

## **ПРЕПРИНТЫ ПОМИ РАН**

**ГЛАВНЫЙ РЕДАКТОР**

**С.В. Кисляков**

### **РЕДКОЛЛЕГИЯ**

В.М.Бабич, Н.А.Вавилов, А.М.Вершик, М.А.Всемирнов, А.И.Генералов, И.А.Ибрагимов,  
Л.Ю.Колотилина, Б.Б.Лурье, Ю.В.Матиясевич, Н.Ю.Нецеветаев, С.И.Репин, Г.А.Серегин

Учредитель: Федеральное государственное бюджетное учреждение науки  
Санкт-Петербургское отделение Математического института  
им. В. А. Стеклова Российской академии наук

Свидетельство о регистрации средства массовой информации: ЭЛ №ФС 77-33560 от 16  
октября 2008 г. Выдано Федеральной службой по надзору в сфере связи и массовых  
коммуникаций

Контактные данные: 191023, г. Санкт-Петербург, наб. реки Фонтанки, дом 27

телефоны:(812)312-40-58; (812) 571-57-54

e-mail: [admin@pdmi.ras.ru](mailto:admin@pdmi.ras.ru)

[http://www.pdmi.ras.ru /preprint/](http://www.pdmi.ras.ru/preprint/)

Заведующая информационно-издательским сектором Симонова В.Н

# Нижние оценки на размер резолюционных доказательств некоторых слабых формул\*

Д.М. Ицыксон<sup>†1</sup>, В.В. Опарин<sup>‡2</sup>, М.Г. Слабодкин<sup>§2</sup>, and Д.О. Соколов<sup>¶1</sup>

<sup>1</sup>Санкт-Петербургское отделение Математического института им. В. А. Стеклова Российской академии наук

<sup>2</sup>Санкт-Петербургский Академический университет Российской академии наук

12 ноября 2013

## Аннотация

Мы доказываем, что для каждого  $d$  и для достаточно больших  $n$  для каждой последовательность натуральных чисел  $d_1, d_2, \dots, d_n$ , не превосходящих  $d$  можно построить граф на  $n$  вершинах, степень которого ограничена константной (но степень  $i$ -й вершины не меньше  $d_i$ ), что из графа нельзя удалить несколько ребер так, чтобы в получившемся графе степень  $i$ -й вершины равнялась  $d_i$  и доказательство этого утверждения в резолюциях имеет размер  $2^{\Omega(n)}$ . Из этого результата следуют известные результаты об экспоненциальных нижних оценках для принципа Дирихле и цейtinских формул.

Мы доказываем, что древовидная резолюционная сложность формулы, которая кодирует невозможность разбиения квадрата  $n \times n$  на доминошки при нечетных  $n$ , равняется  $2^{\Theta(n)}$ .

Мы доказываем, что древовидная резолюционная сложность CSP задачи, которая кодирует утверждение о том, что невозможно в каждую клетку квадрата  $n \times n$  расставить по одной стрелке из множества  $\{\leftarrow, \nwarrow, \uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow\}$ , так чтобы стрелки в двух соседних клетках отличались поворотом не более, чем на  $45^\circ$  градусов, а стрелки на границе квадрата не были бы направлены вне квадрата, равняется  $2^{\Theta(n)}$ .

---

\*Поддержано грантом президента МК-4108.2012.1, грантами РФФИ 12-01-31239-мол\_a, 11-01-00760-а и программой фундаментальный исследований ОМН РАН.

<sup>†</sup>dmitrits@pdmi.ras.ru

<sup>‡</sup>oparin.vsevolod@gmail.com

<sup>§</sup>slabodkinm@gmail.com

<sup>¶</sup>sokolov.dmt@gmail.com

ПРЕПРИНТЫ  
Санкт-Петербургского отделения  
Математического института им. В. А. Стеклова  
Российской академии наук

PREPRINTS  
of the St. Petersburg Department of Steklov Institute of Mathematics

---

ГЛАВНЫЙ РЕДАКТОР  
С. В. Кисляков

РЕДКОЛЛЕГИЯ

В. М. Бабич, Н. А. Вавилов, А. М. Вершик, М. А. Всемирнов, А. И. Генералов,  
И. А. Ибрагимов, Л. Ю. Колотилина, Г. В. Кузьмина, П. П. Кулиш, Б. Б. Лурье,  
Ю. В. Матиясевич, Н. Ю. Нецеветаев, С. И. Репин, Г. А. Серегин, В. Н. Судаков,  
О. М. Фоменко

# 1 Введение

Резолюционная система доказательств является одной из наиболее простых и изученных систем доказательства. Известны методы, которые позволяют доказывать нижние и верхние оценки на сложность некоторых типов формул. Однако неизвестно универсального метода, который бы позволял по семейству формул определить порядок роста сложности резолюционного вывода для этого семейства.

Мы говорим, что семейство противоречивых формул в КНФ  $F_n$  слабее семейства противоречивых формул в КНФ  $H_n$ , если каждый дизъюнкт формулы  $H_n$  следует из константного числа дизъюнктов формулы  $F_n$ . Поскольку резолюционная система доказательств полна относительно импликаций, то размер минимального доказательства формулы  $H_n$  не меньше, чем размер минимального доказательства  $F_n$ . Таким образом интересно доказывать нижние оценки для как можно более слабых семейств формул.

Бэйкер [Bak95] обобщил резолюционную систему доказательств на задачи выполнения ограничений (CSP) над произвольными алфавитами. Резолюционная система доказательств связана с алгоритмами расщепления (DPLL алгоритмами), каждый алгоритм расщепления на невыполнимой CSP задаче над алфавитом из  $k$  элементов строит  $k$ -ичное дерево, каждой вершине дерева соответствует переменная, а ребра, ведущие к сыновьям, соответствуют  $k$  различным подстановкам этой переменной. В листьях дерева записаны ограничения, которые нарушаются, если сделать подстановку, которая соответствует пути от корня к этому листу. Мы называем такое дерево — деревом поиска противоречия. Известно, что размер минимального дерева поиска противоречия для невыполнимой CSP задачи совпадает с размером минимального древовидного резолюционного доказательства.

Каждой невыполнимой CSP задаче над алфавитом из  $k$  элементов соответствует три параметра:  $S(F)$  — размер минимального резолюционного доказательства,  $S_T(F)$  — размер минимального резолюционного древовидного доказательства и  $d(F)$  — минимальная глубина дерева поиска противоречия. Эти параметры связаны тривиальным неравенством:  $S(F) \leq S_T(F) \leq k^{d(F)}$ . В работе [BEGJ00] построено семейство формул, которое экспоненциально разделяет  $S$  и  $S_T$ , а в работе [Urq11] строится семейство формул  $F_n$ , для которой  $S_T(F_n) = O(n)$ , а  $d(F_n) = \Omega(n/\log n)$ . Число  $d(F)$  является нижней оценкой в наихудшем случае на число запросов, которое нужно сделать к переменным, чтобы найти ограничение, которое не выполняется.

**Классы PPA и PPAD.** Класс PPA (polynomial parity argument) был введен Пападимитриу в [Pap94] и состоит из NP-задач поиска, которые гарантированно имеют решения благодаря принципу четности, который гласит, что в любом графе четное число вершин нечетной степени. Рассмотрим полиномиальную по времени Тьюринга  $M$ , которая для всех  $n$  и для всех  $x \in \{0, 1\}^n$  задает граф из  $2^n$  вершин, вершины которого помечены строками из  $\{0, 1\}^n$ , причем степени каждой вершины в этом графе не превосходят двух, а степень вершины  $0^n$  равняется одному. Для каждого  $c \in \{0, 1\}^n$  машина  $M(x, c)$  выдает список соседей вершины  $c$ . Задача  $A$ , связанная с машиной  $M$  состоит в том, чтобы по  $x$  в графе, которая задает  $M(x, \cdot)$  найти вершину степени 1, отличную от  $0^n$ . Класс PPAD состоит из всех таких задач  $A$  и всех

NP-задач поиска, которые к ним сводятся по Левину.

Класс PPAD (polynomial parity argument directed) определяется аналогичным образом, только рассматриваются ориентированные графы. Ориентированные графы обладают следующими свойствами: в каждую вершину входит и выходит не более одного ребра. Машина  $M(x, c)$  задает множество входящих и исходящих ребер, а вершина  $0^n$  имеет суммарную степень 1. Задача  $A$ , связанная с машиной  $M$  также состоит в том, чтобы по  $x$  в графе, которая задает  $M(x, \cdot)$  найти вершину степени 1, отличную от  $0^n$ . Класс PPAD состоит из всех таких задач  $A$  и всех NP-задач поиска, которые к ним сводятся по Левину. Класс PPAD содержит много интересных задач, в частности, такие задачи являются полными в этом классе: поиск неподвижной точки в дискретной версии теоремы Брауэра, поиск разноцветного треугольника, который существует по двумерной лемме Шпернера [Pap94], поиск равновесия Нэша [CDT09]. Для CSP задачи, кодирующими двумерную лемму Шпернера в работе [CS98] доказана линейная нижняя оценка на глубину дерева поиска противоречия. Это означает, что соответствующую лемму Шпернера PPAD-задачу, нельзя решить быстрее, чем за  $2^{\Omega(n)}$  с помощью алгоритмов, которые используют машину Тьюринга как черный ящик.

