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.Нецветаев, 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

http://www. pdmi.ras.ru /preprint/

Заведующая информационно-издательским сектором Симонова В.Н

ПРЕПРИНТ ПОМИ — 22/2012

О коротких эвристических и оптимальных инвариантных доказательствах [1]

Д.М. Ицыксон **Д.О. Соколов** **О.Ю. Хмелевская**

Санкт-Петербургское отделение
Математического института им. В. А. Стеклова РАН

dmitrits@pdmi.ras.ru
sokolov.dmt@gmail.com
okhmelevskaya@gmail.com

Декабрь, 2012

Аннотация

В этой статье мы рассматриваем два типа неклассических систем доказательств: эвристические системы доказательств и инвариантные системы доказательств, и приводим нетривиальные примеры таких систем доказательств. Мы приводим пример распределенной задачи (Y, D), которая лежит в классе HeurNP, но при этом $Y \notin NP$, если NP $\neq$ coNP, и $(Y, D) \notin$ HeurBPP, если $(\mathrm{NP}, \mathrm{PSamp}) \not\subseteq$ HeurBPP. Для языка L и полинома q определим язык L_q, который состоит из пар (x, r), где x — это элемент L, а r — произвольная битовая строка длины $q(|x|)$. Если $D = \{D_n\}_{n=1}^{\infty}$ — это ансамбль распределений на строках, обозначим $D \times U^q$ — распределение на парах (x, r), где строка x распределена согласно D_n, а r распределена равномерно среди строк длины $q(n)$. Мы показываем, что для каждого языка L из класса AM существует такой полином q, что для любого распределения D с носителем на дополнении языка L, распределенная задача $(L_q, D \times U^q)$ имеет полиномиально ограниченную эвристическую систему доказательств. Поскольку язык пар неизоморфных графов $GNI \in$ AM, то упомянутый результат применим к этому языку GNI.

Мы определяем понятие инвариантных систем доказательств для языка GNI, доказательство в которых остается верным при перестановке вершин графов. Мы строим p-оптимальную инвариантную систему доказательств, т.е. доказательство в любой другой инвариантной системе может быть за полиномиальное время переделано в доказательство в этой системе.

[1]Исследование поддержано Министерством образования и науки Российской Федерации (соглашение 8216 от 06.08.12), грантами РФФИ 12-01-31239 и 11-01-12135, а также грантом президента МК-4108.2012.1.

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, В. Н. Кублановская, Г. В. Кузьмина, П. П. Кулиш,
Б. Б. Лурье, Ю. В. Матиясевич, Н. Ю. Нецветаев, С. И. Репин, Г. А. Серегин, В. Н. Судаков,
О. М. Фоменко

1 Введение

В этой статье мы рассматриваем два типа неклассических систем доказательств: эвристические системы доказательств и инвариантные системы доказательств и приводим нетривиальные примеры таких систем доказательств, которые обладают свойствами, которые не удается добиться от классических систем доказательств.

1.1 Эвристические доказательства

Понятие эвристической системы доказательств вводится в работе [HIMS12]. Распределенной задачей доказательств называется пара, состоящая из языка L и ансамбля распределений D, определенных на дополнении языка L. Эвристической системой доказательств для распределенной задачи доказательств (L, D) называется полиномиальный по времени алгоритм $\Pi(x, w, \delta)$, для которого выполняются следующие свойства:

1. Для каждого $x \in L$ существует строка $w \in \{0,1\}^*$ такая, что $\Pi[x, w, \delta] = 1$;
2. $\Pr_{x \leftarrow D_n}[\exists w\ \Pi[x, w, \delta] = 1] < \delta$.

Говоря неформально, алгоритм Π проверяет, что строка w является доказательством того факта, что $x \in L$. Число δ - это параметр ошибки, чем меньше δ, тем меньшее число неверных утверждений вида $x \in L$ имеют доказательства в Π. Будем говорить, что система Π полиномиально ограничена на $A \subseteq L$, если для каждого $x \in A$ и $\delta \in (0,1)$ существует строка $w \in \{0,1\}^*$ такая, что $|w| < poly(|x|/\delta)$ и $\Pi[x, w, \delta] = 1$.

Интерес к изучению эвристических вычислений возник из структурной теории сложности. Многие структурные утверждения (теорема об иерархии по времени, полные задачи, оптимальные алгоритмы) доказаны в эвристическом случае и остаются открытыми проблемами в классическом смысле. Среди таких результатов стоит отметить результат Харника и др. о существовании полной криптосистемы с открытым ключом [HKN$^+$05], такая криптосистема существует, если декодирующий алгоритм ошибается на небольшой доле входов; теорему об иерархии по времени для вероятностных эвристических вычислений с ограниченной ошибкой, доказанную Фортноу и Сантанамом [FS04] (для классических вероятностных алгоритмов с ограниченной ошибкой теорема об иерархии по времени остается открытой проблемой); результат Ицыксона о существовании полной задачи в эвристическом аналоге класса BPP [Its10] (существование полной задачи в BPP — открытая проблема). Одним из важнейших открытых вопросов в структурной теории сложности является существование p-оптимальной системы доказательств; доказательство в любой другой системе доказательств может быть за полиномиальное время преобразовано в доказательство в p-оптимальной системе доказательств. В 1989 году Крайчек и Пудлак показали, что существование оптимальной системы доказательств для языка пропозициональных тавтологий TAUT эквивалентно существованию оптимального аксептора для TAUT, где аксептор - это алгоритм, который останавливается на словах из языка и не останавливается на дополнении языка; оптимальный аксептор останавливается за минимальное (с точностью до полинома) возможное время на каждом элементе языка. В 1999 году Месснер [Mes99] обобщил этот результат на все языки, допускающие добавление паддинга. В статье [HIMS12] показано, что для любого рекурсивно перечислимого языка L и полиномиально моделируемого распределения D, заданного на дополнении L, существует оптимальный вероятностный эвристический аксептор для задачи (L, D). Эвристический аксептор останавливается на всех словах из языка, а также на небольшой (согласно распределению D) доле входов

