной входящей степени ноль), в которой все вершины кроме выходов (вершин исходящей степени ноль) помечены переменными формулы ϕ, при этом из вершины, помеченной переменной x ровно два ребра, одно из них помечено подстановкой $x = 0$, другое $x = 1$. Каждый выход помечен каким-нибудь дизъюнктом формулы ϕ и для каждого набора значений переменных, если выйти из входа программы и идти по ребрам, согласованным с данным набором, то такой путь закончится в выходе, в котором написан дизъюнкт, который ложен для данного набора значений переменных. Размером ветвящейся программы мы называем число вершин графа.

Отметим, что не каждому пути из входа ветвящейся программы соответствует подстановка, поскольку в определении не запрещается тестировать переменную несколько раз на одном пути. Но можно рассматривать только консистентные пути, в которой одной переменной не

подставляются разные значения. Каждому набору значений переменных соответствует некоторый консистентный путь от входа к выходу. Поэтому достаточно проверить, что подстановка вдоль любого консистентного пути от входа к выходу опровергает дизъюнкт, записанный в выходе.

Нетрудно понять, что если у формулы ϕ есть ветвящаяся программа, то она невыполнимая, поскольку каждый набор значений переменных приведет в дизъюнкт, который опровергается этим набором. Кроме того дерево решений тоже является ветвящейся программой, поэтому ветвящаяся программа существует для любой невыполнимой формулы. Однако ветвящиеся программы не являются системой доказательств, поскольку по помеченному графу трудно проверить, что он является корректной ветвящейся программой для формулы. Мы не можем проверять все наборы значений переменных, поскольку число путей может быть гораздо больше, чем размер ветвящейся программы. Чтобы убедиться в этом поймем, что для невыполнимой формулы ϕ всегда существует ветвящаяся программа размера $O(\phi)$. Эта программа устроена следующим образом: пусть формула $\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$, Ветвящаяся программа будет иметь выделенные вершины $v_1, v_2, \ldots, v_m$ (это не все вершины, а некоторые), при этом вершина v_1 будет входом ветвящейся программы. В вершине v_i для $i \in [m-1]$ будет запрос к первой переменной дизъюнкта C_i, по значению, которое этот дизъюнкт выполняет, мы переходим в v_{i+1}, по противоположному значению мы переходим в вершину, помеченную второй переменной C_i, по значению, которое выполняет C_i мы переходим в v_{i+1}, а по значению, которое не выполняет C_i переходим в вершину, помеченную третьей переменной C_i и т.д., если все переменные закончились, то это очередная вершина будет выходом, помеченным дизъюнктом C_i. Вершина v_m будет выходом, помеченным дизъюнктом C_m, поскольку, раз формула невыполнимая и ни один из предыдущих дизъюнктов не опровергся, то обязательно опровергается последний дизъюнкт.

Чтобы ветвящиеся программы можно было бы эффективно проверять на корректность, мы потребуем дополнительное ограничений, одноразовость программы. Ветвящаяся программа называется одноразовой, если на каждом пути от входа до выхода каждая переменная как пометка вершины встречается не более одного раза. Нетрудно понять, что минимальное дерево решений обязательно является одноразовой ветвящейся программой.

Оказывается, корректность одноразовой программы для формулы ϕ можно проверить за полиномиальное от размера формулы ϕ время. Мы должны проверить выполнение трех свойств: 1) То, что граф является ориентированным графом без циклов. На самом деле это проверять необязательно, достаточно требовать, чтобы нам вместе с графом предоставили его топологическую сортировку, т.е. нумерацию вершин, при которой ребра идут от вершин с меньшими номерами к вершинам с большими номерами. 2) То, что на любом пути от входа до выхода каждая переменная встречается один раз. Для каждой переменной можно посчитать список переменных, которые можно увидеть, выходя из этой вершины. Легче всего это сделать, начав обход от вершины с максимальным номером, в обратном порядке. Такие списки для очередной вершины получаются, как объединение списков для потомков. Имея такие списки легко проверить свойство одноразовости. 3) Вместо того, чтобы проверять, что для каждого набора значений переменных мы найдем дизъюнкт, который этим набором опровергается, мы будем проверять, что каждый путь из входа до выхода противоречит дизъюнкту, который в этом выходе написан. Проверку этого свойства мы ненадолго отложим. Для него нам потребуется определить резолюционную систему доказательств.

3.2 Резолюции

Говорят, что из дизъюнктов $A \vee x$ и $B \vee \neg x$ по правилу резолюции можно вывести дизъюнкт $A \vee B$. Говорят, что дизъюнкты $A \vee x$ и $B \vee \neg x$ резольвируются по переменной x. Правило резолюции обладает таким свойством, если какой-то набор значений переменных выполняет обе посылки правила, то этот же набор выполняет и заключение правила. Или наоборот, если какой-то набор значений переменных опровергает заключение правила, то он же опровергает и одну из посылок.

Резолюционным доказательством невыполнимости формулы ϕ является вывод пустого дизъюнкта (тождественно ложного) из дизъюнктов формулы ϕ по правилам резолюции. Обычно доказательство представляется в виде ориентированного графа без циклов, выходами которого помечены дизъюнкты формулы ϕ, каждая из остальных вершин помечена каким-нибудь дизъюнктом, причем из каждой вершины исходит два ребра, и дизъюнкт в вершине — это результат применения правила резолюции к дизъюнктам, которыми помечены концы исходящих ребер. Один из входов графа помечен пустым дизъюнктом. Размером резолюционного доказательства мы называем число вершин в графе.

Если резолюционное доказательство формулы ϕ существует, то формула ϕ невыполнимая. Действительно, пустой дизъюнкт опровергается любым набором значений переменных, значит, для каждого набора хотя бы один из дизъюнктов, из которых этот пустой дизъюнкт получился, тоже опровергается, значит опровергается хотя бы одна из посылок для того дизъюнкта и т.д., получаем, что один из дизъюнктов формулы ϕ опровергается этим набором.

Резолюционное доказательство нетрудно проверить за полиномиальное от его размера время. Нам еще нужно убедиться в полноте этой системы доказательств, т.е. в том, что у любой невыполнимой формулы есть резолюционное доказательство. Это мы покажем, указав связь между деревьями решений и ветвящимися программами и резолюционными доказательствами.

Если граф доказательства является деревом, то такое доказательство называют *древовидным*. Если на любом пути от входа до выхода правила резолюции применяются по различным переменным, то такое доказательство называют *регулярным*.

3.2.1 Построение ветвящейся программы по резолюционному доказательству

По резолюционному доказательству формулы ϕ легко построить ветвящуюся программу для формулы ϕ.

1. Мы удалим все входы, кроме одного, помеченного пустым дизъюнктом. После этого мы повторим это, пока не останется граф только с одним входом.

2. Развернем стрелки в графе. Теперь у графа будет единственный вход, помеченный пустым дизъюнктом, это будет входом ветвящейся программы. Каждый выход помечен дизъюнктами формулы ϕ.

3. Каждую вершину кроме выходов пометим переменной, по которой происходит применение правила резолюции в данной вершине.

4. Пусть из вершины помеченной $A \vee B$ идут ребра в вершины, помеченные $A \vee x$ и $B \vee \neg x$. Пометим ребро из $A \vee B$ в $A \vee x$ подстановкой $x = 0$, а ребро из $A \vee B$ в $B \vee \neg x$ подстановкой $x = 1$.

Заметим, что мы строили ветвящуюся программу так, чтобы подстановка вдоль любого консистентного пути от входа до вершины опровергала дизъюнкт в этой вершине. Это верно для пустого дизъюнкта во входе, и это сохраняется при увеличении пути на одно ребро. Значит, в выходе подстановка тоже опровергает дизъюнкт (а там записан дизъюнкт формулы ϕ). Таким образом мы получили корректную ветвящуюся программу.

Важно отметить, что описанное преобразование сохраняет граф. Т.е. из древовидного резолюционного доказательства мы построим дерево решений, а из регулярного резолюционного доказательства мы построим одноразовую ветвящуюся программу.

3.2.2 Построение регулярного резолюционного доказательства по одноразовой ветвящейся программе

Иногда в резолюционном выводе разрешают использовать еще одно правило: правило ослабления. Это правило позволяет вывести из дизъюнкта A дизъюнкт $A \vee \ell$ для любого литерала ℓ.

Упражнение 3.1. Покажите, что если формула ϕ имеет опровержение из S дизъюнктов, которое использует правило резолюции и правило ослабления, то ϕ имеет опровержение из не более, чем S дизъюнктов, которое использует только правило резолюции.

Теперь мы покажем, что для деревьев решений и одноразовых ветвящихся программ возможно и обратное построение. Тем самым мы докажем, что любая невыполнимая формула имеет резолюционное доказательство.

Мы будем доказывать для одноразовых ветвящихся программ. Дерево решений может не являться одноразовой программой (если мы два раза спрашиваем значение какой-то переменной), но в таком случае его можно сократить.

Будем строить резолюционное доказательство по одноразовой ветвящейся программе для формулы ϕ. Конструкцию удобно начинать с выходов, они уже помечены дизъюнктами формулы ϕ, мы пометим дизъюнктами все оставшиеся вершины программы так, чтобы дизъюнкт опровергался подстановкой от входа программы до текущей вершины. Если это свойство будет выполняться для входа, то во входе будет написан пустой дизъюнкт. Это построение удобно делать в обратном топологическом порядке, от выходов. Рассмотрим вершину v ветвящейся программы, пусть из нее выходят ребра в вершины u и w, которые уже помечены дизъюнктами C_u и C_w. Пусть переменная v помечена переменной x, ребро (v, u) помечено подстановкой $x = 0$, а ребро (v, w) помечено подстановкой $x = 1$. Мы знаем, что C_u не содержит $\neg x$, а C_w не содержит x, иначе бы подстановки выполняли бы дизъюнкт. Если C_u содержит x, а C_w содержит $\neg x$, то пометим вершину v результатом применения правила резолюции, примененному к C_v и C_w. Если же C_u не содержит x, то C_u, то мы можем с помощью правила ослабления вывести дизъюнкт $C_u \vee \neg x$, он тоже опровергается опровергается подстановкой на пути от корня до u, аналогично из C_w, если он не содержит x, можно ослаблением вывести $C_w \vee x$, а дальше v пометить результатом применения правила резолюции к этим дизъюнктам, как и в предыщум случае. Заметим, что дизъюнкт в вершине v также опровергается путем до этой вершины.