**Результаты.** Мы строим семейство двудольных графов  $G_n$  ограниченной константой степени, которые состоят из двух долей по  $n$  и  $n - 1$  вершине соответственно. По графу  $G_n$  строится формула  $\Phi_{G_n}$ , переменные которой соответствуют ребрам графа. Формула  $\Phi_{G_n}$  кодирует существование совершенного паросочетания в графе  $G_n$ , т.е. для каждой вершины есть ограничение, которое говорит, что ровно одно исходящее ребро имеет значение 1. Мы доказываем нижнюю оценку  $2^{\Omega(n)}$  на резолюционную сложность  $\Phi_{G_n}$ . Нижняя оценка на размер доказательства следует из нижней оценки на ширину доказательства [BSW01]. Для оценки ширины доказательства мы используем метод, предложенный Бен-Сассоном и Вигдерсоном в [BSW01], однако мы используем нестандартное понятие семантического следствия и нестандартную меру на множестве дизъюнктов.

Формулы  $\Phi_{G_n}$  слабее, чем формулы RHP <sub>$n$</sub> , кодирующие принцип Дирихле, поэтому из полученного результата следует нижняя оценка  $2^{\Omega(n)}$  на размер резолюционного доказательства RHP <sub>$n$</sub> , полученная в работе [Hak85].

Используя этот результат, мы доказываем более общее утверждение:

**Теорема 1.1.** Для любого  $d \in \mathbb{N}$  существует  $D \in \mathbb{N}$ , что для любого достаточно большого  $n$  и для любой функции  $h : V \rightarrow \{1, 2, \dots, d\}$ , где  $|V| = n$ , существует граф  $G(V, E)$ , степень каждой вершины этого графа не превосходит  $D$ , что формула  $\Psi_G^{(h)}$ , которая кодирует, что в графе  $G$  есть подграф, в котором степень вершины  $v$  равна  $h(v)$ , является невыполнимой и размер любого резолюционного доказательства этой формулы не меньше, чем  $2^{\Omega(n)}$ .

Из этой теоремы следует оценка  $2^{\Omega(n)}$  на цейтинские формулы, полученная в работе [Urq87].

Частный случай этой теоремы при  $d = 2$  соответствует полной задаче в классе PPA.

Мы также рассматриваем конкретный граф  $H_n$ , вершины которого являются клетками квадрата  $n \times n$ , две вершины соединены ребром, если клетки являются

соседними. Для такого графа вопрос существования совершенного паросочетания эквивалентен разбиению квадрата  $n \times n$  на доминошки из двух клеток. Мы доказываем, что древовидная резолюционная сложность формулы  $\Phi_{H_n}$  при нечетных  $n$  равняется  $2^{\Theta(n)}$ . Верхняя оценка доказывается бинарным поиском, а нижняя оценка доказывается с помощью игры доказывающего и откладывающего<sup>1</sup>.

Мы также рассматриваем задачу выполнения ограничения, которая соответствует следующему утверждению: невозможно в каждую клетку квадрата  $n \times n$  расставить по одной стрелке из множества  $\{\leftarrow, \nwarrow, \uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow\}$ , так чтобы стрелки в двух соседних клетках отличались поворотом не более, чем на  $45^\circ$  градусов, а стрелки на границе квадрата не были бы направлены вне квадрата. Это утверждение следует из двумерной теоремы Брауэра о существовании неподвижной точки. Мы доказываем, что резолюционная сложность этой CSP задачи равняется  $2^{\Theta(n)}$ . Из PPAD-полноты двумерной теоремы Брауэра следует, что задача поиска, которая соответствует задаче о стрелочках, лежит в классе PPAD.

## 2 Предварительные сведения

Мы рассматриваем простые графы без петель и кратных ребер. Двудольным графом называется такой граф, вершины которого можно разбить на два непересекающихся множества  $X$  и  $Y$ , так что все ребра соединяют вершины из множества  $X$  с вершинами из множества  $Y$ . Множества  $X$  и  $Y$  мы называем долями.

Паросочетанием в графе называется произвольное множество ребер, концы которых различны. Мы говорим, что паросочетание покрывает вершину  $v$ , если существует ребро из этого паросочетания с концом в вершине  $v$ .

Для двудольного графа солями  $X$  и  $Y$  и  $A \subseteq X$  обозначим  $\Gamma(A)$  множество всех соседей вершин из множества  $A$ .

**Лемма 2.1** (Холл). Пусть в двудольном графе солями  $X$  и  $Y$  для некоторого подмножества  $A \subseteq X$  для всех  $B \subseteq A$  выполняется неравенство  $\Gamma(B) \geq |B|$ , то в графе существует паросочетание, которое покрывает все вершины множества  $A$ .

Пропозициональной переменной мы называем переменную, которая принимает одно из двух значений 0 или 1. Литералом называется пропозициональная переменная или ее отрицание. Дизъюнктом называется дизъюнкция литералов. Формулой в конъюнктивной нормальной форме (КНФ) называется конъюнкция нескольких дизъюнктов. Формула называется формулой в  $k$ -КНФ, если в каждый ее дизъюнкт входит не более  $k$  литералов. Формула называется выполнимой, если существует такое присваивание значений ее переменным, при которой значение формулы становится равным 1 (такое присваивание мы называем выполняющим набором).

Доказательством невыполнимости формулы  $\varphi$  в резолюционной системе является последовательность дизъюнктов, заканчивающаяся пустым дизъюнктом (пустой дизъюнкт будем обозначать  $\square$ ), каждый дизъюнкт является либо дизъюнктом формулы  $\varphi$ , либо получается из предыдущих по правилу резолюции. Правило резолюции позволяет из пары дизъюнктов  $(l_1 \vee l_2 \vee \dots \vee l_n)$  и  $(l'_1 \vee l'_2 \vee \dots \vee l'_m)$ , где  $l'_m = \neg l_n$  вывести

---

<sup>1</sup>Prover-Delayer games

дизъюнкт  $(l_1 \vee \cdots \vee l_{n-1} \vee l'_1 \vee \cdots \vee l'_{m-1})$ . Размером резолюционного доказательства называется число дизъюнктов, которые в него входят.

Бен-Сассон и Вигдерсон в [BSW01] вводят понятие ширины. Ширина дизъюнкта — это количество литералов в нем. Ширина формулы в КНФ — это ширина самого широкого дизъюнкта. Ширина резолюционного доказательства формулы — это ширина самого широкого дизъюнкта, входящего в него.

**Теорема 2.1** ([BSW01]). Размер резолюционного доказательства формулы  $\varphi$  в  $k$ -КНФ не меньше, чем  $2^{\Omega\left(\frac{(w-k)^2}{n}\right)}$ , где  $w$  — это минимальная ширина резолюционного доказательства  $\varphi$ , а  $n$  — число переменных формулы  $\varphi$ .

**Лемма 2.2.** Если формула  $\phi$  получается из невыполнимой формулы  $\psi$  подстановкой некоторых переменных, то формула  $\phi$  невыполнима и размер любого резолюционного доказательства формулы  $\psi$  не меньше, чем размер минимального доказательства формулы  $\phi$ .

Пусть  $X = \{x_1, x_2, \dots, x_n\}$  конечное множество переменных, которые принимают значения из конечного множества  $W = \{w_1, w_2, \dots, w_k\}$ , а  $S$  — это множество ограничений, каждое ограничение определяет зависит от какого-то подмножества переменных  $X'$  и задает те значения, которые эти переменные могут одновременно принимать. Тройка  $\langle X, W, S \rangle$  называется задачей выполнения ограничений (CSP). Если существует значение переменных, которые выполняют все ограничения CSP задачи  $\phi$ , то такая задача называется выполнимой, в противном случае — невыполнимой.

Запретом<sup>2</sup> называется ограничение вида  $\neg(x_1 = a_1 \wedge \cdots \wedge x_\ell = a_\ell)$ , где  $x_1, \dots, x_\ell \in X, a_1, \dots, a_\ell \in W$ . В случае алфавита  $W = \{0, 1\}$  запрет совпадает с дизъюнктом. Любое ограничение, которое зависит от  $\ell$  переменных можно представить в виде конъюнкции не более, чем  $|W|^\ell$  запретов.

Резолюционная система доказательств может быть обобщена на случай CSP, если все ограничения представлены в виде конъюнкции запретов. Резолюционным доказательством CSP формулы  $\phi$  называется последовательность запретов  $C_1, C_2, \dots, C_t$ , которая заканчивается пустым запретом  $\square$ . Каждый запрет является либо запретом из ограничений формулы  $\phi$ , либо получается по правилу резолюции из  $|W|$  других запретов. Пусть  $\{N_a\}_{a \in W}$  — это такое множество запретов, что  $N_a = \neg(x = a \wedge \alpha_a)$  для всех  $a \in W$ , тогда запрет  $\neg(\bigwedge_{a \in W} \alpha_a)$  является резольвентой запретов  $\{N_a\}_{a \in W}$ . Резолюционное доказательство формулы называется древовидным, если существует такое  $k$ -ичное дерево, листья которого помечены запретами исходной формулы, запрет в каждой вершине является резольвентой запретов сыновей, а корень дерева помечен пустым запретом.