из дополнения L. Из этого результата следует существование оптимальной слабо автоматизируемой эвристической системы доказательств. Тем не менее, доказательство эквивалентности существования оптимальной системы доказательств и существования оптимального аксептора не работает в эвристическом случае. Интересно было бы получить пример эвристической системы доказательств, которая при разумных предположениях имела бы короткие доказательства для большего количества утверждений, чем любая классическая система доказательств. В этой статье мы приводим примеры таких эвристических систем доказательств.

Результаты. Мы рассматриваем язык $X = \cup_{n\in\mathbb{N}}\{(C,r) \mid C$ — это схема с n входами, $r \in \{0,1\}^n, \sharp C > r\}$, где $\sharp C$ — это количество выполняющих наборов схемы C, этот язык является PP-полным. Это значит, что при предположении, что $\mathrm{P} \neq \mathrm{NP}$, язык X не содержится в классе P. Для каждого ансамбля распределений $D = \{D_n\}_{n=1}^{\infty}$, носитель которого — это схемы с n входами полиномиального от n размера, мы доказываем, что распределенная задача $(X, D \times U)$ лежит в классе HeurP. Таким образом, мы предъявляем пример нетривиальной распределенной задачи, которая лежит в классе HeurP.

Аналогично, мы рассматриваем язык $Y = \cup_{n\in\mathbb{N}}\{(C,r) \mid C$ — это недетерминированная схема с n входами, $r \in \{0,1\}^n, \sharp C > r\}$, где $\sharp C$ — это количество выполняющих наборов недетерминированной схемы C. Про этот язык доказываем, что он является $\mathrm{PP}\cdot\mathrm{NP}$-полным. Поскольку класс $\mathrm{PP}\cdot\mathrm{NP}$ замкнут относительно дополнений, то из предположения $\mathrm{NP} \neq \mathrm{coNP}$ следует, что язык Y не содержится в классе NP. Для каждого ансамбля распределений $D = \{D_n\}_{n=1}^{\infty}$, носитель которого — это недетерминированные схемы с n входами полиномиального от n размера, мы доказываем, что распределенная задача $(Y, D \times U)$ лежит в классе HeurNP. Кроме того, мы показываем, что, если существует распределенная задача в классе $(\mathrm{NP}, \mathrm{PSamp})$, которая не лежит в классе HeurBPP (это стандартное предположение для сложности в среднем), то найдется такой полиномиально моделируемый ансамбль распределений D'', что распределенная задача $(Y, D'' \times U)$ не лежит в классе HeurBPP. Таким образом, мы предъявляем пример нетривиальной распределенной задачи, которая лежит в классе HeurNP.

Для каждого языка $L \in \mathrm{AM}$ мы показываем, что существует такой полином q, что для любого ансамбля распределений D_n с носителем на строчках полиномиальной от n длины, распределенная задача $(L^q, D \times U^q)$ лежит в классе HeurNP, где $L^q = \{(x,r) \mid x \in L, r \in \{0,1\}^{q(|x|)}\}$, а распределение $D \times U^q$ — это распределение на парах, в которых первый элемент распределен согласно D, а второй элемент равномерно. Очевидно, что язык L^q лежит в классе NP тогда и только тогда, когда в классе NP лежит язык L. Мы так же показываем, что если $(L^q, D \times U^q) \in \mathrm{HeurBPP}$, то и $(L, D) \in \mathrm{HeurBPP}$. Аналогично этому результату можно показать, что существует эвристическая система доказательств для распределенной задачи доказательств $(L^q, D \times U^q)$, где D — произвольный полиномиально ограниченный ансамбль с носителем в дополнении к языку L.

Примером языка из класса AM является язык GNI, состоящий из пар неизоморфных графов на одинаковом числе вершин. Ясно, что $\mathrm{GNI} \in \mathrm{coNP}$, поскольку сертификатом того, что $(G_1, G_2) \notin \mathrm{GNI}$, является перестановка, демонстрирующая изоморфизм графов G_1 и G_2. В настоящее время вопрос $\mathrm{GNI} \overset{?}{\in} \mathrm{NP}$ является открытым, и мы не знаем, как коротко сертифицировать факт, что два графа неизоморфны. Результат А. Кливанса и Д. Мелкебика [KvM99] говорит о том, что, если класс EXP не совпадает с третьим уровнем полиномиальной иерархии Σ_3^{P}, то для бесконечно большого числа длин входов существуют доказательства принадлежности языку GNI субэкспоненциального размера.