Нетрудно проверить, что из дерева решений мы получим древовидное резолюционное доказательство, а из одноразовой ветвящейся программы мы получим регулярное резолюционное доказательство.

Также заметим, что построение резолюционного доказательства по одноразовой ветвящейся программе помогает нам проверить одноразовую ветвящуюся программу на корректность. Для этого достаточно восстановить по одноразовой ветвящейся программе регулярное резолюционное доказательство и проверить его. Если в итоге будет выведен пустой дизъюнкт, то ветвящаяся программа корректна.

3.3 Ширина резолюционных доказательств

Мы введем еще одну меру сложности формулы — это минимальная ширина доказательства. Шириной дизъюнкта мы называем число литералов в нем (мы считаем, что дизъюнкт не содержит несколько литералов по одной переменной), шириной резолюционного доказательства мы называем максимальную ширину дизъюнкта, который в этом доказательстве использовался. Для невыполнимой формулы ϕ мы будем обозначать $w_R(\phi)$ минимальную возможную ширину резолюционного доказательства формулы ϕ.

Предложение 3.5. Для невыполнимой формулы ϕ выполняется неравенство $S_R(\phi) \leq (2n + 1)^{w_r(\phi)}$, где n — число переменных формулы ϕ.

Доказательство. Число дизъюнктов ширины не более k можно оценить сверху так: есть k мест для литералов, на каждое место можно поставить один из $2n$ литералов или не поставить ничего, т.е. таких дизъюнктов не больше $(2n + 1)^k$. □

Тем самым, если $w_R(\phi)$ маленькая (например, константная), то и размер доказательства маленький. Более того, если известно, что ширина доказательства не превосходит константы , то такое доказательство можно найти за полиномиальное время таким алгоритмом: будем выводить с помощью правила резолюции все дизъюнкты ширины не более c, если они новые. Число различных дизъюнктов оценено в предыдущем предложении. Например, если исходная формула в 2-КНФ, то такой алгоритм будет работать полиномиальное время, так как все выводимые дизъюнкты будут содержать не более двух литералов. Таким образом, мы построили алгоритм, проверяющий выполнимость формулы в 2-КНФ: если вывелся пустой дизъюнкт, то формула невыполнима, если не вывелся, то выполнима.

Позже мы покажем, что если у формулы ϕ существует короткое резолюционное доказательство, то и $w_R(\phi)$ маленькая. А сейчас мы предъявим пример формулы, у которой можно доказывать нижнюю оценку на минимальную ширину доказательства.

3.4 Цейтинские формулы

Цейтинская формула $T(G, f)$ строится по простому неориентированному графу $G(V, E)$. каждому ребру $e \in E$ соответствует переменная x_e, есть функция $f : V \to \{0, 1\}$, для каждой вершины $v \in E$ записывается формула в КНФ, кодирующая $\sum\limits_{u \in V : (u,v) \in E} p_{(u,v)} \bmod 2 = f(v)$, где $\oplus$ используется для обозначения суммы по модулю 2. Конъюнкция этих формул называется цейтинской формулой. Заметим, что если степень вершин G ограничена константой d, то $T(G, f)$ — это формула в d-КНФ, в которой число дизъюнктов не превосходит $2^d|V|$.

Если для какой-то компоненты связности $U \subseteq V$ выполняется $\sum\limits_{v \in U} f(v) \bmod 2 = 1$, то цейтинская формула невыполнима $T(G, f)$. Действительно, достаточно просуммировать (по модулю два) все равенства, которые записаны в вершинах и получится $0 = 1$, поскольку каждая переменная будет встречаться ровно два раза.

Если же для каждой компоненты связности $\sum_{v \in V} f(v) \bmod 2 = 0$, то цейтинская формула выполнима: присвоим всем переменным значения 0, равенства не будут выполняться в четном числе вершин для каждой компоненты связности. Рассмотрим две вершины из одной компоненты связности, в которых не выполняются равенства, и заменим значения переменных на противоположные вдоль пути между этими двумя вершинами. Заметим, что после такой операции число вершин, в которых не выполняется равенство, уменьшится на 2. Осталось повторить эту операцию, пока равенство не начнет выполняться во всех вершинах.

Для неориентированного графа $G(V, E)$ и для двух дизъюнктных множеств $A, B \subseteq V$ обозначим через $E(A, B)$ множество ребер, в котором один конец лежит в A, другой в B. Расширительной способностью графа G называется число $e(G)$, которое равняется минимальному значению $|E(U, V \setminus U)|$ по всем $U \subseteq V$, что $\frac{|V|}{3} \leq |U| \leq \frac{2|V|}{3}$.

Теорема 3.1. Пусть $T(G, f)$ — невыполнимая цейтинская формула, построенная по связному графу G. Тогда $w_R(T(G, f)) \geq e(G)$.

Доказательство. Введем меру μ на множестве дизъюнктов. $\mu(C)$ равняется минимальному множеству вершин графа G, что из условий четности в этих вершинах семантически следует дизъюнкт C. Для дизъюнктов формулы $T(G, f)$ мера равняется 1. Мера пустого дизъюнкта равняется n, поскольку все условия цейтинской формулы кроме одного в связном графе можно выполнить, так как у нас есть ребра, которые участвуют только один раз, значит в каждой компоненте связности, которые получаются при удалении вершины, можно сделать четную сумму пометок.

Нетрудно убедиться, что мера μ полуаддитивна, т.е., что если дизъюнкт C получился по правилу резолюции из дизъюнктов A и B, то $\mu(C) \leq \mu(A) + \mu(B)$. Действительно, дизъюнкт C всегда следует из условий для объединения множества вершин, из которых следует A и B.

Рассмотрим некоторое доказательство формулы $T(G, f)$. В этом доказательстве рассмотрим дизъюнкт C с мерой $n/3 \leq \mu(C) \leq 2n/3$. Такой есть из того, что мера дизъюнктов формулы равны 1, мера пустого дизъюнкта равна n и свойства полуаддитивности. Пусть дизъюнкт C семантически следует из условий для вершин множества $U \subseteq V$ и $|U| = \mu(C)$. Мы покажем, что для любого ребра $e \in E(U, V \setminus U)$ переменная x_e содержится в дизъюнкте C, из этого будет следовать, что $|C| \geq |E(U, V \setminus U)| \geq e(G)$, что и требуется доказать.

Рассмотрим $e \in E(U, V \setminus U)$, пусть $v \in U$ — конец ребра e. Мы знаем, что из множества вершин $U \setminus \{v\}$ семантически не следует дизъюнкт C, следовательно существует набор значений переменных σ, который выполняет все условия $U \setminus \{v\}$, но не выполняет дизъюнкт C. Набор σ не выполняет условие в вершине v, так как иначе бы он выполнял бы все условия для вершин из U, а следовательно выполнял бы и C. Если x_e не входит в C, то можно поменять значение переменной x_e в σ, чтобы получившийся набор σ_e выполнил бы условие в вершине v (поскольку условие — это просто условие о четности). Поскольку ребро e не имеет больше концов в U, то σ_e выполняет все U, но все еще не выполняет C. Противоречие, поэтому x_e должно входить в C. □

Существует такое семейство графов G_n на n-вершинах (экспандеры) у таких графов степень ограничена константой d (это значит, что цейтинская формула в d-КНФ и сама имеет маленькую ширину), но при этом $e(G) = \Omega(n)$. Формула $T(G_n, f)$ имеет $O(n)$ переменных и резолюционную ширину $\Omega(n)$.

3.5 Размер и ширина резолюционных доказательств

Пусть ϕ — невыполнимая формула в КНФ. Введем несколько обозначений:

- $S_T(\phi)$ — размер минимального древовидного резолюционного доказательства для формулы ϕ. Как мы знаем, с точностью до мультипликативной константы $S_T(\phi)$ совпадает с размером минимального дерева решений для ϕ.
- $S_R(\phi)$ — минимальный размер резолюционного доказательства формулы ϕ.

Поскольку древовидное доказательство является частным случаем обычного, то всегда выполняется неравенство $S_T(\phi) \geq S(\phi)$.

Лемма 3.1. Пусть ϕ — невыполнимая формула в k-КНФ, x — переменная формулы ϕ, $a \in \{0,1\}$. Тогда если для некоторого натурального числа w выполняется $w_R(\phi|_{x=a}) \leq w-1$ и $w_R(\phi|_{x=1-a}) \leq w$, то $w_R(\phi) \leq \max\{k,w\}$.

Доказательство. Пусть x^1 обозначает x, а x^0 обозначает $\neg x$. Рассмотрим вывод $\phi|_{x=a}$ ширины не более $w-1$, этот вывод легко перестроить либо в вывод x^{1-a} либо в вывод пустого дизъюнкта из ϕ ширины w. Для этого достаточно вернуть литералы x^{1-a} во все дизъюнкты формулы $\phi|_{x=a}$, которые получились выкидыванием из дизъюнктов ϕ литерала x^{1-a}. Если получился вывод пустого дизъюнкта, то на этом можно закончить. Используя дизъюнкт x^{1-a}, можно вывести все дизъюнкты формулы $\phi|_{x=1-a}$ с помощью вывода ширины не более k. А после этого можно воспользоваться выводом пустого дизъюнкта из формулы $\phi|_{x=1-a}$ ширины w. Итого, мы получили вывод пустого дизъюнкта из ϕ ширины не более $\max\{k,w\}$. □

Теорема 3.2 (Бен-Сассон, Вигдерсон [BSW01])**.** Для каждой невыполнимой формулы ϕ в k-КНФ выполняются неравенства:

1. $S_T(\phi) \geq 2^{w_R(\phi)-k}$
2. $S_R(\phi) \geq 2^{\frac{(w_R(\phi)-k)^2}{8n}}$

Доказательство. 1. Докажем по индукции по числу переменных. Удобнее доказывать утверждение для минимального размера дерева решений. База индукции для формулы из 1 переменной, чтобы она была невыполнимой, формула должна содержать дизъюнкты x и $\neg x$. Размер минимального дерева решений равен 3, а ширина 1.