Древом поиска противоречия для невыполнимой CSP формулы  $\phi$  называется корневое  $k$ -ичное дерево, вершины которого помечены переменными, а ребра, исходящие из вершины, помеченной переменной  $x$  к сыновьям помечены подстановками  $x := w$  для каждого  $w \in W$ . В листьях дерева написаны ограничения, которые опровергаются подстановками, которые написаны на пути от корня дерева до этого листа.

---

<sup>2</sup>английский термин nogood

**Утверждение 2.1** ([Bak95][Hwa04]). Размер минимального древовидного резолюционного доказательства невыполнимой CSP формулы  $\phi$  совпадает с размером минимального дерева поиска противоречия для формулы  $\phi$ .

## 3 Выделение подграфа

### 3.1 Существование совершенного паросочетания

По каждому неориентированному графу  $G(V, E)$  построим формулу  $\Phi_G$ , которая устроена следующим образом: для каждого ребра  $e \in E$  в формуле есть переменная  $x_e$ . Формула представляет из себя конъюнкцию условий: для каждой вершины  $v$  записываем, что значение ровно одного ребра, исходящего из  $v$ , помечено единицей. Записывается такое условие как конъюнкция утверждения о том, что хотя бы одно ребро, исходящее из  $v$  имеет значение 1:  $\bigvee_{(v,u) \in E} x_{(v,u)}$ , и утверждений, что для каждой пары ребер  $e_1, e_2$ , инцидентных вершине  $v$ , выполняется  $\neg x_{e_1} \vee \neg x_{e_2}$ .

Нетрудно заметить, что если степени всех вершин графа  $G$  не превосходят  $D$ , то формула  $\Phi_G$  записана в  $D$ -КНФ.

В этом разделе мы докажем следующую теорему:

**Теорема 3.1.** Существует такая константа  $D$ , что для всех достаточно больших  $n$  за полиномиальное от  $n$  время можно построить такой граф  $G$  на  $n$  вершинах, что степени всех вершин графа не превосходят  $D$ , формула  $\Phi_G$  невыполнима, а размер любого ее резолюционного доказательства не меньше  $2^{\Omega(n)}$ .

**Определение 3.1.** Двудольный граф солями  $X$  и  $Y$  называется  $(r, d, c)$ -границным экспандером, если степени всех вершин доли  $X$  не превосходят  $d$  и для любого множества вершин  $A \subseteq X$  если  $|A| \leq r$ , то выполняется  $|\delta(A)| \geq c|A|$ , где  $\delta(A)$  — это множество вершин из  $Y$ , из которых исходит ровно одно ребро в множестве  $A$ .

**Лемма 3.1.** Если в двудольном графе солями  $X$  и  $Y$  существует паросочетание, покрывающее все вершины  $Y$ , и существует паросочетание, которое покрывает все вершины множества  $A \subseteq X$ , то существует паросочетание, которое покрывает все вершины  $Y$  и все вершины  $A$  одновременно.

*Доказательство.* Пусть  $L$  обозначает паросочетание, которое покрывает все вершины множества  $A$ , а  $F$  — паросочетание, которое покрывает все вершины из множества  $Y$ . Докажем, что если  $F$  не покрывает все вершины из  $A$ , то можно построить новое паросочетание  $F'$ , которое покрывает все вершины из  $Y$  и число покрытых вершин из  $A$  в  $F'$  больше, чем в  $F$ . Тогда можно будет построить паросочетание, которое покрывает вершины из  $A$  и  $Y$ .

Рассмотрим вершину  $v_1 \in A$ , которая не покрыта паросочетанием  $F$ , и рассмотрим путь  $v_1, u_1, v_2, u_2, \dots, u_{k-1}, v_k$ , в котором ребра  $(v_i, u_i) \in L$ , а ребра  $(u_i, v_{i+1}) \in F$ , при этом все  $v_1, v_2, \dots, v_{k-1} \in A$ , а  $v_k \notin A$ .

Такой путь существует, поскольку он определяется однозначно. Мы чередуем ребра паросочетания, по которым выходим. Из каждой вершины доли  $X$  выходит не более одного ребра паросочетания  $L$ , а из каждой вершины доли  $Y$  выходит ровно

одно ребро паросочетания  $F$ . Из вершины  $v_1$  не выходит ребер паросочетания  $F$ , поэтому зациклиться этот путь не может. Поскольку из каждой вершины множества  $A$  выходит одно ребро паросочетания  $L$ , то путь рано или поздно придет в вершину вне множества  $A$ .

Пусть паросочетание  $F'$  получается из  $F$  удалением всех ребер  $(v_i, v_{i+1})$  и добавлением ребер  $(u_i, v_i)$  для  $1 \leq i < k$ . Паросочетание  $F'$  покрывает все вершины доли  $Y$  и покрывает на одну вершину больше из множества  $A$ , чем паросочетание  $F$ .  $\square$

**Лемма 3.2.** Пусть  $G$  — двудольный  $(r, d, c)$ -граничный экспандер солями  $X$  и  $Y$ , в котором  $c > 2$ , а в доле  $X$  число вершин больше, чем в доле  $Y$ . Пусть двудольный граф  $H$  получается из экспандера  $G$  добавлением паросочетания, покрывающего все вершины доли  $Y$ . Тогда формула  $\Phi_H$  невыполнима и ширина ее резолюционного опровержения не меньше, чем  $\frac{c-1}{2}r$ .

*Доказательство.* В долях  $X$  и  $Y$  разное количество вершин, следовательно в графе  $H$  нет совершенного паросочетания, следовательно формула  $\Phi_H$  невыполнима.

Граф  $H$  является  $(r, d+1, c-1)$ -граничным экспандером, поскольку добавление паросочетания увеличило степень вершин доли  $X$  не более, чем на 1 и для каждого множества  $A \subseteq X$  размер  $\delta(A)$  мог уменьшится не более, чем на  $|A|$ .

Будем называть набор значений переменных формулы  $\Phi_H$  допустимым, если для всех вершин графа  $H$  не более, чем одно исходящее ребро, имеет значение 1. Пусть  $S$  — это подмножество вершин графа  $H$ , будем говорить, что дизъюнкт  $C$  допустимо следует из множества вершин  $S$ , если любой допустимый набор, который выполняет все условия в вершинах множества  $S$ , выполняет также и дизъюнкт  $C$ . Будем обозначать это так:  $S \vdash C$ .

Введем меру  $\mu$  на множестве дизъюнктов резолюционного доказательства формулы  $\Phi_H$ .  $\mu(C) = \min\{|S \cap X| \mid S \vdash C\}$ .

Мера  $\mu$  обладает следующими свойствами:

1. Мера всех дизъюнктов формулы  $\Phi_H$  равняется нулю или единице.
2. Полуаддитивность:  $\mu(C) \leq \mu(C_1) + \mu(C_2)$ , если дизъюнкт  $C$  получается применением правила резолюции из дизъюнктов  $C_1$  и  $C_2$ .

Пусть  $S_1 \vdash C_1$  и  $|S_1 \cap X| = \mu(C_1)$ , а  $S_2 \vdash C_2$  и  $|S_2 \cap X| = \mu(C_2)$ , тогда  $S_1 \cup S_2 \vdash C_1$  и  $S_1 \cup S_2 \vdash C_2$ , следовательно  $S_1 \cup S_2 \vdash C$ , поэтому  $\mu(C) \leq |S_1 \cap X| + |S_2 \cap X| = \mu(C_1) + \mu(C_2)$ .

3. Мера пустого дизъюнкта  $\square$  больше  $r$ .

От противного, пусть  $\mu(\square) \leq r$ , рассмотрим такое множество вершин  $S$ , что  $S \vdash \square$  и  $|S \cap X| \leq r$ . Для любого  $A \subseteq S \cap X$  выполняется  $|\Gamma(A)| \geq |\delta(A)| \geq (c-1)|A| \geq |A|$ , следовательно по лемме Холла (лемма 2.1) в графе  $H$  существует паросочетание, которое покрывает все вершины множества  $S \cap X$ . По построению в графе  $H$  есть паросочетание, которое покрывает все вершины множества  $Y$ , следовательно по лемме 3.1 существует паросочетание, которое покрывает все вершины множества  $S \cap X$  и все вершины множества  $Y$ , а следовательно покрывает все вершины множества  $S$ . Этому паросочетанию соответствует набор значений переменных, который выполняет все вершины

множества  $S$ , но пустой дизъюнкт выполнить невозможно, поэтому мы получили противоречие с тем, что  $\mu(C) \leq r$ .