Мы рассматриваем язык $\mathrm{GNI}_{pad} = \{(G_1, G_2, r) \mid G_1, G_2$ - неизоморфные графы на n вершинах, r - строка длины $q(n)\}$, где q - некоторый фиксированный полином. Из наших результатов следует, что существует полиномиально ограниченная система доказательств для произвольного распределения на парах изоморфных графов и равномерного распределения на строке r. Отметим, что легко за полиномиальное время моделировать распределения на парах изоморфных графов: к примеру, можно смоделировать равномерное распределение: достаточно сгенерировать один равномернно распределеный случайный граф и случайную перестановку.

1.2 Доказательства, инвариантные относительно перестановок

Мы вводим понятие инвариантных систем доказательств. Для простоты изложения мы рассматриваем эти системы только для языка GNI. Доказательство в инвариантной системе доказательств не меняется, если случайным образом переставить вершины графа. Понятие инвариантной системы доказательств можно обобщить следующим образом.

Рассмотрим последовательность множеств M_n и групп H_n такую, что элементы M_n и H_n можно представить словами полиномиальной от n длины, и равномерное распределение на элементах H_n можно смоделировать за полиномиальное время. (Для языка GNI M_n - это множество всех помеченных графов на n вершинах, H_n — группа перестановок n-элементного множества.) Предположим, что для всех n группа H_n действует на множестве M_n. (Для GNI элемент группы переставляет вершины графа.) Определим язык

$$L = \bigcup_{n \in \mathbb{N}} \{(a, b, n) \mid a, b \in M_n, Orb(a) \cap Orb(b) = \emptyset\},$$

где $Orb(x)$ - орбита элемента $x \in M_n$. Для языка L можно определить систему доказательств, в которой доказательства не меняются при действии группы H_n.

Рассмотрим еще один пример: множество $M_n = H_n = (\mathbb{Z}/n\mathbb{Z})^*$ — мультипликативная группа вычетов по модулю n. Определим действие элемента $x \in H_n$ на M_n так: $y \mapsto x^2 y \bmod n$. К языку L, определенному такими M_n и H_n, в частности, сводится язык квадратичных невычетов, поскольку x является невычетом, если x и 1 находятся в разных орбитах. И, наоборот, x и y лежат в разных орбитах тогда и только тогда, когда $\frac{x}{y}$ является невычетом. Для такого языка L существует полиномиально ограниченная инвариантная система доказательств: доказательством может служить разложение n на простые множители, если такое разложение известно, то проверить само это разложение и проверить, является ли число квадратичным вычетом, можно за полиномиальное время.

Шнор в работе [Sch76] строит оптимальный в наихудшем случае аксептор для всех самосводимых языков из класса NP. В работе [HI12] строится вероятностный аксептор для языка GNI, который имеет оптимальное медианное время по всем перестановкам вершин двух графов.

Результат. В этой работе мы применяем идеи из [HI12] и строим p-оптимальную вероятностную инвариантную систему доказательств для языка GNI.

Структура работы. В разделе 2 мы даем определение основных понятий, в разделе 3 приводим нетривиальный пример языка из HeurP, в разделе 4, используя похожие идеи, мы приводим нетривиальный пример языка из класса HeurNP. В разделе 5 мы показываем, что каждый язык из класса AM с паддингом с распределением, равномерным по паддингу, попадает в

класс HeurNP, и строим полиномиально ограниченную эвристическую систему доказательств для языков из AM с паддингом. В разделе 6 мы определяем понятие инвариантной системы доказательств для языка пар неизоморфных графов и строим p-оптимальную инвариантную систему доказательств.

2 Основные определения

Ансамблем распределений $\{D_n\}_{n=1}^{\infty}$ называется последовательность распределений $D_i : \{0,1\}^* \to [0,1]$ с конечным носителем. Тут n — это параметр задачи, в криптографии его называют параметром надежности. Носителем распределения D_n называется множество строк для которых $D_n(x) > 0$, обозначается $\operatorname{supp} D_n$.

Равномерным ансамблем мы будем называть ансамбль U_n для которого $U_n(x) = 2^{-n}$, если $|x| = n$.

Для двух ансамблей распределений D и F мы определим ансамбль $\alpha D + (1-\alpha)F$ для $0 \le \alpha \le 1$ естественным равенством $(\alpha D + (1-\alpha)F)_n(x) = \alpha D_n(x) + (1-\alpha)F_n(x)$. Мы также определим ансамбль распределений $D \times F$, носитель которого содержится в множестве пар строчек и определяется равенством $(D \times F)_n(x,y) = D_n(x)F_n(y)$. В частности, $D \times U$ — это такой ансамбль, при котором $(D \times U)_n(x,r) = D_n(x)2^{-n}$, если $|r| = n$. Если $g : \{0,1\}^* \to \{0,1\}^*$, а D — ансамбль распределений, то обозначим $g(D)$ — ансамбль распределений, где $g(D)_n(x) = \sum\limits_{y \in g^{-1}(x)} D_n(y)$. Аналогично, если $h : \{0,1\}^* \times \mathbb{N} \to \{0,1\}$, то $h(D)$ — ансамбль распределений, где $h(D)_n(x) = \sum\limits_{y : g(y,n) = x} D_n(y)$.