Пусть в корне минимального дерева решений для формулы ϕ стоит расщепление по переменной x, то в одной из ветви стоит минимальное дерево решений для формулы $\phi|_{x=0}$, а в другом минимальное дерево решений для формулы $\phi|_{x=1}$. Т.е. $S_T(\phi) = S_T(\phi|_{x=0}) + S_T(\phi_{x=1})$.

Поймем, что в случае, когда $w_R(\phi) \leq k$, утверждение очевидно, поэтому далее будем считать, что $w_R(\phi) \geq k+1$.

Нетрудно понять, что $w_R(\phi|_x = b) \leq w_R(\phi)$ для всех $b \in \{0,1\}$. Из леммы 3.1 следует, что не может такого быть, что $w_R(\phi|_x = b) \leq w_R(\phi)-2$ для всех $b \in \{0,1\}$, так как из этого следовало бы, что $w_R(\phi) \leq \max k, w_R(\phi)-1$. Аналогично не может быть, что $w_R(\phi|_x = b) \leq w_R(\phi)-1$ и $w_R(\phi|_x = 1-b) \leq w_R(\phi)-2$ для некоторого $b \in \{0,1\}$. Значит, есть два варианта, либо а) $w_R(\phi|x=0) = w_R(\phi|_{x=1}) = w_R(\phi)-1$, либо б) $w_R(\phi|_{x=a}) = w_R(\phi)$ для некоторого $a \in \{0,1\}$. Разберем эти два случая отдельно.

а) $S_T(\phi) \geq S_T(\phi|_{x=0}) + S_T(\phi|_{x=1}) \geq 2^{w_R(\phi)-1-k} + 2^{w_R(\phi)-1-k} = 2^{w_R(\phi)-k}$. Во втором неравенстве мы дважды применили индукционное предположение.

б) $S_T(\phi) \geq S_T(\phi|_{x=a}) \geq 2^{w_R(\phi)-k}$. Во втором неравенстве мы применили индукционное предположение.

2. Пусть $W \geq 1$ — некоторый параметр, значение которого мы подберем позже. Мы будем говорить, что дизъюнкт толстый, если в нем как минимум W литералов.

Утверждение 1. Пусть F — множество толстых дизъюнктов от n переменных. Тогда существует такой литерал ℓ, который входит в как минимум $\frac{W}{2n}|S|$ дизъюнктов из множества S.

Доказательство. Всего есть $2n$ литералов, если бы каждый литерал входил в меньше, чем $\frac{W}{2n}|S|$ дизъюнктов, то общее число литералов можно было бы менее $W|S|$. Но в каждый толстый дизъюнкт входит как минимум W литералов, следовательно общее число литералов должно быть не менее $W|S|$, противоречие. □

Из утверждения 1 следует, что существует литерал, выполнив который, выполнится как минимум доля $\frac{W}{2n}$ толстых дизъюнктов. Т.е. после подстановки, которая выполняет этот литерал, останутся невыполненными не более доли $\left(1 - \frac{W}{2n}\right)$ толстых дизъюнктов.

Обозначим $a = \left(1 - \frac{W}{2n}\right)^{-1} \geq e^{W/2n}$.

Утверждение 2. Если невыполнимая k-КНФ формула содержит резолюционное опровержение, в котором не более a^b толстых дизъюнктов, то $w_R(\phi) \leq W + b + k$.

Доказательство. Доказательство по индукции по b и числу переменных n. База $n = 1$ очевидна. Проверим базу для $b = 0$, в этом случае доказательство либо не содержит толстых дизъюнктов совсем, тогда его ширина не больше, чем $\max\{W, k\} \leq W + k$, либо содержит всего один толстый дизъюнкт. Заметим, что его ширина не может быть хотя бы $W + 1$, так как если мы применим к этому дизъюнкту какое-то правило, то его ширина уменьшится не более, чем на 1, т.е. есть еще хотя бы один толстый дизъюнкт. Еще возможен вариант, что этот дизъюнкт не используется в доказательстве, тогда его можно просто удалить. Значит, ширина доказательства не больше $\max\{W + 1, k\} \leq W + k$, поскольку $k \geq 1$.

Теперь допустим, что утверждение выполняется для всех меньших значений n и b. По утверждению 1 существует литерал x, который входит в как минимум долю $1 - \frac{1}{a}$ всех толстых дизъюнктов резолюционного доказательства. Если мы сделаем подстановку $x = 1$ в доказательство, то получим доказательство формулы $\phi|_{x=1}$, в котором будет не более a^{b-1} толстых дизъюнктов. Тогда по индукционному предположению $w_R(\phi|_{x=1}) \leq W + b + k - 1$. Применяя индукционное предположение по числу переменных, получаем $w_R(\phi|_{x=0}) \leq W + b + k$. Тогда по лемме 3.1 $w_R(\phi) \leq W + b + k$. □

Выберем b такое, что $a^b = S_R(\phi)$. Тогда

По утверждению 2 мы получим, что $w_R(\phi) \leq W + b + k$.

$$b = \frac{\log S_R(\phi)}{\log a} \leq \frac{2n \log S_R(\phi)}{W \log e} \leq \frac{2n \log S_R(\phi)}{W}$$

Итого, $w_R(F) \leq W + b + k \leq W + \frac{2n \log S_R(\phi)}{W} + k$. Сумма $W + \frac{2n \log S_R(\phi)}{W}$ минимальна, когда $W = \sqrt{2n \log S_R(\phi)}$. Получаем, что $w_R(\phi) \leq 2\sqrt{2n \log S_R(\phi)} + k$. Отсюда получаем требуемую оценку. □

Из доказанной теоремы следует, что для того, чтобы доказать нижнюю оценку на размер резолюционного доказательства формулы в k-КНФ для небольшого k достаточно доказать нижнюю оценку на ширину $w_R(\phi) = \Omega(n^{\frac{1}{2}+\epsilon})$ для некоторого $\epsilon > 0$. В частности, для цейтинских формул, построенных по экспандерам константной степени получается нижняя оценка $2^{\Omega(n)}$ на размер доказательств, поскольку у таких графов $e(G) = \Omega(n)$.

3.6 Принцип Дирихле для графа

Рассмотрим обобщение принципа Дирихле, в котором не любой кролик может сидеть в любой клетке. У нас есть двудольный граф $G(V, E)$, с двумя долями P и H, m вершин в доле P и n вершин в доле H. Переменными формулы являются переменные $p_{i,j}$, если $(i, j) \in E$. Т.е. кролик может сидеть только в клетках, с которым он соединен ребром. Формула PHP_G содержит такие дизъюнкты:

- Кроличьи аксиомы: $\bigwedge_{j:(i,j)\in E} p_{i,j}$ для всех $i \in P$;
- Аксиомы клеток: $\neg p_{i,k} \vee \neg p_{j,k}$, для всех $k \in H$ и $i, j \in P$, смежных с k.

Формула PHP_G невыполнима тогда и только тогда, когда в графе G нет паросочетания размера $|P|$.

Зафиксируем двудольный граф $G(P, H, E)$. Пусть $A \subseteq P$, тогда границей множества A называется число $\delta(A)$ вершин из H, из которых исходит ровно одно ребро в множество A. Двудольный граф $G(P, H, E)$ называется (r, c) — граничным экспандером, если для любого множества $A \subseteq P$, если $|A| \leq r$, то $\delta(A) \geq c|A|$.

Теорема 3.3. Пусть двудольный граф $G(P, H, E)$ является (r, c) — граничным экспандером для $c \geq 1$. Пусть в графе G не существует паросочетания размера $|P|$. Тогда $w_R(PHP_G) \geq rc/2$.

Доказательство. Назовем выполняющий набор формулы PHP_G допустимым, если он выполняет все аксиомы клетки, т.е., что в каждой клетке сидит не более одного кролика. Будем говорить, что дизъюнкт допустимо следует из множества кроликов, если любой допустимый набор, который выполняет все кроличьи аксиомы для кроликов из этого множества, также выполняет и этот дизъюнкт. Введем меру μ на множестве дизъюнктов. Мера дизъюнкта — это размер наименьшего множества кроликов, из которого этот дизъюнкт допустимо следует. Мера кроличьей аксиомы равняется 1, мера аксиомы клетки равняется нулю.

Утверждение 3. Для любого множества $A \subseteq P$, если $|A| \leq r$, то существует допустимый набор, который выполняет все кроличьи аксиомы для кроликов из множества A.

Доказательство. Для множества A выполняется условие теоремы Холла о паросочетании, поскольку для любого $B \subseteq A$ множество соседей множества B не меньше, чем $|\delta(B)| \geq c|B| \geq |B|$. Следовательно в графе существует паросочетание, которое покрывает все вершины множества A. Осталось рассмотреть допустимый набор, который соответствует этому паросочетанию. □

Из доказанного утверждения следует, что мера пустого дизъюнкта не меньше r. Легко проверяется, что мера μ является полуаддитивной. Значит, в любом резолюционном доказательстве формулы PHP_G есть такой дизъюнкт C, что $r/2 \leq \mu(C) < r$.

Мы покажем, что в дизъюнкт C входит как минимум $\mu(C)c/2 \geq rc/4$ ребер. Пусть дизъюнкт C допустимо следует из множества кроликов $A \subseteq P$ и $|A| = \mu(C)$. Рассмотрим ребро

(v, u), которое соединяет вершину $v \in A$ с вершиной $u \in \delta(A)$. Сопоставим этому ребру уникальную переменную дизъюнкта C. Поскольку таких ребер (u, v) не меньше, чем $|\delta(A)| \geq c|A|$, мы получим, что дизъюнкт C содержит как минимум $cr/4$ переменных.