Из свойств меры следует, что в доказательстве формулы  $\Phi_H$  найдется дизъюнкт  $C$ , мера которого находится в пределах  $\frac{r}{2} \leq \mu(C) \leq r$ . Пусть  $S \vdash C$  и при этом  $|S \cap X| = \mu(C)$ . Для краткости обозначим  $A = S \cap X$ . Поскольку  $H$  является  $(r, d + 1, c - 1)$ -границным экспандером, то  $\delta(A) \geq (c - 1)|A|$ . Пусть  $F$  — множество ребер, соединяющих  $A$  и  $\delta(A)$ , поскольку в каждую вершину множества  $\delta(A)$  входит ровно одно ребро из множества  $A$ , то  $|F| = |\delta(A)|$ . Рассмотрим одно ребро  $f \in F$ , пусть  $f = (u, v)$ , где  $u \in A$ . Поскольку  $|(S \setminus \{u\}) \cap X| < |S \cap X|$ , то из множества  $S \setminus \{u\}$  допустимо не следует дизъюнкт  $C$ , т.е. существует допустимый набор значений переменных  $\sigma$ , который выполняет условия в вершинах  $S \setminus \{u\}$ , но опровергает дизъюнкт  $C$ . Набор значений  $\sigma$  не выполняет условие в вершине  $u$ , так как иначе  $\sigma$  выполнял бы  $S$ , а следовательно и  $C$ . Поскольку набор  $\sigma$  является допустимым, то он назначает всем ребрам, исходящим из  $u$ , значение 0.

Рассмотрим два случая. В первом случае условие в вершине  $v$  набор  $\sigma$  не выполнил. Тогда если мы рассмотрим набор  $\sigma'$ , который отличается от  $\sigma$  значением ребра  $f$ , то  $\sigma'$  является допустимым и выполняет  $S$ , следовательно  $\sigma'$  выполняет  $C$ , а поскольку  $\sigma$  не выполнял  $C$ , следовательно дизъюнкт  $C$  содержит переменную  $f$ . Во втором случае набор  $\sigma$  выполняет условие в вершине  $v$ , это значит, что есть инцидентное вершине  $v$  ребро  $e$ , что  $\sigma(e) = 1$ . Второй конец ребра  $e$  не лежит в множестве  $A$ , так как вершина  $v$  принадлежит границе множества  $A$ . Рассмотрим набор  $\sigma''$ , который отличается от  $\sigma$  значениями переменных  $f$  и  $e$ , он является допустимым и выполняет все условия  $S$ , следовательно выполняет  $C$ . Значит,  $e$  или  $f$  входят в  $C$ . Таким образом мы получили, что для каждой вершины  $v \in \delta(A)$  как минимум одно из ребер, исходящих из  $v$  содержится в  $C$ . Следовательно, размер дизъюнкта  $C$  не меньше  $|\delta(A)| \geq (c - 1)|A| \geq \frac{c-1}{2}r$ .  $\square$

Конструкция графа называется явной, если граф можно построить за полиномиальное от числа вершин время.

**Лемма 3.3** ([IS11], лемма 6.2). Для любого достаточно большого  $d$  и любого  $m$  существует явная конструкция  $(r, d, 0.5d)$ -границного экспандера с размерами долей  $|X| = |Y| = m$ ,  $r = \Omega(m)$ , причем степени всех вершин из доли  $Y$  не превосходит  $d^2$ .

**Следствие 3.1.** Для любого достаточно большого  $d$  и любого  $m$  существует явная конструкция  $(r, d, 0.4d)$ -границного экспандера с размерами долей  $|X| = m$  и  $|Y| = m - 1$  (а также можно  $|Y| = m - 2$ ),  $r = \Omega(m)$ , причем степени всех вершин из доли  $Y$  не превосходит  $d^2$ .

*Доказательство.* Конструкция графа получается из графа из леммы 3.3 выбрасыванием одной или двух вершин из множества  $Y$ .  $\square$

*Доказательство теоремы 3.1.* Пусть  $d > 5$  для которого выполняется условие следствия 3.1, выберем  $(r, d, 0.4d)$ -границный экспандер  $G$  из следствия 3.1, чтобы в нем было ровно  $n$  вершин. По лемме ширина резолюционного доказательства формулы  $\Phi_G$  не меньше, чем  $\Omega(n)$ . Тогда по теореме 2.1 размер резолюционного доказательства не меньше, чем  $2^{\Omega(n)}$ .  $\square$

## 3.2 Выделение подграфа

Пусть  $G(V, E)$  – неориентированный граф, функция  $h : V \rightarrow \mathbb{N}$  такая, что для каждой вершины  $v$ , степень  $v$  не меньше, чем  $h(v)$ . Построим формулу  $\Psi_G^{(h)}$ , переменные которой соответствуют ребрам графа  $G$ . Для каждой вершины  $v$  записывается условие, что ровно  $h(v)$  ребер, исходящих из данной вершины, равны единице. Формула  $\Psi_G^{(h)}$  представляет конъюнкцию этих условий. Формула  $\Phi_G$  является частным случаем формулы  $\Psi_G^{(h)}$ , когда функция  $h$  тождественно равняется единице.

**Лемма 3.4.** Для любого  $d \in \mathbb{N}$  для всех достаточно больших  $n$  для множества  $V$  из  $n$  элементов и  $h : V \rightarrow \{1, 2, \dots, d\}$  существует явная конструкция графа  $G(V, E)$ , который обладает следующими свойствами:

- $V$  разбивается в объединение двух непересекающихся множеств  $U$  и  $T$ , причем между вершин множества  $U$  нет ребер;
- Степень каждой вершины  $u \in U$  равна  $h(u) - 1$ , степень каждой вершины  $v \in T$  равна  $h(v)$ ;
- $|U| \geq \frac{n}{2} - 2d^2$ .

*Доказательство.* Упорядочим вершины  $v_1, v_2, \dots, v_n$  в порядке неубывания  $h(v_i)$ . Пусть  $k$  – максимальное такое число, что выполняется неравенство  $\sum_{i=1}^k (h(v_i) - 1) < \sum_{i=k+1}^n h(v_i) - d(d - 1)$ . Определим  $U = \{v_1, v_2, \dots, v_k\}$  (если  $k = 0$ , то  $U = \emptyset$ ), а  $T = V \setminus U$ . Очевидно, что  $|U| = k \geq n/2 - d(d - 1)$ . Построим граф  $G$  на множестве вершин  $V$ , изначально граф не содержит ребра, мы будем в него ребра добавлять. Для каждой вершины  $v \in T$  будем называть остаточной степенью разницу между  $h(v)$  и текущей степенью  $v$ . Для каждой вершины  $u \in U$  проводим  $h(u) - 1$  ребро в различные вершины множества  $V \setminus U$  так, чтобы степени всех вершин  $v$  множества  $V \setminus U$  не превосходили бы  $h(v)$ .

Так всегда можно сделать, поскольку по построению множества  $U$  (если  $U$  не пусто) суммарная остаточная степень для вершин из  $T$  больше  $d(d - 1)$ , следовательно при больших  $n$  обязательно найдется как минимум  $d$  вершин остаточной степени хотя бы 1. Пока число вершин в  $T$  положительной остаточной степени больше, чем  $d$ , будем проводить из одной такой вершины ее остаточную степень ребер в остальные вершины положительной степени. В итоге останется не более, чем  $d$  вершин из  $V$  степени не больше, чем  $d$ . Их можно соединить с вершинами из множества  $U$ , а концы использованных ребер перенести из  $U$  в  $T$ . Таким образом мы удалим из  $U$  не больше, чем  $d^2$  вершин. Итого мы получим  $|U| \geq n/2 - 2d^2$ .  $\square$

**Теорема 3.2.** Для любого  $d \in \mathbb{N}$  существует  $D \in \mathbb{N}$ , что для любого достаточно большого  $n$  и для любой функции  $h : V \rightarrow \{1, 2, \dots, d\}$ , где  $|V| = n$ , существует граф  $G(V, E)$ , степень каждой вершины этого графа не превосходит  $D$ , что формула  $\Psi_G^{(h)}$  является невыполнимой и размер любого резолюционного доказательства этой формулы не меньше, чем  $2^{\Omega(n)}$ .

*Доказательство.* По лемме 3.4 построим граф  $G_1(V, E_1)$  и множество  $U \subseteq V$ , что для любой вершины  $v \in U$  ее степень равняется  $h(v) - 1$ , а для любой вершины  $v \in V \setminus U$  ее степень равна  $h(v)$  и при этом  $|U| \geq \frac{n}{2} - 2d^2$ . На вершинах множества  $U$

построим граф  $G(U, E_2)$  из теоремы 3.1. Определим граф  $G(V, E)$ , множество ребер  $E$  которого является объединением  $E_1 \cup E_2$ . Заметим, что ребра из множества  $E_2$  соединяют вершины из множества  $U$ , а ребра из множества  $E_1$  не соединяют пары вершин из множества  $U$  (это следует из построения графа  $G_2$  по лемме 3.4).

В формуле для всех вершин  $v \in V \setminus U$ , степень равна ограничению, следовательно, если формула  $\Psi_G^{(h)}$  выполнима, то в любом выполняющем наборе все переменные, которые соответствуют ребрам, смежным с  $V \setminus U$ , имеют значение 1. После подстановки всем таким переменным значения 1, получается формула  $\Phi_{G_2}$ , которая является невыполнимой по теореме 3.1.