Будем называть ансамбль полиномиально ограниченным, если существует такой полином p, что для любого $x \in \operatorname{supp} D_n$ выполняется $|x| \le p(n)$. Ансамбль D_n называется полиномиально моделируемым, если существует такой полиномиальный по времени вероятностный алгоритм G, распределение вероятностей выхода которого на входе 1^n совпадает с D_n. Множество полиномиально моделируемых ансамблей будем обозначать PSamp. Любой полиномиально моделируемый ансамбль является полиномиально ограниченным.

Распределенной задачей (распознавания) мы будем называть пару (L, D), где L — это язык, а D — это ансамбль распределений.

В этой работе мы рассматриваем три основных модели вычислений: детерминированные машины Тьюринга, недетерминированные машины Тьюринга и вероятностные машины Тьюринга с ограниченной ошибкой. Мы считаем, что эти модели вычисления выдают в качестве ответа только 0 или 1. Для каждой из этих моделей вычислений определены понятия времени работы: для детерминированной машины — это просто время работы, для недетерминированной машины — это максимальное время работы по всем недетерминированным выборам, а для вероятностной машины с ограниченной ошибкой — максимум по всем случайным битам. Для каждой из этих моделей вычислений можно определить ответ на данном входе. Для вероятностной машины с ограниченной ошибкой ответ на данном входе — это такой ответ, который принимается с вероятностью хотя бы $\frac{3}{4}$ (этот ответ может и не существовать, в таком случае считаем, что ответ этой машины $\perp$). Классы $\mathrm{P}, \mathrm{NP}, \mathrm{BPP}$ можно определить, как множество языков, которые распознаются за полиномиальное время детерминированными, недетерминированными, вероятностными с ограниченной ошибкой машинами Тьюринга.

Определение 2.1. $\mathrm{HeurP}(\mathrm{HeurNP}, \mathrm{HeurBPP})$ — это класс распределенных задач распозна-

вания (L, D), для которых существует детерминированный (недетерминированный и вероятностный соответственно) алгоритм $A(x, \delta; n)$ и полином p такие, что:

1. Для всех $x \in \operatorname{supp} D_n$ время работы $A(x, \delta; n)$ ограничено $p(\frac{n}{\delta})$;
2. $\Pr_{x \leftarrow D_n}[A(x, \delta; n) \neq L(x)] < \delta$, в случае вероятностного алгоритма вероятность берется так же и по случайным битам алгоритма A.

Распределенной задачей доказательств называется пара (L, D), где L — язык, а D — ансамбль распределений, носитель которого содержится в дополнении языка L.

Следующее определение является детерминированной версией вероятностной эвристической системы доказательств из [HIMS12].

Определение 2.2. Эвристической системой доказательств для распределенной задачи доказательств (L, D) называется алгоритм $\Pi(x, w, \delta)$, для которого выполняются следующие свойства:

1. Для каждого $x \in L$ существует $w \in \{0, 1\}^*$, что $\Pi(x, w, \delta) = 1$;
2. $\Pr_{x \leftarrow D_n}[\exists w\ \Pi[x, w, \delta] = 1] < \delta$;
3. Время работы $\Pi(x, w, \delta)$ ограничено полиномом относительно $\frac{|x|+|w|}{\delta}$.

Эвристическая система доказательств называется полиномиально ограниченной, если существует такой полином p, что для каждого $x \in L$ найдется w, что $\Pi(x, w, \delta) = 1$ и размер w ограничен $p(\frac{|x|}{\delta})$.

Также в [HIMS12] дается определение вероятностного эвристического аксептора.

Определение 2.3. Вероятностным эвристическим аксептором для распределенной задачи доказательств (L, D) называется алгоритм $A(x, \delta)$, для которого выполняются следующие свойства:

1. Для каждого $x \in L$ $A(x, \delta) = 1$;
2. $\Pr_{x \leftarrow D_n, A}[A(x, \delta) = 1] < \delta$.

Аксептор называется полиномиально ограниченным, если для всех $x \in L$ время работы $A(x, \delta)$ ограничено полиномом.

3 Примеры задач в HeurP

Нам потребуется понятие булева сэмплер. Булев сэмплер — это вероятностный алгоритм S, который получает на вход натуральное число n и рациональные числа ϵ, δ. S имеет оракульный доступ к некоторой функции $f : \{0, 1\}^n \to \{0, 1\}$, S может сделать несколько неадаптивных запросов к функции f и выдать число из интервала $[0, 1]$. Обозначим $\overline{f} = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)$. Для каждой функции $f : \{0, 1\}^n \to \{0, 1\}$ должно выполняться $\Pr[|S^f(n, \epsilon, \delta) - \overline{f}| \geq \epsilon] < \delta$. Булев сэмплер называется усредняющим, если он выдает среднее арифметическое запрошенных значений.