Поскольку из множества $A \setminus \{v\}$ допустимо не следует дизъюнкт C, это значит, что существует такой допустимый набор значений переменных σ, что σ выполняет все условия для кроликов из множества $A \setminus \{v\}$, но не выполняет дизъюнкт C. Поскольку из A допустимо следует C, а σ не выполняет C, то σ не выполняет условие для кролика v. Рассмотрим набор σ_v, который отличается от σ тем, что значение переменной $x_{(v,u)}$ теперь равно 1. Если набор σ_v допустимый, то поскольку σ_v выполняет A, следовательно σ_v выполняет C, это значит, что переменная $x_{(v,u)}$ входит в C. Пусть σ_v недопустимый, это значит, что σ присваивает 1 ребру (w, u) для некоторой вершины $w \in P$. Поскольку $u \in \delta(A)$, то $w \notin A$. Рассмотрим набор $\sigma_{v,w}$, который отличается от σ_v значением переменной $x_{(w,u)}$. Набор $\sigma_{v,w}$ допустимый и выполняет условие для всех кроликов из A, следовательно он выполняет дизъюнкт C, следовательно, либо $x_{(v,u)}$, либо $x_{(w,u)}$ входит в C. При этом вершина u однозначно определяется по этим переменным. □

Если граф G является (r, c)-граничным экспандером, где $r = \Omega(n)$, $c > 1$, и степень любой вершины из P ограничена константой, то к такому графу можно применить теорему Бен-Сассона Вигдерсона и получить, что резолюционная сложность есть $2^{\Omega(n)}$ (n — это число клеток, размер H). (r, c)-граничные экспандеры константной степени существуют, например, если $|P| = n + 1$, а $|H| = n$, то если случайным образом выпустить d ребер (d — это константо) из каждой вершины P в H, то можно проверить, что получившийся граф будет с (r, c)-экспандером для $r = \Omega(n/d)$ и $c = \frac{9}{10}d$ с большой вероятностью. И для такого графа у нас есть нижняя оценка на сложность резолюционного вывода $2^{\Omega(n)}$. Заметим, что формулу PHP_G можно получить из PHP_n^{n+1} подстановкой нулей вместо отсутствующих ребер, следовательно мы получили нижннюю оценку $2^{\Omega(n)}$ на вывод PHP_n^{n+1}. Это оценка точная, так как известна верхняя оценка $2^{O(n)}$.

3.7 Игровая интерпретация ширины

Атсериас и Далмау [AD08] предложили следующую интерпретацию ширины резолюционного доказательства. Пусть k — некоторое натуральное число, параметр игры. ϕ — невыполнимая формула в КНФ. Алиса и Боб играют в следующую игру, у них есть доска, на которую помещается не более k равенств вида $x = a$, где x — переменная формулы ϕ, а $a \in \{0, 1\}$, причем все переменные должны быть различные. Изначально доска пустая. За один ход Алиса может либо стереть одно равенство с доски, либо, если на доске есть свободное место, то написать туда переменную, которая еще не встречается на доске, Боб должен написать значение этой переменной. Боб проигрывает, если написанные на доске равенства противоречат дизъюнкту формулы ϕ. Выигрышной стратегией для Боба назовем непустое множество частичных подстановок $\mathcal{H}$, которые удовлетворяют следующим свойствам:

1. Для каждого дизъюнкта C формулы ϕ и каждого элемента $f \in \mathcal{H}$, частичная подстановка f не опровергает C и $|f| \leq k$, где $|f|$ — размер частичной подстановки f.
2. Если $f \in \mathcal{H}$ и $g \subseteq f$, то $g \in \mathcal{H}$.
3. Если $f \in \mathcal{H}$ и $|f| < k$, то для любого x, которому f не подставляет значение, существует такое $a \in \{0, 1\}$, что $f \cup \{x = a\} \in \mathcal{H}$.

Лемма 3.2. Если у невыполнимой формулы ϕ в КНФ есть выигрышная стратегия для Боба в игре с параметром k, то $w_R(\phi) \geq k$.

Доказательство. Пусть $\mathcal{H}$ — выигрышная стратегия Боба порядка k. Допустим, что у ϕ есть резолюционное доказательство $C_1, C_2, \ldots, C_s$ ширины не больше, чем $k-1$. Индукцией по i покажем, что каждая частичная подстановка $f \in \mathcal{H}$ не опровергает C_i. База, когда C_i — это дизъюнкт формулы ϕ. Индукционный переход, пусть $C_i = A \vee B$ и получается по правилу резолюции из $C_{i_1} = A \vee x$ и $C_{i_2} = B \vee \neg x$, где $i_1, i_2 < i$. Предположим противное и найдется $f \in \mathcal{H}$, что f опровергает $A \vee B$, f подставляет значение всем переменным C_i. Найдем $g \subseteq f$, что g подставляет значение только переменным C_i. Поскольку доказательство имеет ширину не более $k-1$, существует такое $a \in \{0,1\}$, что $g \cup \{x = a\} \in \mathcal{H}$, но $g \cup \{x = a\}$ обязательно опровергает либо C_{i_1}, либо C_{i_2}, что противоречит индукционному предположению.

Из доказанного следует, что C_s не может быть пустым дизъюнктом. □

Лемма 3.3. Если для невыполнимой формулы ϕ в r-КНФ выполняется $w_R(\phi) \geq k$, где $r \leq k-1$, то у Боба есть выигрышная стратегия в игре с параметром k.

Доказательство. Пусть $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ — множество дизъюнктов, которые выводимы из ϕ с помощью вывода ширины не более $k-1$. Мы знаем, что $\mathcal{C}$ не содержит пустого дизъюнкта. Определим $\mathcal{H}$ как множество частичных подстановок с носителем не больше k, которые не опровергают все дизъюнкты множества $\mathcal{C}$. $\mathcal{H}$ непусто, поскольку содержит пустую подстановку. Проверим, что $\mathcal{H}$ — выигрышная стратегия для Боба.

1. Если дизъюнкт C принадлежит формуле ϕ, то поскольку $r \leq k-1$, то $C \in \mathcal{C}$, по построению $\mathcal{H}$ никакой его элемент не опровергает C. Также все элементы $\mathcal{H}$ имеют носитель не больше k.

2. По построению очевидно, что, если $f \in \mathcal{H}$ и $g \subseteq \mathcal{H}$, то $g \in \mathcal{H}$.

3. Пусть $f \in \mathcal{H}$, $|f| \leq k-1$ и x не входит в носитель f. Проверим, что либо $f \cup \{x = 0\}$, либо $f \cup \{x = 1\}$ лежит в $\mathcal{H}$. От противного, пусть $f \cup \{x = 0\}$ не лежит в $\mathcal{H}$, это значит, что найдется дизъюнкт в $\mathcal{C}$, что $f \cup \{x = 0\}$ опровергает C. Поскольку f не опровергает C, то C имеет вид $A \vee x$. При этом f подставляет значение всем переменным дизъюнкта A. Аналогично есть дизъюнкт $B \vee \neg x$ в $\mathcal{C}$ и при этом f опровергает B, следовательно f подставляет значения всем переменным B. Поскольку носитель f не превосходит $k-1$, то число переменных $A \vee B$ тоже не превосходит $k-1$, $A \vee B$ является резолюцией $A \vee x$ и $B \vee \neg x$, следовательно $A \vee B$ принадлежит $\mathcal{C}$, но f противоречит $A \vee B$, получаем противоречие с определением $\mathcal{H}$.

□

3.8 Память и ширина

Допустим у нас ограничена память, которую мы можем использовать при доказательстве невыполнимости формулы ϕ. А именно, мы можем записывать ограниченное количество дизъюнктов. Реализацией доказательства называется последовательность $S_0, S_1, S_2, \ldots, S_t$, где S_i — это множество дизъюнктов, S_0 — это пустое множество, а S_t состоит из пустого дизъюнкта. S_{i+1} получается из S_i по одному из следующих правил:

- Загрузка дизъюнкта формулы ϕ: мы добавляем к S_i дизъюнкт формулы ϕ;
- Удаление дизъюнкта: удаляем дизъюнкт из S_i;
- Вывод дизъюнкта: добавление резольвенты двух дизъюнктов из S_i.

Сложностью реализации доказательства по памяти называется максимальный размер $|S_i|$. $CSpace(\phi)$ — это минимальная возможность сложность по памяти при реализации доказательства формулы ϕ.

Предложение 3.6. Для невыполнимой формулы ϕ в КНФ от n переменных выполняется неравенство $CSpace(\phi) \leq n + 2$.

Доказательство. Доказательство индукцией по числу переменных. Если из $\phi|_{x=0}$ есть вывод со сложностью по памяти $n+1$, то из ϕ есть вывод x (или пустого дизъюнкта) со сложностью по памяти $n + 1$, добавим этот вывод к выводу $\neg x$ из ϕ с памятью $n + 1$ и в конце применим правило резолюции для вывода пустого дизъюнкта. □

Игровая интерпретация ширины позволяет ограничить память через ширину снизу.

Теорема 3.4. Пусть ϕ — невыполнимая формула в r-КНФ. Тогда $CSpace(\phi) \geq w_R(\phi) - r + 1$.

Доказательство. Если $w_R(\phi) \leq r$, то утверждение теоремы очевидно. Далее будем считать, что $w_R(\phi) \geq r + 1$. По лемме 3.3 у Боба есть выигрышная стратегия в игре с параметром $w_R(\phi)$. Тогда теорема следует из следующего утверждения.

Утверждение 4. Если в игре с параметром $k + r - 1$ есть выигрышная стратегия Боба, то $CSpace(\phi) \geq k$.