Формула  $\Phi_{G_2}$  получается из  $\Psi_G^{(h)}$  подстановкой, следовательно по лемме 2.2 резолюционное доказательство  $\Psi_G^{(h)}$  не меньше, чем некоторое резолюционное доказательство  $\Phi_G$ , что по теореме 3.1 не меньше, чем  $2^{\Omega(n)}$ .  $\square$

### 3.3 Следствия

**Принцип Дирихле.** Формулы, кодирующие принцип Дирихле, зависят от переменных  $p_{i,j}$ , которые означают, что  $i$ -й кролик сидит в  $j$ -й клетке,  $1 \leq i \leq n$ ,  $1 \leq j \leq n - 1$ , и представляют из себя конъюнкцию таких дизъюнктов:

- Каждый кролик сидит хотя бы где-то:  $\bigvee_{j=1}^{n-1} p_{i,j}$  для всех  $1 \leq i \leq n$ ;
- В каждой клетке сидит не более одного кролика:  $(\neg p_{i,k} \vee \neg p_{j,k})$  для всех  $1 \leq i \neq j \leq n$  и  $1 \leq k \leq n - 1$ .

Будем обозначать такую формулу  $\text{PHP}_n$ . Невыполнимость этой формулы следует из принципа Дирихле.

Для графа  $G$  из следствия 3.1 формула  $\Phi_G$  получается из  $\text{PHP}_n$  подстановкой нулей вместо ребер, которых нет в графе  $G$ , но есть в полном двудольном графе с долями  $|X| = n$ ,  $|Y| = n - 1$  и добавлением нескольких дизъюнктов.

Таким образом, из леммы 3.1 и леммы 2.2 следует, что нижняя оценка  $2^{\Omega(n)}$  на размер резолюционного доказательства формул  $\text{PHP}_n$ . Эта оценка была впервые доказана в работе [Hak85].

**Цейтинские формулы.** Цейтинские формула  $T_G^{(f)}$  строится по произвольному графу  $G(V, E)$  и функции  $f : V \rightarrow \{0, 1\}$ , переменным соответствуют ребра. Для каждой вершины  $v$  записывается условие в КНФ, которое кодирует, что число ребер, исходящих из вершины  $v$  имеет такую же четность, как и число  $f(v)$ .

По функции  $f : V \rightarrow \{0, 1\}$  построим функцию  $h : V \rightarrow \{1, 2\}$ , которая определяется следующим образом:  $h(v) = 2 - f(v)$ . Другими словами, если  $f(v) = 1$ , то и  $h(v) = 1$ , а если  $f(v) = 0$ , то  $h(v) = 2$ . По теореме 3.2 существует такое число  $D$ , что для достаточно больших  $n$  можно построить граф  $G$  на  $n$  вершинах, степени всех вершин не превосходят  $D$ , что размер любого резолюционного доказательства формулы  $\Psi_G^h$  имеет размер не меньше, чем  $2^{\Omega(n)}$ .

Заметим, что условие в вершинах цейтинской формулы  $T_G^{(h)}$  следуют из условий в вершинах формулы  $\Psi_G^h$ . Поскольку резолюционная система доказательств полна относительно импликаций, то каждое условие цейтинской формулы  $T_G^{(h)}$  может быть

выведено из условия  $\Psi_G^h$  с помощью вывода длины  $2^D$ . Следовательно все дизъюнкты цейтинской формулы получаются из дизъюнктов формулы  $\Psi_G^h$  с помощью вывода длины  $O(n)$ . Отсюда следует, что размер резолюционного доказательства формулы  $T_G^{(f)}$  не меньше, чем  $2^{\Omega(n)}$ . Впервые подобная нижняя оценка была доказана в работе [Urq87].

**Полный граф.** Пусть  $K_n$  — полный граф на  $n$  вершинах, функция  $h : V \rightarrow \{0, 1, \dots, d\}$ ,  $d$  — некоторая константа. Пусть формула  $\Psi_{K_n}^{(h)}$  невыполнима. По теореме 3.2 существует такое число  $D$ , что для достаточно больших  $n$  можно построить граф  $G$  на  $n$  вершинах, степени всех вершин не превосходят  $D$ , что размер любого резолюционного доказательства формулы  $\Psi_G^h$  имеет размер не меньше, чем  $2^{\Omega(n)}$ . Граф  $G$  получается из  $K_n$  удалением ребер, поэтому формула  $\Psi_G^{(h)}$  получается из  $\Psi_{K_n}^{(h)}$  подстановкой нуля вместо тех ребер, которых нет в  $G$ . Следовательно, по лемме 2.2 размер резолюционного доказательства  $\Psi_{K_n}^{(h)}$  не меньше, чем  $2^{\Omega(n)}$ .

## 4 Клетчатые задачи

Пусть дана CSP формула  $\phi$  над конечным алфавитом  $W = \{w_1, w_2, \dots, w_k\}$  от переменных множества  $X = \{x_1, x_2, \dots, x_n\}$  и состоит из ограничений  $C_1, C_2, \dots, C_m$ .

Для доказательства нижней оценки на размер дерева поиска противоречия рассматривают следующую игру. Игра определяется невыполнимой CSP формулой  $\phi$ , в игру играют два игрока: Алиса и Боб. Алиса в тайне от Боба выбирает дерево поиска противоречия для  $\phi$  и ставит в корень этого дерева фишку. Ход Алисы состоит в вопросе о значении переменной, которая соответствует вершине, в которой стоит фишка. Ход Боба состоит в том, что он либо выдает значение переменной, о которой его спрашивают, либо предлагает Алисе выбрать значение самостоятельно, предложив ей на выбор подмножество  $W$ , размер которого не меньше 2. Во втором случае мы будем говорить, что Боб делает ход ChooseAny. Алиса передвигает фишку в сына вершины согласно ответу Боба, если Боб походил ChooseAny, то Алиса самостоятельно выбирает, в какую вершину передвинуть фишку. Игра заканчивается, если найдено противоречие, т.е. фишка пришла в лист. Цель Боба на пути от корня до листа как можно больше раз сказать ChooseAny.

**Лемма 4.1.** Пусть для CSP формулы  $\phi$  существует такая стратегия Боба, что при любой стратегии Алисы Боб делает ход ChooseAny как минимум  $t$  раз. Тогда размер любого дерева поиска противоречия для формулы  $\phi$  не меньше  $2^t$ .

*Доказательство.* Рассмотрим какое-нибудь дерево поиска противоречия  $T$  для формулы  $\Phi$ . Построим вероятностное распределение на листьях дерева  $T$ , которое соответствует стратегии Боба и следующей вероятностной стратегии для Алисы. Алиса выбирает дерево  $T$ , а при получении ответа ChooseAny, Алиса выбирает случайный ответ с равной вероятностью. Тогда по условию леммы вероятность оказаться в конкретном листе не превысит  $2^{-t}$ . Поскольку с вероятностью 1 фишка приходит в какой-нибудь лист дерева, то всего листьев не меньше  $2^t$ .  $\square$

Для применения леммы 4.1 удобно описывать стратегию Боба в следующих терминах: Боб хранит некоторую частичную подстановку, которая назначает значения

переменным, которые спрашивала Алиса и, возможно, некоторым другим. При этом текущая подстановка Боба не должна нарушать ограничения. Если Алиса спрашивает значение переменной, значение которой определено текущей подстановкой, то Боб отвечает это значение. Если Алиса спрашивает значение переменной, которая еще не имеет значения в подстановке, которую хранит Боб, то Боб отвечает ChooseAny, добавляет в подстановку значение, которое выбирает Алиса и, возможно, подставляет значение еще некоторых переменных.

## 4.1 Разбиение на доминошки

В этом разделе мы рассмотрим невыполнимую CSP формулу  $\phi_n$ , которая соответствует задаче разбиения клетчатой доски  $n \times n$  на доминошки из двух клеток. Переменные этой задачи соответствуют клеткам доски и принимают значения из алфавита  $\{\leftarrow, \uparrow, \rightarrow, \downarrow\}$ . Стрелка указывает, в какую сторону продолжается доминошка, покрывающая данную клетку. Ограничения соответствуют перегородкам между клетками: для внутренних перегородок запрещается ситуация, когда ровно одна стрелка (из двух клеток, примыкающим к перегородке) направлена на перегородку, а для внешних перегородок запрещается, чтобы стрелка, стоящая в примыкающей к ней клетке, была направлена вне квадрата. Из соображений четности следует, что формула  $\phi_n$  невыполнима при нечетных  $n$ .

**Следствие 4.1.** При нечетных  $n$  для CSP формулы  $\phi_n$  есть дерево поиска опровергний глубины  $O(n)$ , а следовательно размера  $2^{O(n)}$ .

*Доказательство.* Противоречие ищется двоичным поиском: надо разделить квадрат вертикалью на две примерно равные части и сделать запрос про каждую клетку, одна из частей дальше не будет разбиваться на доминошки из-за четности. Потом эту часть поделить пополам горизонталью, сделать запросы ко всем клеткам из нее. Останется часть примерно в два раза меньше исходной по высоте и ширине и сделано примерно  $1.5n$  запросов. Итого за  $O(n)$  запросов противоречие будет найдено.  $\square$

**Теорема 4.1.** Любое дерево поиска противоречия для формулы  $\phi_n$  при нечетных  $n$  имеет размер  $2^{\Omega(n)}$ , а следовательно глубину  $\Omega(n)$ .

Теорема следует из леммы 4.1 и следующей леммы:

**Лемма 4.2.** При нечетных  $n$  для формулы  $\phi_n$  Боб имеет стратегию, гарантирующую  $\Omega(n)$  ответов типа ChooseAny для любой стратегии Алисы.