Теорема 3.1 ([Gol11]). Существует усредняющий булев сэмплер S, который использует n случайных битов, делает $O(\frac{1}{\epsilon^2\delta})$ запросов к функции и работает полиномиальное относительно $n, \frac{1}{\epsilon}, \frac{1}{\delta}$ время.

Пусть $f : \{0,1\}^* \to \{0,1\}$ — некоторая функция. Определим язык $L_f = \cup_{n\in\mathbb{N}}\{r \in \{0,1\}^n \mid 0.r < \overline{f_n}\}$, где $\overline{f} = \frac{1}{2^n}\sum_{x\in\{0,1\}^n} f(x)$.

Теорема 3.2. Если f вычислима за полиномиальное время, то $(L_f, U) \in \mathrm{HeurP}$.

Доказательство. Опишем алгоритм $A(x,\delta)$ (параметр n можно не указывать, так как он совпадает с $|x|$). Пусть S — усредняющий булев сэмплер из теоремы 3.1.

- Если $\delta \le 2^{-n+2}$, то вычислить $\overline{f}$ и выдать 1, если $0.x < \overline{f_n}$, и 0 иначе.
- Запустить сэмплер $S^f(n, \delta/4, \delta/4)$, который будет использовать строку x вместо случайных битов. Обозначим через ν_n результат его работы. Если $\nu_n > 0.x$, то выдать 1, иначе выдать 0.

Если $\delta \le 2^{-n+2}$, то алгоритм не ошибается, а время работы алгоритма равняется произведению 2^n на полином от n, что полиномиально относительно $\frac{n}{\delta}$.

Во втором случае время работы алгоритма ограничено полиномом от n и $\frac{1}{\delta}$, это следует из того, что функция f вычислима за полиномиальное время, и свойств сэмплера S. Оценим равномерную долю входов x, при которых алгоритм $A(x,\delta)$ выдает неверный ответ. Множество таких входов содержится в объединении следующих двух множеств:

1. Множество $x \in \{0,1\}^n$, при которых сэмплер S не выдает $\delta/4$ приближение к $\overline{f_n}$. Равномерная мера этого множества не превосходит $\frac{\delta}{4}$ по определению сэмплера.
2. множество $x \in \{0,1\}^n$, при которых $|0.x - \overline{f_n}| \le \delta/4$. Равномерная доля таких x не превосходит $\frac{\delta}{2} + \frac{1}{2^n}$.

Итого, равномерная мера $x \in \{0,1\}^n$, при которых $A(x,\delta) \ne L_f(x)$, не превосходит $\frac{3}{4}\delta + 2^{-n} < \delta$ при $\delta > 2^{-n+2}$. □

Рассмотрим язык $X = \cup_{n\in\mathbb{N}}\{(C,r) \mid C$ — это схема с n входами, $r \in \{0,1\}^n, \sharp C > r\}$, где $\sharp C$ — это количество выполняющих наборов схемы C.

Напомним, что класс сложности PP состоит из языков L, для которых существует вероятностная полиномиальная по времени машина Тьюринга M, что для каждого x выполняется $x \in L \iff \Pr_M[M(x) = L(x)] \ge 1/2$.

Предложение 3.1. Язык X является PP-полным.

Доказательство. Сначала покажем, что $X \in \mathrm{PP}$. Опишем вероятностный алгоритм из класса PP. Пусть на вход подается схема $C(x_1, x_2, \ldots, x_n)$ и число $0 \le r \le 2^n - 1$. Рассмотрим следующий предикат $P(b, x_1, x_2, \ldots, x_n) = (b \wedge C(x_1, x_2, \ldots, x_n)) \vee ((\overline{x_1x_2\ldots x_n} \le 2^n - r - 1) \wedge \neg b)$, где $\overline{x_1x_2\ldots x_n}$ — обозначает число в битовой записи.

- Предикат P зависит от $n+1$ булевой переменной.
- Предикат P вычислим за полиномиальное время.

- Число выполняющих наборов предиката P равняется $\sharp C + 2^n - r - 1$. Таким образом, $\sharp C > r$ тогда и только тогда, когда число выполняющих наборов P не менее 2^n.

Алгоритм состоит в следующем: рассмотреть случайный набор значений переменных $b, x_1, x_2, \ldots, x_n$ и выдать $P(b, x_1, x_2, \ldots, x_n)$.

Теперь покажем, что любой язык из PP сводится к X. Пусть $L \in$ PP. Пусть M — такая полиномиальная по времени вероятностная машина Тьюринга, что при каждом x выполняется $x \in L$ тогда и только тогда, когда $\Pr[M(x) = 1] \geq \frac{1}{2}$. Машина M на входе x использует $q(|x|)$ случайных битов для некоторого полинома q. Построим схему C, которая имеет $q(|x|)$ входов и моделирует машину M на входе x, используя случайные биты из входа. Тогда сведение отображает $x \mapsto (C, 2^{q(|x|)-1} - 1)$. □

Теорема 3.3. Пусть ансамбль D является полиномиально ограниченным и носитель D_n состоит из схем с n входами, тогда $(X, D \times U) \in$ HeurP.