Доказательство. Рассмотрим реализацию некоторого вывода, использующую память менее k: $S_0, S_1, \ldots, S_t$. Мы покажем, что для любого i можно выполнить все дизъюнкты S_i. Пусть $\mathcal{H}$ — это выигрышная стратегия Боба в игре с параметром $k+r-1$. Индукцией по i покажем, что существует $f_i \in \mathcal{H}$, который выполняет все дизъюнкты из S_i и при этом носитель f_i имеет разамер не более $|S_i|$. База $i = 0$ очевидна, так как пустая подстановка лежит в $\mathcal{H}$. Индукционный переход, рассмотрим случаи, как S_{i+1} получается из S_i:

- Пусть S_{i+1} получается загрузкой дизъюнкта C формулы ϕ. Пусть $f_i \in \mathcal{H}$, f_i выполняет все дизъюнкты S_i и $|f_i| \leq |S_i| \leq k - 1$. Поскольку в дизъюнкте C не более r переменных, то существует $f \in \mathcal{H}$, который продолжает f_i на все переменные дизъюнкта C. Поскольку f не может опровергать C, f выполняет C. Осталось выкинуть из f лишние подстановки, просто оставим по одной переменной на каждый дизъюнкт из S_{i+1} — той переменной, которая выполняет этот дизъюнкт.
- Пусть S_{i+1} получается из S_i удалением дизъюнкта. Тогда можно удалить одно равенство из f_i и получить f_{i+1}. Чтобы выполнить $|S_{i+1}|$ дизъюнктов, хватает подстановки не более $|S_{i+1}|$ переменных.
- Пусть S_{i+1} получается из S_i выводом нового дизъюнкта. Можно выбрать $f_{i+1} = f_i$.

□

□

4 Секущие плоскости

Пусть есть невыполнимая формула ϕ в КНФ. Будем представлять каждый ее дизъюнкт $\bigwedge_{i\in P} x_i \vee \bigwedge_{i\in N} \neg x_j$ в виде линейного неравенства $\sum_{i\in P} x_i + \sum_{i\in N}(1-x_i) \geq 1$. Также добавим неравенства, гарантирующие, что переменные принимают значения из $\{0,1\}$: $x_i \geq 0$ и $1 - x_i \geq 0$ для всех переменных.

Выводом противоречия в системе секущие плоскости (Cutting Plane, CP) называется вывод неравенства $0 \geq 1$ из указанных неравенств по правилу неотрицательной линейной комбинации и деления с округлением.

Предложение 4.7. Секущие плоскости моделируют резолюцию.

Доказательство. Будем выводить неравенства, представляющие дизъюнкты из резолюционного доказательства. Пусть у нас есть неравенство $f + x \geq 1$ и $g + (1 - x) \geq 1$, где f и g — сумма литералов. Из них выведем $f + g \geq 1$. Это то, что нужно, если переменные f и g не пересекаются. Пусть пересекаются, обозначим сумму общих литералов через h, т.е. $f = h + f'$, а $g = h + g'$, тогда мы вывели $2h + g' + h' \geq 1$, пользуясь тем, что для всех переменных $x \geq 0$ и $1 - x \geq 0$, выведем $g' \geq 0$ и $h' \geq 0$. Сложив эти неравенства, получим $2h + 2g' + 2h' \geq 1$. Применим деление с округлением и получим $h + g' + h' \geq 1$. □

Предложение 4.8. Существует доказательство PHP^{n+1}_n в древовидной CP размера $poly(n)$.

Доказательство. Для каждого кролика $i \in [n+1]$ будем выводить $p_{i,1} + \ldots p_{i,k} \leq 1$ для всех $k \in [n+1]$. Для $k = 2$ это неравенство есть и так. Покажем, как из неравества для k получить неравенство для $k+1$: домножим неравенство $p_{i,1} + \ldots p_{i,k} \leq 1$ на $(k-1)$ и получим $(k-1)p_{i,1} + \ldots (k-1)p_{i,k} \leq k-1$. Сложим это неравенство с $p_{i,j} + p_{i,k+1} \leq 1$ для всех $j \in [k]$ и получим $kp_{i,1} + \ldots kp_{i,k} + kp_{i,k+1} \leq 2k-1$. Применим правило деления с округлением и получим $p_{i,1} + \ldots p_{i,k} + p_{i,k+1} \leq 1$. □

4.1 Нижняя оценка

Экспоненциальная нижняя оценка была получена в работе Пудлака [Pud97]. Идеей доказательства было извлечение из доказательства монотонной булевой схемы и применении оценок на монотонную схемную сложность.

Булева схема — это такое вычислительное устройство, которое представляет собой ориентированный граф без циклов, в этом графе есть n вершин, которые мы называем входами схемы, в эти вершины ребра не входят, будем считать, что входы помечены $x_1, x_2, \ldots, x_n$. Каждая из вершин, которая не является входом, в которую входит k ребер помечена конкретной булевой функцией $\{0,1\}^k \to \{0,1\}$, одна из вершин схемы называется выходом. На вход схемы подаются 0/1 переменные, вершины схемы сортируются так, чтобы ребра вели из вершин с меньшими номерами в вершины с большими номерами (для ориентированных графов без циклов это всегда можно сделать), после этого одним проходом по списку вершин в каждой вершине можно посчитать значение с помощью функции, которая задана в этой вершине. Результатом схемы является значение, посчитанное в выходе схемы. Размером схемы мы будем называть количество вершин в графе, не считая входы.

Монотонной булевой схемой мы называем такие схемы, в которых в качестве операций в вершинах используются только бинарные конъюнкции и дизъюнкции. Известно, что монотонные схемы вычисляют монотонные булевы функции и только их.

Вещественные схемы отличаются от булевых тем, что в ее вершинах вычисляются функции из $\mathbb{R}^k \to \mathbb{R}$. Чтобы вещественная схема задавала булеву функцию необходимо также потребовать, чтобы функция в выходе принимала значения только из $\{0, 1\}$.

Монотонной вещественной схемой будем называть такую схему, которая вычисляет булевы функции, во всех вершинах вычисляются монотонные бинарные функции, т.е. такие функции $f : \mathbb{R}^2 \to \mathbb{R}$, что для всех $x_1, y_1, x_2, y_2 \in \mathbb{R}$, если $x_1 \leq y_1$ и $x_2 \leq y_2$, то $f(x_1, x_2) \leq f(y_1, y_2)$.

Пусть $A(x, y) \wedge B(x, z)$ — невыполнимая формула в КНФ. При каждом значении x, либо $A(x, y)$, либо $B(x, z)$ является невыполнимой.

Лемма 4.1. Существует алгоритм, который за полиномиальное время из СР доказательства формулы $A(x, y) \wedge B(x, z)$ для данного значения $x = a$ найдет либо СР доказательство формулы $A(a, y)$, либо СР доказательство формулы $B(a, z)$.

Доказательство. Будем считать, что все неравенства в доказательстве имеют вид $f \leq \alpha$. Каждому неравенству $f(x)+g(y)+h(z) \leq \alpha$ из доказательства мы сопоставим два неравенства: $g(y) \leq \beta$ и $h(z) \leq \gamma$. При этом мы потребуем, чтобы $\beta+\gamma+f(a) \leq \alpha$. Это условие гарантирует, что при $x = a$ исходное неравенство следует из двух новых.

Все исходные неравенства представляют либо дизъюнкты $A(x, y)$ и имеют вид либо $f(x) + g(y) \leq \alpha$, либо дизъюнкты $B(x, z)$ и имеют вид $f(x) + h(z) \leq \alpha$. В первом случае мы запишем такие два неравенства: $g(y) \leq \alpha - f(a)$ и $0 \leq 0$, а во втором случае $0 \leq 0$ и $h(z) \leq \alpha + f(a)$.

Пусть неравенство $f_1(x)\delta + f_2(x)\epsilon + g_1(y)\delta + g_2(y)\epsilon + h_1(z)\delta + h_2(z)\epsilon \leq \alpha_1\delta + \alpha_2\epsilon$ получено из $f_1(x) + g_1(y) + h_1(z) \leq \alpha_1$ и $f_2(x) + g_2(y) + h_2(z) \leq \alpha_2$ по правилу линейной комбинации. Пусть первому неравенству мы сопоставили неравенства $g_1(y) \leq \beta_1$ и $h_1(z) \leq \gamma_1$, причем выполняется $\beta_1 + \gamma_1 + f_1(a) \leq \alpha_1$, а второму неравенству мы сопосотавили неравенства $g_2(y) \leq \beta_2$ и $h_2(z) \leq \gamma_2$, причем выполняется $\beta_2 + \gamma_2 + f_2(a) \leq \alpha_2$. Тогда итоговому неравенству мы сопоставим неравенство $g_1(y)\delta + g_2(y)\epsilon \leq \beta_1\delta + \beta_2\epsilon$ и $h_1(z)\delta + h_2(z)\epsilon \leq \gamma_1\delta + \gamma_2\epsilon$, при этом неравенство $\beta_1\delta+\beta_2\epsilon+\gamma_2\epsilon+f_1(a)\delta+f_2(a)\epsilon \leq \alpha_1\delta+\alpha_2\epsilon$ выполняется как линейная комбинация аналогичных неравенств для посылок.

Пусть неравенство $f(x)+g(y)+h(z) \leq \lfloor \frac{\alpha}{d} \rfloor$ получено из $df(x)+dg(y)+dh(z) \leq \alpha$ по правилам деления с округлением. Пусть посылке мы сопоставили такие два неравенства: $dg(y) \leq \beta$ и $dh(z) \leq \gamma$, при этом $\beta+\gamma+df(a) \leq \alpha$. Сопоставим итоговому неравенству такие два: $g(y) \leq \lfloor \frac{\beta}{d} \rfloor$ и $h(z) \leq \lfloor \frac{\gamma}{d} \rfloor$. $\lfloor \frac{\beta}{d} \rfloor + \lfloor \frac{\gamma}{d} \rfloor + f(a) = \lfloor \frac{\beta+\gamma+df(a)}{d} \rfloor \leq \lfloor \frac{\alpha}{d} \rfloor$.