*Доказательство.* Боб будет хранить текущий набор значений переменных в виде доски  $n \times n$ , которая частично уже заполнена доминошками. По техническим причинам нам будет удобно считать, что  $n$  делится на 3. Если это не так, то Боб может самостоятельно выбрать левый верхний квадрат  $(n-2) \times (n-2)$  или  $(n-4) \times (n-4)$ , а остаток разбить на доминошки каким-нибудь фиксированным способом.

Начиная с левого края, Боб мысленно разбивает доску на вертикальные полосы  $n \times 4$ , которые мы будем называть *слотами*. Поскольку  $n$  нечетно, то останется полоска размера  $n \times 1$  или  $n \times 3$ . Изначально в самом правом слоте Боб выбирает полосу  $n \times 3$ , выровненную по левому краю слота, это полосу мы будем называть

буфером. В дальнейшем буфер будет перемещаться, но он всегда будет занимать левую полосу  $n \times 3$  от какого-то слота. Справа от буфера останется полоса  $n \times 2$  или  $n \times 4$ , Боб замостит ее доминошками. Это изображено на рисунке 1.

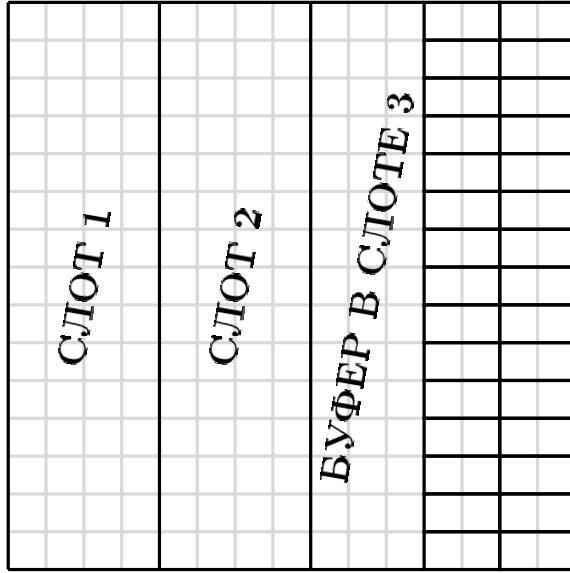


Рис. 1: Слоты, буфер и область правее буфера

Боб поддерживает следующие инварианты:

1. Буфер всегда пуст.
2. Область справа от буфера всегда замощена.
3. Боб разбивает область слева от буфера на прямоугольники размера  $3 \times 2$ . Каждая доминошка из левой части лежит внутри прямоугольника  $3 \times 2$ . Каждый прямоугольник  $3 \times 2$  либо полностью замощен, либо пуст.
4. Каждый слот справа от текущего буфера содержит клетку-запрос Алисы, на которую Боб ответил ChooseAny.

Изначально все инварианты выполняются.

Стратегия Боба будет иметь следующий вид: если Алиса запрашивает уже покрытую клетку, Боб возвращает то, что в ней записано. В противном случае Боб отвечает ChooseAny для некоторого набора значений. После этого, если необходимо, Боб восстанавливает инвариант, сдвигая буфер влево. Если инвариант восстановить не удалось, то дальше Боб действует произвольным образом.

Мы уточним, что именно делает Боб, если запрашиваемая клетка еще не покрыта и докажем, если Боб не может в какой-то момент поправить инвариант, то он к этому моменту уже сделал  $\Omega(n)$  ходов ChooseAny.

Пусть Алиса запрашивает пустую клетку  $c$ . Рассмотрим два случая: 1) клетка  $c$  находится слева от буфера; 2) клетка  $c$  находится внутри буфера.

В первом случае, Боб находит прямоугольник  $3 \times 2$ , содержащий клетку  $c$ , и предоставляет Алисе ответ ChooseAny согласно двум замощениям, показанным на Рисунке 3.

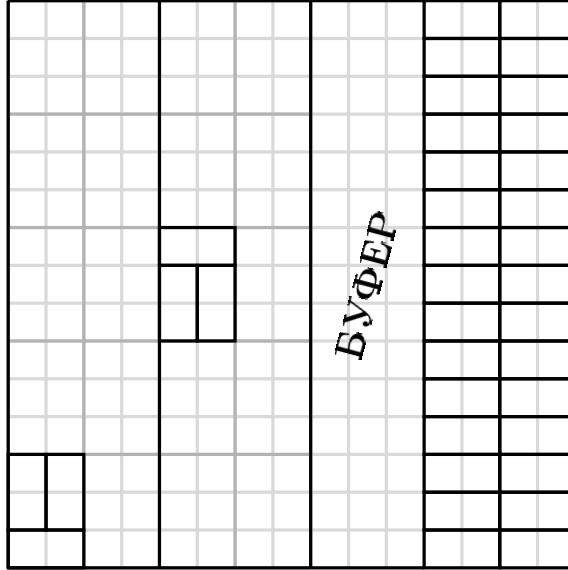


Рис. 2: Инварианты



Рис. 3: Замощения прямоугольника  $3 \times 2$

В случае, когда клетка  $c$  находится внутри буфера, то после хода Боба нарушается инвариант. Возможны два случая: а) слева от буфера есть пустые слоты или б) их нет.

а) Если пустые слоты есть, обозначим самый правый пустой слот через  $S$ . Полоса ширины  $n \times 3$  из слота  $S$ , прижатая к левому краю, будет местом для нового буфера  $B$ . Обозначим старый буфер через  $B'$ . Область между слотом  $S$  и старым буфером  $B'$  обозначим через  $M$ . Найдем в  $M$  пустую горизонтальную полосу  $H$  высотой 3, чтобы и сверху и снизу от нее было кратное 6 число клеток (см. рисунок 4). Всего таких горизонтальных полос  $\Omega(n)$ . Если ни одна из них не окажется пустой, то Боб уже сделал  $\Omega(n)$  ответов ChooseAny (так как клетка левее буфера могла быть заполнена только в результате ответа ChooseAny). Покроем буфер  $B$  и полосу  $H$  прямоугольниками  $3 \times 2$ , как показано на рисунке 4. Оставшуюся часть полосы ширины 1 в слоте  $S$  замостим доминошками. Заметим, что каждая клетка буфера покрыта прямоугольником  $3 \times 2$ , это значит, что в клетке  $c$  Боб может сделать ход ChooseAny согласно рисунку 3. В  $M$  могли быть другие заполненные клетки, но все они расположены в прямоугольниках  $3 \times 2$  и эти прямоугольники полностью заполнены. Остальные прямоугольники  $3 \times 2$  Боб замощает произвольным образом (см. Рисунок 4).

Отметим, что из построения следует, что в каждом слоте справа от буфера Боб дал как минимум один ответ ChooseAny.

б) Пусть пустого слота не нашлось, тогда для каждого слота, кроме слота с буфером, Боб как минимум один раз дал ответ ChooseAny, но всего слотов  $\Omega(n)$ .  $\square$

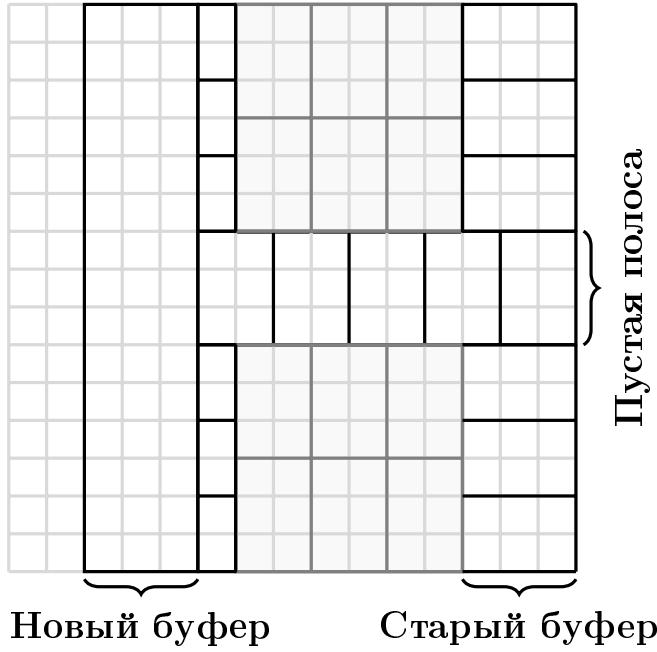


Рис. 4: Замощение при смещении буфера

**Следствие 4.2.** Рассмотрим граф  $G_n$ , вершины которого это клетки квадрата  $n \times n$ , две вершины соединены ребром, если клетки имеют общую сторону. Тогда при нечетных  $n$  формула  $\Phi_G$  является невыполнимой и размер любого дерева поиска противоречия равняется  $2^{\Omega(n)}$ .