Доказательство. Опишем алгоритм $B(x, \delta; n)$. Нас интересуют только строки ненулевой вероятности согласно распределению $D \times U$. Строка x имеет вид (C, r), где C — схема от n переменных, а $r \in \{0,1\}^n$, размер схемы C ограничен полиномом от длины n, это следует из полиномиальной ограниченности ансамбля D. Пусть $A_C(x, \delta)$ — это алгоритм из доказательства теоремы 3.2, который решает язык L_{g_C}, который соответствует такой функции g_C, которая равна тождественному нулю на всех длинах входа, кроме длины n, а на длине n вычисляется схемой C. По построению алгоритм A_C пользуется схемой C как оракулом, поэтому время его работы — это полином от размера схемы C и $\frac{|x|}{\delta}$.

Алгоритм $B((C, r), \delta; n)$ запускает $A_C(r, \delta)$ и выдает его ответ. Очевидно, что время работы B ограничено полиномом относительно $\frac{n}{\delta}$.

$\Pr_{(C,r) \leftarrow (D \times U)_n}[B((C, r), \delta; n) \neq X(C, r)] = \mathrm{E}_{C \leftarrow D_n}[\Pr_{r \leftarrow U_n}[A_C(r, \delta) \neq L_{g_C}(r)]] < \delta$.

□

4 Примеры задач в HeurNP

Теорема 4.1. Если f вычислима за полиномиальное время недетерминированным алгоритмом (т.е. язык $\{x \in \{0,1\}^* \mid f(x) = 1\} \in$ NP), то $(L_f, U) \in$ HeurNP.

Доказательство. Доказательство будет похоже на доказательство теоремы 3.2. Опишем недетерминированный алгоритм $A(x, \delta)$ (параметр n можно не указывать, так как он совпадает с $|x|$). Пусть S — усредняющий булев сэмплер из теоремы 3.1.

- Если $\delta \leq 2^{-n+2}$, то угадать подсказки для всех $x \in \{0,1\}^n$, вычислить $\overline{f}$ и выдать 1, если $0.x \leq \overline{f_n}$, и 0 иначе.

- Если $\delta > 2^{-n+2}$, то подсказка алгоритма будет состоять из q строк $w_1, w_2, \ldots, w_q$, где $q = O(\frac{1}{\delta^3})$ — это число запросов, которые делает сэмплер $S^f(n, \delta/4, \delta/4)$, используя строку x как строку случайных битов. Запустить сэмплер $S^f(n, \delta/4, \delta/4)$, который использует строку x как строку случайных битов, для вычисления функции f в i-ой точке будет использоваться подсказка w_i. Обозначим через ν_n результат работы сэмплера. Если $\nu_n > 0.x$, то выдать 1, иначе выдать 0. Заметим, что сэмплер является монотонным, так как он усредняет полученные значения, значит, если некоторые подсказки были неверные, и

значения стали нулями, вместо единицы, то результат ν_n мог разве что уменьшится, т.е. ответ мог измениться с 1 на 0, но не наоборот.

Если $\delta \leq 2^{-n+2}$, то при правильных подсказках для всех x, при которых $f(x) = 1$, алгоритм не ошибается, а время работы алгоритма равняется произведению 2^n на полином от n, что полиномиально относительно $\frac{n}{\delta}$.

Во втором случае время работы алгоритма ограничено полиномом от n и $\frac{1}{\delta}$, это следует из того, что функция f вычислима за полиномиальное недетерминированное время и свойств сэмплера S. Аналогично доказательству теоремы 3.2 равномерная доля входов x, при которых алгоритм $A(x, \delta)$ выдает неверный ответ, меньше $\frac{3}{4}\delta + 2^{-n} < \delta$ при $\delta > 2^{-n+2}$. □

Недетерминированной схемой мы будем называть схему, входные вершины которой разбиты на две части: вершины первой части мы называем входами, вершины второй части подсказками. Значение такой схемы на входе x равняется 1, если существует такая подсказка y, при которой значение $C(x; y) = 1$. Мы также можем определить $C(x) = \exists y C(x; y)$.

Рассмотрим язык $Y = \cup_{n \in \mathbb{N}} \{(C, r) \mid C$ — это недетерминированная схема с n входами, $r \in \{0,1\}^n, \sharp C > r\}$, где $\sharp C$ — это количество выполняющих наборов схемы C, другими словами, если схема C использует m битов подсказки, то $\sharp C = |\{x \in \{0,1\}^n \mid \exists y \in \{0,1\}^m : C(x, y) = 1\}|$.

Определение 4.1. Класс PP·NP состоит из языков L, для которых существует вероятностная полиномиальная по времени машина Тьюринга M, что для всех x выполняется $x \in L$ тогда и только тогда, когда $\Pr[M(x) \in SAT] \geq \frac{1}{2}$.

Предложение 4.2. Язык Y является $\mathrm{PP} \cdot \mathrm{NP}$-полным.

Доказательство. Доказательство аналогично доказательству предложения 3.1. Опишем PP-сведение языка Y к SAT. Пусть на вход подана схема C от n входов (схема может использовать подсказку) и число $0 \leq r \leq 2^n - 1$. Построим предикат $P(b, x_1, x_2, \ldots, x_n) = \exists w (b \wedge C(x_1, x_2, \ldots, x_n; w)) \vee ((\overline{x_1 x_2 \ldots x_n} \leq 2^n - r - 1) \wedge \neg b)$.