Доказательство заканчивалось неравенством $0 \leq -1$. Ему мы сопоставили два неравенства $0 \leq \alpha$ и $0 \leq \beta$, при этом $\alpha + \beta \leq -1$, так как α и β целые, то либо $\alpha \leq -1$, либо $\beta \leq -1$. Т.е. мы построили вывод, либо из дизъюнктов $A(a, y)$, либо из дизъюнктов $B(a, z)$. □

Теорема 4.1. [Пудлак, [Pud97]] Если формула $A(x, y)$ обладает свойством, что переменные x входит в нее только без отрицаний или формула $B(x, z)$ обладает свойством, что x входит в нее только с отрицанием, то по доказательству $A(x, y) \wedge B(y, z)$ в СР размера s можно построить вещественную монотонную схему C размера не более s, что $C(x) = 1$ для всех $x \in U$ и $C(x) = 0$ для всех $x \in V$, где $U = \{x \mid \exists y A(x, y) = 1\}$, а $V = \{x \mid \exists z B(x, z) = 1\}$.

Доказательство. Рассмотрим только случай, когда x входит в $A(x, y)$ только без знаков отрицаний. Второй случай рассматривается аналогично. В этом случае во всех неравенствах, которые задают $A(x, y)$ в виде $f(x) + g(y) \leq \alpha$ все коэффициенты f отрицательны. Рассмотрим преобразование, описанное в доказательстве леммы 4.1. Мы будем вычислять только коэффициент β у неравенств $g(y) \leq \beta$. У неравенств, которые происходят из неравенства

$f(x)+g(y) \le \alpha$, представляющего дизъюнкт $A(x,y)$, этот коэффициент равен $\alpha - f(a)$, т.е. монотонно вычисляется по входу, поскольку все коэффициенты f отрицательны. У неравенств, которые происходят из $B(x,z)$, этот коэффициент равен нулю.

Если мы выводим новое неравенство по правилу линейной комбинации с неотрицательными коэффициентами, то это же верно и для коэффициента β, т.е. мы применяем монотонную функцию.

Если мы применяем деление на d с округлением, то к β применяется функция $x \to \lceil \frac{x}{d} \rceil$, которая является монотонной.

Наконец, в итоге, нам нужно применить функцию, которая при $\beta \ge 0$ выдаст 1 (т.е. не получилось вывести $A(x,z)$, следовательно $B(x,z)$ является невыполнимой), а при $\beta < 0$ выдаст 0.

При $x \in U$ формула $A(x,y)$ выполнима, следовательно схема выдаст 1, а если $x \in V$, то формула $B(x,z)$ выполнима, тогда схема выдаст 0. □

4.2 Клики и раскрашиваемые графы

Первая суперполиномиальная нижняя оценка на размер монотонной булевой схемы, проверяющей, есть ли в графе полный подграф (клика) на k вершинах была получена А.А. Разборовым [Raz85].

Более сильный результат тем же методом был получен Алоном и Боппаной, а затем усилен Пудлаком.

Теорема 4.2 ([AB87], [Pud97])**.** Размер любой (1) булевой [] (2) вещественной [] монотонной схемы, которая принимает $\frac{n(n-1)}{2}$ входов (для каждого возможного ребра графа на n вершинах написано, присутствует ли оно в графе) и выдает 1, если граф содержит клику на k вершинах и выдает 0, если вершины графа можно правильным образом раскрасить в $k-1$ цвет, не меньше $n^{\Omega(\sqrt{k})}$ при $k \le \sqrt[4]{n}$.

Запишем формулой факт, что граф одновременно имеет клику размера k и правильным образом красится в $k-1$ цвет.

Пусть переменные $z_{i,j}$ для $i \ne j \in [n]$ кодируют наличие ребра между вершинами i и j, переменная $r_{i,l}$ для $i \in [n], l \in [k]$ означает, что вершина i является l-ой вершиной клики, а переменная $q_{i,m}$ для $i \in [n], m \in [k-1]$ означает, что вершина i покрашена в цвет m. Условие $Clique_{n,k}(r,z)$ наличия клики кодируется с помощью следующего множества дизъюнктов:

- $\bigvee_{i=1}^{n} r_{i,l}$ для всех $l \in [k]$: хотя бы одна вершина является l-ой вершиной клики.
- $\neg r_{i,l} \vee \neg r_{i,j}$ для всех $l \ne j \in [k]$ и всех $i \in [n]$: одна вершина не может быть двумя вершинами клики.
- $\neg r_{i,l} \vee \neg r_{j,m} \vee z_{i,j}$ для всех $i \ne j \in [n]$ и всех $l \ne m \in [k]$: если две вершине в клике, то между ними есть ребро.

Заметим, что в $Clique_{n,k}(r,z)$ переменные $z_{i,j}$ входят без отрицания.

Условие $Coloring_{n,k}(q,z)$ раскрашиваемости графа в $k-1$ цвет кодируется так:

- $\bigvee_{j=1}^{n-1} q_{i,j}$ для всех $j \in [k-1]$: каждая вершина покрашена в какой-то цвет.

- $\neg q_{i,l} \vee \neg q_{j,l} \vee \neg z_{i,j}$ для всех $i \neq j \in [n]$ и всех $l \in [k-1]$: если две вершины соединены ребром, то они окрашены в разный цвет

Заметим, что в $Coloring_{n,k}(q,z)$ переменные $z_{i,j}$ входят только с отрицанием.

Таким образом, из теоремы 4.2 и теоремы 4.1 следует

Теорема 4.3. При $k \leq \sqrt[4]{n}$ размер CP доказательства формулы $Clique_{n,k}(r,z) \wedge Coloring_{n,k}(q,z)$ есть $n^{\Omega(\sqrt{k})}$.

5 Исчисление полиномов

Пусть $\mathbb{F}$ — поле. Исчисление полиномов — это система доказательств, с помощью которой можно показывать, что множество полиномов из $\mathbb{F}[x_1, x_2, \ldots, x_n]$ не имеет общих корней на булевом кубе $\{0,1\}^n$.

Правила вывода:

1. Для каждой переменной x_i можно вывести многочлен $\frac{}{x_i^2 - x_i}$, который означает, что переменная из множества $\{0,1\}$.
2. Правило линейной комбинации: $\frac{p_1, p_2}{\alpha p_1 + \beta p_2}$ для любых $\alpha, \beta \in \mathbb{F}$.
3. Умножение на переменную: $\frac{p}{px_i}$ для любой переменной x_i.

Говорят, что многочлен q выводится из множества многочленов $\{p_1, \ldots, p_k\}$, если он может быть получен из них применением некоторого количества правил. При этом степень вывода — это степень максимального многочлена, который использовался в выводе. Обозначение: $p_1, \ldots, p_k \vdash_d q$. Доказательством отсутствия общих нулей многочленов на булевом кубе называется вывод 1.

Мультилинеаризацией многочлена называется замена всех вхождений x_i^k при $k > 1$ на x_i. Пусть для многочлена p его мультилинеаризацией называется многочлен $m(p)$.

Предложение 5.9. Из p можно вывести $m(p)$ со степенью вывода $\deg p$ и из $m(p)$ можно вывести p со степенью вывода $\deg p$.

Доказательство. Для каждой переменной последовательно будем понижать степень по x_i, заменять $x_i^k t$ на $x_i^{k-1} t$. Для этого достаточно вывести $x_i^k t - x_i^{k-1} t = (x_i^2 - x_i) x_i^{k-2} t$, а этот многочлен выводится из $x_i^2 - x_i$ домножением на переменные.

Обратный вывод получается точно также. □

Теорема 5.1. Каждая функция $\{0,1\}^n \to \mathbb{F}$ имеет единственное представление в виде мультилинейного многочлена.

Доказательство. Для существования представления достаточно показать, что функция, которая в точке $a \in \{0,1\}^n$ равна 1, а в остальных равна нулю, представима. Это делается просто: $\chi_a = (x_1 + a_1 - 1)(x_2 + a_2 - 1) \ldots (x_n + a_n - 1) \frac{1}{(2a_1 - 1)(2a_2 - 1) \ldots (2a_n - 1)}$.

Теперь $f(x) = \sum_{a \in \{0,1\}^n} f(a) \chi_a$.

Для доказательства единственности заметим, что множество всех функций из $\{0,1\}^n \to \mathbb{F}$ является линейным пространством размерности 2^n. Всего различных мультилинейных мономов ровно 2^n. По доказанному любая функция является линейной комбинацией этих мономов,

следовательно, эти мономы линейно независимы и такое представление является единственным. □

Лемма 5.1. Пусть многочлены $p_1, p_2, \ldots, p_k, q \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ имеют степень не более d и обладают таким свойством, что любой общий ноль $p_1, \ldots, p_k$ на булевом кубе является также нулем q. Тогда q выводится из $p_1, p_2, \ldots, p_k$ в исчислении полиномов, причем степень вывода не превосходит $\max\{n+1, d\}$.

Доказательство. По предложению 5.9 достаточно доказать лемму в предположении, что все многочлены мультилинейные. Будем доказывать по индукции по числу переменных. База индукции $n = 1$.

Индукционный переход. По теореме 5.1 для каждого мультилинейного многочлена f выполняется $f = f|_{x_i=0}(1 - x_i) + f|_{x_i=1}x_i$. Это верно, так как многочлены слева и справа оба являются мультилинейными и совпадают на булевом кубе, следовательно их коэффициенты совпадают по теореме 5.1.

Покажем, что из любого многочлена f можно вывести $f|_{x_i=0}(1 - x_i)$ и $f|_{x_i=1}x_i$. Действительно, домножим f на $(1-x_i)$ получим многочлен $f|_{x_i=0}(1-x_i)^2+f|_{x_i=1}x_i(1-x_i)$, из которого надо вычесть $f|_{x_i=1}x_i(1-x_i) = (x_i^2-x_i)(-f|_{x_i=1}+f|_{x_i=0})$ и получим $f|_{x_i=0}(1-x_i)$. Аналогично выведем $f|_{x_i=1}x_i$.