*Доказательство.* Формула  $\Phi_G$  тоже кодирует факт, что квадрат  $n \times n$  нельзя разрезать на доминошки при нечетных  $n$ . Стратегия для Боба, которая гарантирует, что он дает  $\Omega(n)$  ответов ChooseAny совпадает с изложенной выше. Алиса вместо клеток запрашивает информацию про границу между клетками. В случае, когда в текущем разбиении про эту границу ответить можно, то Боб отвечает, в противном случае Боб считает, что его спрашивают про одну из клеток, которая примыкает к этой границе кроме одного исключения. Единственное исключение происходит в части левее буфера, когда Боб выделяет прямоугольники  $3 \times 2$ , если Алиса делает запрос к границе этого прямоугольника, то Боб сразу дает ответ 0. Можно проверить, что два разбиения прямоугольника  $3 \times 2$  на доминошки отличаются всеми внутренними границами (см. рисунок 3).  $\square$

## 4.2 Стрелки

Рассмотрим еще одну CSP формулу  $\psi_n$ , которая соответствует клетчатой доске  $n \times n$ . Переменные соответствуют клеткам и принимают значение из множества:  $\{\leftarrow, \nwarrow, \uparrow, \nearrow, \rightarrow, \searrow, \downarrow, \swarrow\}$ .

Ограничения соответствуют перегородкам между клетками:

- Две стрелки, которые стоят в клетках, имеющих общее ребро, различается не больше, чем на 45 градусов.
- Все стрелки в клетках на границе квадрата не направлены наружу.

Невыполнимость формулы  $\phi_n$  следует из теоремы Брауэра, но мы приведем альтернативный способ доказательства невыполнимости  $\phi_n$ , он будет использован для доказательства верхней оценки на размер дерева поиска противоречия.

Рассмотрим замкнутый путь на доске, который проходит по соседним по ребру клеткам. Будем идти вдоль пути и считать сумму поворотов, которые делает стрелка в текущей клетке. Поскольку путь возвращается в исходную клетку, то суммарный поворот кратен  $360^\circ$ .

**Лемма 4.3.** Пусть стрелки расставлены в клетках клетчатого прямоугольника так, что суммарный поворот по замкнутому пути по границе прямоугольника не равен нулю, то внутри этого прямоугольника есть две соседние клетки, в которых стоят стрелки, угол между которыми больше  $45^\circ$ .

*Доказательство.* Доказательство по индукции по размеру прямоугольника. База — прямоугольник  $2 \times 2$ . Для прямоугольника  $R$  с суммой поворотов  $S$ , разделим его на две части, как показано на рисунке 5, посчитаем сумму поворотов  $S_1$  и  $S_2$  на границах меньших прямоугольников  $R_1$  и  $R_2$ . Заметим, что  $S_1 + S_2 = S$ . Значит, хотя бы для одного из меньших прямоугольников сумма поворотов не равна нулю.

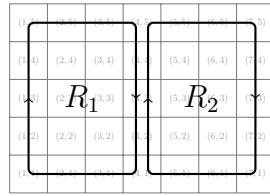


Рис. 5: Деление прямоугольника  $R$  на  $R_1$  и  $R_2$ .

□

**Теорема 4.2.** CSP формула  $\psi_n$  невыполнима.

*Доказательство.* Докажем, что, если рассмотреть замкнутый маршрут вдоль границы квадрата  $n \times n$ , то суммарный поворот будет  $360^\circ$ . Рассмотрим левую верхнюю клетку  $A$  и правую нижнюю  $B$ . Пусть при прохождении от клетки  $A$  до клетки  $B$  суммарный поворот равен  $l + 360^\circ k$ , где  $0 \leq l < 360^\circ$ ,  $k \in \mathbb{Z}$ . В клетке  $A$  стрелка может быть из следующего множества  $\{\rightarrow, \searrow, \downarrow\}$ , а в клетке  $B$  —  $\{\leftarrow, \nwarrow, \uparrow\}$ , соответственно, таким образом  $l \neq 0$ . Заметим, что, если  $k > 0$ , то на пути от клетки  $A$  до клетки  $B$  найдется стрелка  $\nearrow$ , чего не может быть. Следовательно на рассматриваемом участке пути поворот положительный и строго меньше, чем  $360^\circ$ . Аналогично можно рассмотреть участок от клетки  $B$  до клетки  $A$ . Таким образом при прохождении полного контура суммарный поворот не равен нулю и не превосходит  $720^\circ$ , и так как суммарный поворот кратен  $360^\circ$ , то он равен  $360^\circ$ .

Теорема следует из леммы 4.3.

□

**Следствие 4.3.** Для CSP формулы  $\psi_n$  есть дерево поиска опровергений глубины  $O(n)$ , а следовательно размера  $2^{O(n)}$ .

*Доказательство.* Сделаем запросы ко всем граничным клеткам. После этого поделим квадрат по вертикали на две примерно равные части. Так как суммарный поворот по границам частей будет отличен от нуля, то по лемме 4.3 противоречие будет в одной из них. Противоречивую часть поделим по горизонтали и получим задачу для квадрата размером не больше, чем  $(\frac{n}{2} + 1) \times (\frac{n}{2} + 1)$  с известными стрелками на границе. Итого, мы сделали  $1.5n$  запросов и свели задачу к 2 раза меньшей. Глубина дерева будет  $O(n)$ .  $\square$

**Теорема 4.3.** Любое дерево поиска противоречия для CSP формулы  $\psi_n$  имеет размер  $2^{\Omega(n)}$ , а следовательно глубину  $\Omega(n)$ .

Доказательство теоремы 4.3 следует из леммы 4.1 и следующей леммы:

**Лемма 4.4.** Для CSP формулы  $\psi_n$  существует такая стратегия Боба, при которой Боб даст  $\Omega(n)$  ответов типа ChooseAny для любой стратегии Алисы.

*Доказательство.* Структура доказательства леммы 4.4 похожа на структуру доказательства леммы 4.2.

Начиная с левого края, Боб разбивает доску на вертикальные полосы  $n \times 8$ , которые мы будем называть *слотами*. Возможно  $n$  не делится на 8, в этом случае останется неразбитая часть. В самом правом слоте Боб выбирает полосу  $n \times 4$ , выровненную по левому краю, такую полосу будем называть *буфером*. При таком разбиении справа от буфера останется небольшая полоса, в которую мы расставим стрелки так, как показано на рисунке 6.

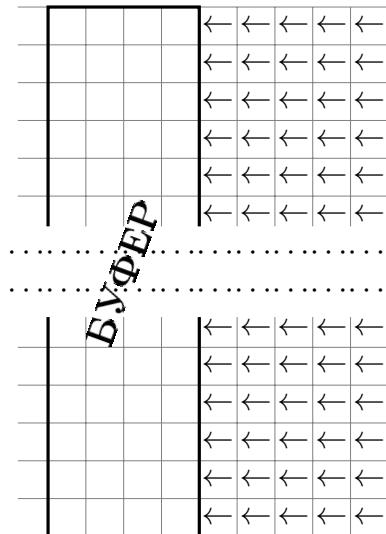


Рис. 6: Замощение правой части

Боб будет поддерживать следующие инварианты:

1. Буфер всегда пуст.
2. Область справа от буфера полностью заполнена. Самая левая полоса  $n \times 1$  из области справа от буфера всегда заполнена стрелками  $\leftarrow$  как на рисунке 6.

3. Все соседи незаполненных клеток левее буфера либо не заполнены, либо заполнены так:  $\rightarrow$ .
4. Пусть  $k$  — это число слотов правее текущего буфера, тогда Боб дал не менее  $\frac{k}{2}$  ответов ChooseAny для клеток правее буфера.
5. Расставленные стрелки не нарушают ограничений.

Отметим, что изначально все инварианты выполнены.

Стратегия Боба будет иметь следующий вид:

1. Если Алиса запрашивает уже заполненную клетку, Боб возвращает то, что в ней записано.
2. Если Алиса запрашивает пустую клетку, то Боб дает ответ ChooseAny и расставляет несколько стрелок, чтобы выполнялись все инварианты.
3. Если инвариант восстановить не удается, то Боб действует произвольным образом.

Мы уточним второй шаг и докажем, что если Боб не может исправить инвариант, то он к этому моменту дал уже  $\Omega(n)$  ответов ChooseAny.

Пусть Алиса запрашивает пустую клетку  $c$ . Мы рассмотрим три случая:

1. клетка  $c$  находится слева от буфера и не имеет общую границу с клетками буфера;
2. клетка  $c$  находится внутри буфера.
3. клетка  $c$  находится на полосе  $n \times 1$ , примыкающей к буферу слева.

В первом случае Боб делает ход ChooseAny, предлагая выбрать из следующих вариантов:

- $\{\rightarrow, \searrow\}$ , если клетка  $c$  находится на верхней границе;
- $\{\rightarrow, \nearrow\}$ , иначе.

Незаполненных соседей по стороне клетки  $c$  Боб заполняет так:  $\rightarrow$  (см. рисунок 7). Несложно заметить, что такое заполнение не нарушает инвариантов.