Если мы зафиксируем значения переменных $b, x_1, x_2, \ldots, x_n$, то легко построить формулу $\phi_{b,x}$, которая выполнима тогда и только тогда, когда $P(b, x_1, x_2, \ldots, x_n) = 1$. Вероятностная машина выбирает случайным образом значения $b, x_1, x_2, \ldots, x_n$ и выдает соответствующую формулу $\phi_{b,x}$.

Теперь покажем, что любой язык из PP·NP сводится к Y. Пусть некоторая схема C по входу r вычисляет пропозициональную формулу $C(r)$. А схема A получает на вход строку y, строку r и описание схемы C и выдает результат подстановки значений из строки y в формулу $C(r)$. Пусть C — это схема, моделирующая машину M на входе x, входами этой схемы являются случайные биты машины M. Схема C по входу r вычисляет пропозициональную формулу $C(r)$. Пусть недетерминированная схема A_C получает на вход строку r и выдает 1 тогда и только тогда, когда существует такая строка y, что результат ее подстановки в формулу $C(r)$ равен 1. Сведение L к Y отображает вход x в $(A_C(r; y), 2^{q(n)-1} - 1)$. □

Теорема 4.2. Пусть ансамбль D является полиномиально ограниченным и носитель D_n состоит из недетерминированных схем с n входами, тогда $(Y, D \times U) \in \mathrm{HeurNP}$.

Доказательство. Доказательство полностью аналогично доказательству теоремы 3.3. □

Предложение 4.3. По каждой задаче $(L, D) \in (\mathrm{NP}, \mathrm{PSamp})$ можно построить распределение $D'' \in \mathrm{PSamp}$, что если $(Y, D'' \times U) \in \mathrm{HeurBPP}$, то $(L, D) \in \mathrm{HeurBPP}$.

Доказательство. Рассмотрим произвольную распределенную задачу $(L, D) \in (\mathrm{NP}, \mathrm{PSamp})$. Рассмотрим сведение g языка L к языку CircuitSAT. Пусть распределение $D' = g(D)$ — соответствует распределению образов функции g, если на вход подавать элементы, распределенные согласно D. Нетрудно понять, что если $(\mathrm{CircuitSAT}, D') \in \mathrm{HeurBPP}$, то и $(L, D) \in \mathrm{HeurBPP}$. Пусть функция $h : \{0,1\}^* \times \mathbb{N} \to \{0,1\}^*$ преобразует схемы в недетерминированные схемы, причем старый вход становится подсказкой, новая схема содержит n фиктивных входов, в реальности от этих входов схема не зависит. Т.е. функция $h(C, n) = E$, где $E(y; x) = C(x)$ и размерность y равняется n. От подсказки схема зависит точно также, как и старая схема. Пусть $D'' = h(D')$. Носитель распределения D''_n состоит из недетерминированных схем $h(C)$ с n входами, где C — обычная схема. Если схема C выполнима, то для всех $0 \le r \le 2^n - 1$ выполняется $(h(C), r) \in Y$ и наоборот, если C невыполнима, то для всех $0 \le r \le 2^n - 1$ выполняется $(h(C), r) \notin Y$.

Пусть алгоритм $A(x, \delta; n)$ решает задачу $(Y, D'' \times U)$ в классе HeurBPP, опишем алгоритм $B(x, \delta; n)$, который генерирует случайную строку $r \in \{0,1\}^n$ и запускает алгоритм $A((h(g(x)), r), \delta; n)$.

$\Pr_{x \leftarrow D_n}[B(x, \delta; n) \neq L(x)] = \Pr_{x \leftarrow D_n, r \leftarrow U_n}[A((h(g(x)), r), \delta; n) \neq Y(x, r)] = \Pr_{x \leftarrow D''_n, r \leftarrow U_n}[A(x, \delta; n) \neq Y(x, r)] < \delta.$ □

Следствие 4.1. Существует такое распределение D'', что из включение $(Y, D'' \times U) \in \mathrm{HeurBPP}$ следует, что $(\mathrm{NP}, \mathrm{PSamp}) \subseteq \mathrm{HeurBPP}$.

Доказательство. Надо применить предложение 4.3 к полной задаче в классе $(\mathrm{NP}, \mathrm{PSamp})$. □

5 Эвристическая система доказательств для AM

Напомним, что такое класс AM. Язык L лежит в классе AM, если существует такая полиномиальная по времени недетерминированная машина M и такой полином q, что для всех строк x выполняется:

- если $x \in L$, то $\Pr_{z \leftarrow U_{q(|x|)}}[M(x, z) = 1] > \frac{3}{4}$,
- если $x \notin L$, то $\Pr_{z \leftarrow U_{q(|x|)}}[M(x, z) = 1] < \frac{1}{4}$.

Известно, что при $x \in L$ ошибку можно сделать нулевой, т.е. язык L лежит в классе AM, то существует такая полиномиальная по времени недетерминированная машина M и такой полином q, что для всех строк x выполняется:

- если $x \in L$, то $\Pr_{z \leftarrow U_{q(|x|)}}[M(x, z) = 1] = 1$,
- если $x \notin L$, то $\Pr_{z \leftarrow U_{q(|x|)}}[M(x, z) = 1] < \frac{1}{4}$.