По индукционному предположению из $p_1|_{x_n=0}, \ldots, p_k|_{x_n=0}$ можно вывести $q|_{x_n=0}$, следовательно из $p_1|_{x_n=0}(1 - x_n), \ldots, p_k|_{x_n=0}(1 - x_n)$ выводится $q|_{x_n=0}(1 - x_n)$, аналогично из $p_1|_{x_n=1}x_n, \ldots, p_k|_{x_n=1}x_n$ выводится $q|_{x_n=1}x_n$. Поскольку все $p_i|_{x_n=0}(1 - x_n)$ и $p_i|_{x_n=1}x_n$ выводятся из p_i, то получаем, что есть вывод $q|_{x_n=0}(1 - x_n) + q|_{x_n=1}x_n$. □

5.1 Исчисление полиномов как доказательство невыполнимости

Исчисление полиномов можно использовать и как доказательство невыполнимости формулы в КНФ. Каждый дизъюнкт формулы $\bigvee_{i\in P} x_i \vee \bigvee_{i\in N} \neg x_j$ представляется в виде полинома $\prod_{i\in P}(1 - x_i)\prod_{i\in N} x_i$. Формула выполнима тогда и только тогда, когда получающиеся множество полиномов имеет общий ноль.

Обобщим обозначение $\vdash_d$ на формулы и будем говорить, что $\phi \vdash_d 1$, если есть вывод степени не более d из полиномов, которые соответствуют ϕ.

Предложение 5.10. Если для невыполнимой формулы ϕ выполняется, что $w_R(\phi) = w$, то $\phi \vdash_{w+1} 1$.

Доказательство. Для каждого дизъюнкта резолюционного доказательства выведем соответствующий ему многочлен. Многочлен для пустого дизъюнкта — это 1. Многочлены для исходных дизъюнктов уже есть. Для правила резолюции, которое применяется к дизъюнктам $A \vee x$ и $B \vee \neg x$, мы сначала выведем полиномы, которые соответствуют $A \vee B \vee x$ и $A \vee B \vee \neg x$, они получаются из исходных просто умножением, а затем сложим получившиеся многочлены. □

Если смотреть не только на степень вывода, но и на размер (число мономов в выводе), то такое моделирование не работает, поскольку число мономов при кодировании дизъюнкта может быть большим. Для этого рассматривают еще одну систему доказательств: Polynomial calculus + Resolution (PCR). В этой системе доказательств есть переменные не только для переменной x, но и для отрицания переменной $\overline{x}$. Кроме того можно вывести многочлен $1-x-\overline{x}$.

Стандартным представлением дизъюнкта в PCR является такое, что многочлен становится мономом.

Предложение 5.11. PCR моделирует резолюцию.

Следующее утверждение показывает, что PCR над полем $\mathbb{F}_2$ строго сильнее резолюции:

Предложение 5.12. Пусть ϕ невыполнимая формула, которая кодирует в КНФ невыполнимую систему линейных уравнений над $\mathbb{F}_2$. Тогда ϕ имеет короткое доказательство в PCR над полем $\mathbb{F}_2$.

Предложение 5.13. Степень доказательства в PC и PCR совпадают.

5.2 Поиск доказательства

Степень доказательства в исчислении полиномов играет ту же роль, что и ширина в резолюционном доказательстве.

Далее мы будем считать, что вывод оперирует исключительно мультилинейными многочленами, т.е. мы сразу выводим мультилениаризацию любого полученного многочлена и дальше работаем только с ней.

Известно, что если существует вывод степени d, то его можно найти за время $n^{O(d)}$. Для этого можно воспользоваться алгоритмом поиска базиса Гребнера.

5.3 Связь между размером и степенью доказательства

Между размером PC (PCR) доказательства и степени такая же связь, как между шириной и размером.

Следующая лемма похожа на лемму 3.1.

Лемма 5.2. Если невыполнимой формулы ϕ в КНФ и натурального числа d выполняется, что $\phi|_{x=0} \vdash_{d-1} 1$ и $\phi|_{x=1} \vdash_d 1$, то $\phi \vdash_d 1$.

Доказательство. Упражнение. □

Теорема 5.2. Пусть ϕ — невыполнимая формула в k-КНФ. Тогда число мономов (с учетом повторений) в любом PCR (и PC) доказательстве формулы ϕ не меньше, чем $2^{\Omega}((\deg \phi - k)^2/n)$, где $\deg \phi$ — это минимальная степень доказательства ϕ, а n — число переменных формулы ϕ.

Доказательство. Доказательство практически дословно повторяет доказательство теоремы 3.2 с заменой ширины на степень и дизъюнктов на мономы. □

Таким образом, чтобы доказать нижнюю оценку на размер доказательства формулы в $O(1)$-КНФ в PC и PCR достаточно доказать нижнюю оценку на степень доказательства.

5.4 Нижняя оценка на степень опровержения случайных формул

Мы рассмотрим распределение $\mathcal{F}^d_{n,m}$ на формулах в d-КНФ от n переменных, состоящих из m дизъюнктов. Формула из распределения $\mathcal{F}^d_{n,m}$ строится так: m раз независимо выбирается дизъюнкт размера d из всех дизъюнктов размера d от переменных $x_1, x_2, \ldots, x_n$ равновероятно.

Факт 5.4. Для любого натурального d существует $c > 0$, что с вероятностью $1-o(1)$ случайная формула из $\mathcal{F}^d_{n,cn}$ будет невыполнимой.

Пусть формула ϕ от переменных $x_1, x_2, \ldots, x_n$ имеет m дизъюнктов. Построим по этой формуле двудольный граф. В левой доле A будет m вершин, они соовтетствуют дизъюнктам ϕ, в правой доле будет n вершин, они соответствуют переменным. Ребром соединяются дизъюнкты с переменными, которые в этот дизъюнкт входят.

Факт 5.4. Для любой константы a и любого $\epsilon > 0$ и для любой достаточно большой константы $d \geq 3$ граф, построенный по формуле из $\mathcal{F}^d_{n,an}$ с вероятностью $1-o(1)$ является (r, c)-граничным экспандером, где $r = \Omega(n)$, $c = (1 - \epsilon)d$.

Следующую теорему мы оставим без доказательств:

Теорема 5.3. [AR01] Пусть граф невыполнимой формулы ϕ является (r, c)-граничным экспандером. Тогда $\deg \phi \geq rc/2$.

6 Системы Фреге

6.1 Эквивалентность систем Фреге

С помощью систем Фреге можно доказывать, что пропозициональная формула является тавтологией. Все системы Фреге имеют некоторый конечный набор правил вывода вида $\frac{\phi_1,\phi_2,\ldots,\phi_k}{\psi}$, где ϕ_i, ψ — пропозициональные формулы такие, что формула $(\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_k) \to \psi$ являются тавтологией. У правил может быть пустое множество посылок, такие правила называют аксиомами. Выводом формулы ϕ из списка пропозициональных формул L в системе Фреге является называется конечная последовательность формул, в которой каждая формула либо принадлежит списку L, либо получается из предшествующих формул по правилу вывода, в котором вместо переменных можно подставлять произвольные пропозициональные формулы. Формула τ называется выводимой в системе Фреге, если она выводима из пустого списка. Система Фреге должна обладать свойством импликационной полноты: если из формул $\phi_1, \phi_2, \ldots, \phi_m$ семантически следует формула ψ, то ψ должна выводится из списка $\{\phi_1, \phi_2, \ldots, \phi_m\}$. Рассмотрим секвенциальное исчисления высказываний (LK), которое эквивалентно по силе системам Фреге, хотя формально и не является системой Фреге. Секвенцией называется строка $\Gamma \vdash \Delta$, где Γ, Δ — это списки формул. Каждой секвенции соответствует пропозициональная формула: $\bigwedge_{\phi\in\Gamma} \phi \to \bigvee_{\phi\in\Delta} \phi$. Секвенция называется тавтологией, если тавтологией является соответствующая формула. Аксиомами называются секвенции, у которых Γ и Δ содержат общую формулу. Правила вывода соответствуют каждой связки и тому, слева или справа от $\vdash$ они располагаются.

$(\neg \vdash)$ $\frac{\Gamma\vdash\Delta,A}{\Gamma,\neg A\vdash\Delta}$

$(\vdash \neg)$ $\frac{A,\Gamma\vdash\Delta}{\Gamma\vdash\neg A,\Delta}$

$(\vee \vdash)$ $\frac{\Gamma,A\vdash\Delta;\quad \Gamma,B\vdash\Delta}{\Gamma,A\vee B\vdash\Delta}$

$(\vdash \vee)$ $\frac{\Gamma\vdash A,B,\Delta}{\Gamma\vdash A\vee B,\Delta}$

$(\wedge \vdash)$ $\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}$

$(\vdash \wedge)$ $\frac{\Gamma \vdash A, \Delta; \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}$

Кроме этих правил есть еще одно *правило сечения*:

$$\frac{\Gamma, A \vdash \Delta; \Gamma \vdash \Delta, A}{\Gamma \vdash \Delta}$$

Секвенция называется выводимой тогда и только тогда, когда ее можно вывести из аксиом с помощью перечисленных правил вывода. Легко проверить, что если секвенция выводима, то ей соответствует тавтология. Для этого достаточно проверить, что тавтология соответствует аксиомам и проверить про каждое правило вывода, что если выполняются формулы, соответствующие секвенциям-посылкам, то выполняются и формулы, соответствующие секвенциям-заключениям парвила. Полноту исчисления секвенций можно показать даже в случае, если правило сечения не используются. Пусть есть секвенция, которая является тавтологией. Будем применять правила снизу в верх и строить дерево. Изначально дерево состоит из одной вершины, той секвенции, которую мы хотим вывести. Каждый раз мы берем какой-нибудь лист дерева, выбираем в секвенции, которая соответствует этому листу, связку и применяем правило, которое соответствует этой связке. Заметим, что в этом дереве при переходе от родителя к ребенку число связок уменьшается. Значит, дерево рано или поздно оборвется, и в листьях этого дерева будут стоять секвенции без логических связок. Если все секвенции без логических связок являются аксиомами, то это значит, что мы построили вывод. Пусть в каком-то листе стоит секвенция, которая не является аксиомой, т.е. переменные слева и справа от $\vdash$ различные. Рассмотрим такой набор значений переменных: всем переменным слева от $\vdash$ мы подставим 1, а всем переменным справа от $\vdash$ подставим 0. Легко проверить, что все правила вывода обладают таким свойством: если значение одной из посылок при данном значении переменных ложно, то и заключение правила ложно при том же наборе значений переменных. Таким образом и исходная секвенция ложна при данном наборе значений переменных, следовательно эта секвенция не является тавтологией. Выводм формулы ϕ в исчислении секвенций называется вывод секвенции $\vdash \phi$.