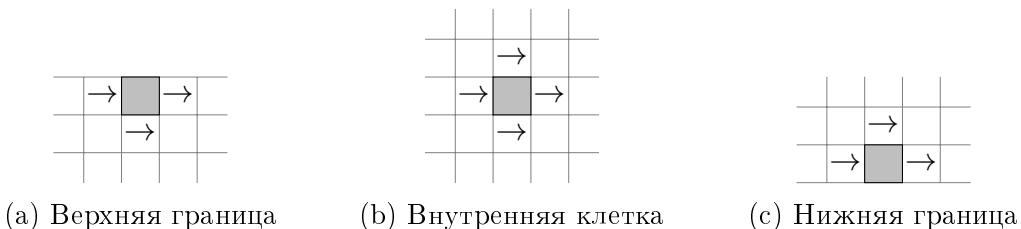


Рис. 7: Три вида окружения стрелки

Во втором случае Алиса делает запрос к клетке буфера. Возможны два варианта:

а) слева от буфера нет пустого слота;

б) слева от буфера есть пустой слот.

а) Пусть пустого слота слева от буфера нет. Значит, в каждом из  $t$  слотов слева от текущего буфера есть заполненная клетка, поскольку на каждый запрос в пустую клетку слева от буфера Боб дает ответ ChooseAny и затрагивает не более двух слотов, то Боб дал не менее  $\frac{t}{2}$  ответов ChooseAny слева от буфера. Поскольку на предыдущем шаге инвариант выполнялся, то Боб дал  $\frac{k}{2}$  ответов ChooseAny для клеток справа от буфера, где  $k$  — это число слотов справа от буфера. Итого, Боб дал не менее  $\frac{t+k}{2} = \Omega(n)$  ответов ChooseAny.

б) Если пустые слоты есть, то выберем из них самый правый. Обозначим его через  $S$ . Полоса  $n \times 4$  из слота  $S$ , прижатая к левому краю, будет местом для нового буфера  $B$ . Обозначим старый буфер через  $B'$ . Область между слотом  $S$  и старым буфером  $B'$  обозначим через  $M$ . Найдем в  $M$  пустую горизонтальную полосу  $H$  высоты 8, которая не прижата ни к верхней, ни к нижней границе.

Заметим, что если такой полосы не найдется, то внутри части  $M$  Боб сделал  $\Omega(n)$  ответов ChooseAny. Во все незаполненные клетки  $M \setminus H$  Боб ставит  $\rightarrow$ . Часть слота  $S \setminus B$  и  $M \cap H$  Боб заполняет как показано на Рисунке 8.

Старый буфер  $B'$  разделим на три части: выше полосы  $H$ , на полосе  $H$  и ниже полосы  $H$ . Заполним буфер  $B'$  одним из двух способов: по шаблону 1 или по шаблону 2, изображенным на рисунке 8. Заметим, что эти два шаблона обладают такими свойствами: одинаково расположенные клетки разных шаблонов заполнены по-разному. Благодаря этому Боб может ответить на запрос Алисы ChooseAny, предложив выбрать два варианта стрелки: из первого шаблона или из второго шаблона.

Нетрудно проверить, что после такого хода инварианты будут выполняться и ограничения нарушены не будут. Пусть  $k$  — это число слотов правее старого буфера  $B'$ ,  $s$  — число слотов между старым и новыми буферами  $B$  и  $B'$ . Тогда всего слотов, которые правее, чем  $B'$  равняется  $k+s+1$ . Справа от  $B'$  Боб давал не менее  $\frac{k}{2}$  ответов ChooseAny, один ответ ChooseAny был дан для буфера  $B'$ , все  $s$  слотов между  $B$  и  $B'$  оказались пустые. Первый запрос в каждый из этих слотов получал ответ ChooseAny, при этом при восстановлении инварианта могло быть затронуто не более двух слотов. Т.е. как минимум  $\frac{s}{2}$  ответов ChooseAny Боб дал для этих  $s$  слотов. Итого число ответов ChooseAny правее  $B'$  не меньше  $\frac{s}{2} + \frac{k}{2} + 1 > \frac{s+k+1}{2}$ .

В третьем случае Алиса делает запрос к клетке, которая является левым соседом буфера. В этом случае Боб дает ответ ChooseAny по правилам первого случая, но одна из клеток окружения заходит на буфер, поэтому нельзя поступить так же, как и в первом случае. Поэтому в этом случае буфер будет передвигаться так, как это описано во втором случае. Единственное отличие в том, что сам буфер будет заполнен только по первому шаблону (см. рисунок 8), в этом случае ограничения нарушатся не будут. Количество ответов ChooseAny в этом случае оценивается немного иначе:  $\frac{k}{2}$  ответов справа от  $B'$  и не менее  $\frac{s+1}{2}$  оставшихся (кроме  $k$  слотов правее  $B'$  остается  $s+1$  слот и каждый ответ ChooseAny заполняет не более 2-х слотов). Таким образом Боб дал как минимум  $\frac{s+k+1}{2}$  ответов ChooseAny для запросов правее буфера  $B$ .  $\square$

**Благодарности.** Авторы благодарят Александра Шеня за плодотворные обсуждения и формулировку задачи о стрелочках.

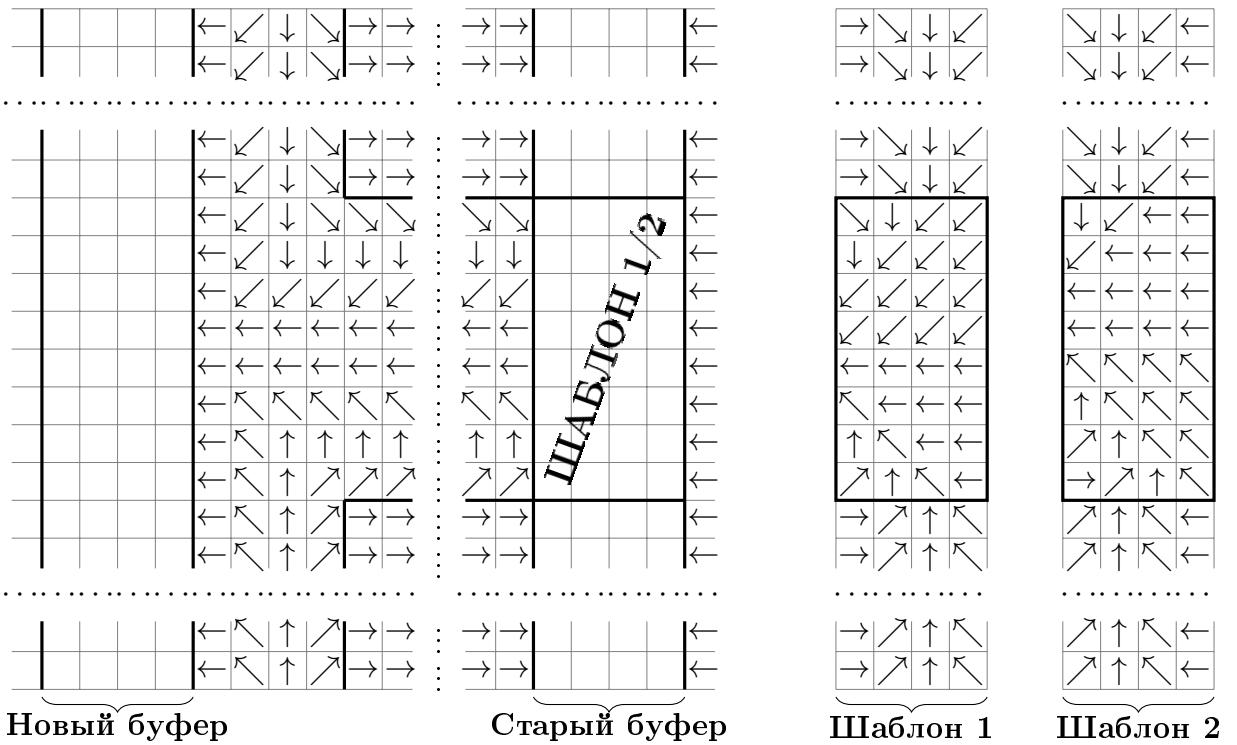


Рис. 8: Замощение промежутка между буферами, шаблоны для старого буфера

## Список литературы

- [Bak95] Andrew B. Baker. Intelligent backtracking on constraint satisfaction problems: Experimental and theoretical results, 1995.
- [BEGJ00] Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM J. Comput.*, 30(5):1462–1484, May 2000.
- [BSW01] E. Ben-Sasson and A. Wigderson. Short proofs are narrow — resolution made simple. *Journal of ACM*, 48(2):149–169, 2001.
- [CDT09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player nash equilibria. *J. ACM*, 56(3):14:1–14:57, May 2009.
- [CS98] P. Crescenzi and R. Silvestri. Sperner’s lemma and robust machines. *Computational Complexity*, 7(2):163–173, May 1998.
- [Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.
- [Hwa04] Cho Yee Joey Hwang. A Theoretical Comparison of Resolution Proof Systems for CSP Algorithms. Master’s thesis, Simon Fraser University, 2004.

- [IS11] Dmitry Itsykson and Dmitry Sokolov. Lower bounds for myopic DPLL algorithms with a cut heuristic. In *Proceedings of the 22nd international conference on Algorithms and Computation*, ISAAC'11, pages 464–473, Berlin, Heidelberg, 2011. Springer-Verlag.
- [Pap94] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, June 1994.
- [Urq87] A. Urquhart. Hard examples for resolution. *JACM*, 34(1):209–219, 1987.
- [Urq11] Alasdair Urquhart. The depth of resolution proofs. *Studia Logica*, 99(1-3):249–364, 2011.