Лемма 6.1. Система Фреге моделирует исчисление секвенций. Древовидная система Фреге моделирует древовидное секвенциальное исчисление.

Доказательство. Достаточно выводить в сисстеме Фреге все формулы, которые соответствуют секвенциям. Аксиомам соответствуют пропозициональные тавтологии, которые являются частным случаем пропозициональных тавтологий от константного числа переменных. Эти тавтологии имеют вывод константного размера в системе Фреге, поскольку системы Фреге импликационно полные. Правила вывода обладают свойством семантического следствия. Это значит, что правила вывода можно представить как семантическое правило вывода формул, в которых используется лишь константное число переменных. Значит, заключение имеет вывод из посылок константного размера, а потом к этому выводу нужно применить подстановку формул вместо переменных. Таким образом, в системе Фреге можно повторять секвенционный вывод с константным замедлением. Если исходное доказательство было древовидным, то и полученное доказательство тоже можно сделать древовидным. □

Наша ближайшая цель показать, что системы Фреге, секвенциальная система и их древовидные аналоги эквивалентны. Для этого мы рассмотрим еще одну систему доказательств —

игры Пудлака и Баса [PB94]. Есть два игрока Павел и Сэм, у них есть тавтология ϕ. Сэм говорит, что знает набор значений переменных, при котором ϕ ложно. Павел пытается уличить Сэма и задает ему вопросы про значение произвольных формул от переменных формулы ϕ. Сэм отвечает. Павел уличает Сэма, если он получил *непосредственное противоречие*, это значит, например, он спрашивал ответы для формул $\phi \vee \psi$, ϕ, ψ, но ответы не сошлись. Аналогично и для других связок. Деревом игры называется такое двоичное дерево, каждая внутренняя вершина которого помечена формулой, одно из ребер к потомку помечено 0, другое 1. В каждом листе должно быть непосредственное противоречие (мы всегда считаем, что есть ответ 0 для исходной формулы ϕ). Длину максимального пути от корня до листа будем называть высотой дерева. Размером дерева мы называем суммарный размер формул, которыми помечены его вершины.

Замечание 6.1. Мы рассматриваем формулы, в которых используются только бинарные операции $\vee$ и $\wedge$ и унарная операция $\neg$. Однако часто формулы естественно переписываются в форме, когда $\vee$ или $\wedge$ не бинарные. В случае не бинарных операций понятие непосредственного противоречия можно было бы расширить. Например, если $\bigvee_{i=1}^{n} \phi_i = 0$, а $\phi_k = 1$, то это уже противоречие. Однако в реальности, чтобы получить это противоречие, нужно сделать несколько запросов. А именно рассмотрим дерево для формулы $\bigvee_{i=1}^{n} \phi_i = 0$, в этом дереве рассмотрим путь от корня до листа ϕ_k, в этом пути есть противоречие в начале и конце, т.е. можно найти настоящее непосредственное противоречие с помощью бинарного поиска, на это дополнительно потратится $O(\log n)$ запросов.

Лемма 6.2. По доказательству формулы ϕ в системе Фреге размера s можно построить дерево игры Пудлака-Баса высоты $O(\log s)$ и размера $poly(s)$, где константа зависит только от правил системы Фреге.

Доказательство. Пусть $\psi_1, \psi_2, \ldots, \psi_s = \phi$ — это вывод формулы ϕ в системе Фреге. Опишем дерево игры в виде стратегии для Павла. Сначала Павел с помощью бинарного поиска находит такое k, что на формулу $\psi_1 \wedge \psi_2 \wedge \ldots \psi_k$ Сэм отвечает 1, а на $\psi_1 \wedge \psi_2 \wedge \ldots \psi_k \wedge \psi_{k+1}$ отвечает 0. Такое должно быть, поскольку на пустой формуле мы считаем, что ответ 1, а на формуле $\psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_s$ ответ должен быть нулем, иначе находится непосредственное противоречие с $\phi = 0$. Когда найдено такое k, то Павел выяснил, что значение ψ_{k+1} есть ноль, тогда как значение всех предыдущих должно быть 1 и Павел будет получать противоречие с применением правила в системе Фреге. Если выяснить про все посылки правила, что их значение 1, а у заключение значения 0, то непосредственное противоречие можно получить, запросив все переменные правила (это формулы), которых конечное количество. □

Лемма 6.3. По дереву игры Пудлака-Баса для формулы ϕ высоты h и размера s можно построить древовидный вывод секвенции $\vdash \phi$ высоты $h + O(1)$ и размера $poly(s)$.

Доказательство. Каждая вершина дерева игры однозначно определяется запросами и ответами на них $(\psi_1, b_1), (\psi_2, b_2), \ldots, (\psi_k, b_k)$. Сопоставим секвенцию вида $\{\psi_i : b_i = 1\} \vdash \{\psi_i : b_i = 0\}$. Секвенция в корне дерева будет в точности $\vdash \phi$, секвенция в вершине получается с помощью правила сечения из секвенций в детях. Осталось показать, что секвенции в листьях выводимы с помощью вывода константного размера. Это проверяется разбором случаев в зависимости от того, какое непосредственное противоречие было получено. Например, мы знаем, что $a \vee b$ ложно, при этом a истинно. Тогда достаточно проверить, что секвенция $a \vdash a \vee b$ имеет вывод

константного размера, а это правда, поскольку эта секвенция является тавтологией и использует конечное количество переменных. □

Теорема 6.1. Системы Фреге, секвенциальное исчисление и древовидные их варианты полиномиально эквивалентны.

Доказательство. Следует из лемм 6.2, 6.3, 6.1. □

Определим поглощенную форму формулы — мы объединяем все $\vee$ от $\vee$-ов в один большой $\vee$ и аналогично поступаем с $\wedge$. Глубиной формулы будем называть глубину ее поглощенной формы. Глубиной доказательства во Фреге является глубина максимальной формулы из доказательства. Глубина дерева игры — это глубина максимальной формулы в дереве.

Упражнение 6.1. Минимальная глубина доказательства Фреге и дерева игры отличаются на константу.

Легко проверить, что если формула ϕ имеет резолюционное доказательство размера s, то $\neg\phi$ имеет дерево игры высоты $O(\log s)$ и глубины 3. Для этого достаточно делать запросы на конъюнкции первых k дизъюнктов вывода и найти первый дизъюнкт, который ложный, а все перед ним истинны.

6.2 Нижние оценки для систем Фреге ограниченной глубины

В системах Фреге неизвестно суперполиномиальных нижних оценок на сложность вывода. В этом параграфе мы ставим менее амбициозную задачу, а именно мы докажем нижнюю оценку в системах Фреге, глубина которых ограничена константой. Сложной формулой будет принцип Дирихле $\mathrm{PHP}_{\mathrm{n}}^{\mathrm{n}+1}$.

А именно верна такая теорема:

Теорема 6.2 ([BH10, UF96])**.** Пусть $\mathcal{F}$ — система Фреге. Тогда для достаточно больших n для любой константы d доказательство $\neg PHP_n^{n+1}$ в $\mathcal{F}$ глубины d имеет размер как минимум 2^{n^μ} для любого $\mu < \frac{1}{2}\left(\frac{1}{5}\right)^{d+c}$, где c — это константа которая зависит только от системы Фреге.

Список литературы

[AB87] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[AD08] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008.

[AR01] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 190–199, 2001.

[BGL10] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. A lower bound for the pigeonhole principle in tree-like resolution by asymmetric prover-delayer games. *Inf. Process. Lett.*, 110(23):1074–1077, 2010.

[BH10] Eli Ben-Sasson and Prahladh Harsha. Lower bounds for bounded depth frege proofs via pudlák-buss games. *ACM Trans. Comput. Log.*, 11(3):19:1–19:17, 2010.

[BSW01] E. Ben-Sasson and A. Wigderson. Short proofs are narrow — resolution made simple. *Journal of ACM*, 48(2):149–169, 2001.

[CR74] Stephen A. Cook and Robert A. Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *Proceedings of the 6th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1974, Seattle, Washington, USA*, pages 135–148, 1974.

[DR01] Stefan S. Dantchev and Søren Riis. Tree Resolution Proofs of the Weak Pigeon-Hole Principle. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity, Chicago, Illinois, USA, June 18-21, 2001*, pages 69–75. IEEE Computer Society, 2001.

[PB94] Pavel Pudlák and Samuel R. Buss. How to lie without being (easily) convicted and the length of proofs in propositional calculus. In *Computer Science Logic, 8th International Workshop, CSL '94, Kazimierz, Poland, September 25-30, 1994, Selected Papers*, pages 151–162, 1994.

[PI00] Pavel Pudlák and Russell Impagliazzo. A lower bound for DLL algorithms for k-SAT (preliminary version). In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, 2000, San Francisco, CA, USA.*, pages 128–136, 2000.

[Pud97] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997.

[Raz85] A. A. Razborov. Lower bounds for the monotone complexity of some boolean functions. *Soviet Mathematics Doklady*, 31:354Ц–357, 1985.

[UF96] Alasdair Urquhart and Xudong Fu. Simplified lower bounds for propositional proofs. *Notre Dame Journal of Formal Logic*, 37(4):523–544, 1996.