Мы покажем, что эти требования можно еще усилить, получив вероятность ошибки $2^{-p(|x|)}$, при том что число случайных битов, используемых Артуром будет всего $q(|x|) + O(p(|x|))$. Для этого нам понадобятся чуть более сильные сэмплеры. Будем называть сэмплер монотонным, если при изменении значения функции f в одной точке с 0 на 1, приближение, которое выдает сэмплер, не может уменьшиться. Усредняющий сэмплер является монотонным.

Теорема 5.1 ([Gol11]). Существует монотонный булев сэмплер S, который использует $n + O(\log \frac{1}{\delta})$ случайных битов, делает $O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$ запросов к функции и работает полиномиальное относительно $n, \frac{1}{\epsilon}, \log \frac{1}{\delta}$ время. Кроме того, если сэмплеру на вход подана функция, которая тождественно равна единице, то он выдает 1 с вероятностью 1.

Одна из конструкций сэмплера, которая удовлетворяет условию теоремы 5.1 - сэмплер, основанный на медиане усреднений.

Теорема 5.2. Если L лежит в классе AM, то существует полином q и константа $c > 0$, что для любого полинома p существует такая полиномиальная по времени недетерминированная машина M, что для всех строк x выполняется:

- если $x \in L$, то $\Pr_{z \leftarrow U_{q(|x|)+cp(|x|)}}[M(x,z) = 1] = 1$,
- если $x \notin L$, то $\Pr_z \leftarrow U_{q(|x|)+cp(|x|)}[M(x,z) = 1] < 2^{-p(|x|)}$.

Доказательство. Рассмотрим машину M' и полином q, что для всех строк x выполняется:

- если $x \in L$, то $\Pr_{z \leftarrow U_{q(|x|)}}[M'(x,z) = 1] = 1$,
- если $x \notin L$, то $\Pr_{z \leftarrow U_{q(|x|)}}[M'(x,z) = 1] < \frac{1}{4}$.

Рассмотрим сэмплер $S(n, \epsilon, \delta)$ из теоремы 5.1, пусть константа c такая, что S использует $n + c \log \frac{1}{\delta}$ случайных битов. Опишем работу недетерминированной машины M на входе (x, z), где $|z| = q(|x|) + cp(|x|)$. M запускает сэмплер S с параметрами $\delta = 2^{-p(|x|)}$, $\epsilon = \frac{1}{8}$ для вычисления среднего значения $M'(x,z)$ по всем $|z| = q(|x|) + cp(|x|)$ и строкой z в качестве случайных битов. Каждый раз, когда S запрашивает значение $M'(x,z)$, подсказка для M' угадывается. Обозначим полученное значение ν. Если $\nu < \frac{1}{2}$, то выдадим 0, иначе выдадим 1. Если $x \notin L$, то вероятность того, что $\nu \geq \frac{1}{2}$, меньше, чем $\delta = 2^{-p(|x|)}$ в силу монотонности сэмплера. Если $x \in L$, то ответ сэмплера должен быть равен 1 с вероятностью 1 при правильно угаданных подсказках для всех запросов к значениям $M'(x,z)$. □

Пусть q — некоторый полином, для любого языка L обозначим $pad_q(L) = \{(x,r) \mid x \in L, r \in \{0,1\}^*, |r| \geq q(|x|)\}$.

Теорема 5.3. Пусть язык $L \in$ AM, и ансамбль D является полиномиально ограниченным, тогда существует такой полином q, что $(pad_q(L), D \times U^{(q)}) \in$ HeurNP, где $U_n^{(q)} = U_{q(n)}$.

Доказательство. Пусть M — это недетерминированная машина Тьюринга из теоремы 5.2, построенная для языка L и полинома p, совпадающего с полиномом q.

Не умаляя общности, будем считать, что полином q оценивает сверху число случайных битов, необходимых для протокола Артура-Мерлина для всех $x \in \operatorname{supp} D_n$.

Опишем недетерминированный алгоритм $A(x, \delta; n)$ для задачи $(pad_q(L), D \times U^{(q)})$, будем считать, что x имеет вид (y, r), где $|r| = q(n)$ (вероятность строк, которые не имеют такой вид, равна нулю). Если $\delta < 2^{-q(n)}$, то перебираем все $z \in \{0,1\}^{q(n)(c+1)}$, угадаем подсказки для машины M для каждой строки $M(y,z)$, если получилось угадать все подсказки, то принять, иначе отвергнуть.

Если $\delta \geq 2^{-q(n)}$, то запускаем машину M на входе (y, r) и выдаем ее ответ. Для каждого x доля строк r, при которых выданный ответ неверный, меньше $2^{-q(n)} = \delta$.

Если $\delta < 2^{-q(n)}$, то недетерминированный алгоритм $A((y,r),\delta;n)$ принимает, если $y \in L$, и отвергает иначе. Время работы ограничено $(\frac{n}{\delta})^{c+1}$.

Если $\delta \geq 2^{-q(n)}$, то время работы ограничено временем работы машины M на носителе D_n, т.е. полиномом от n. $\square$

Следствие 5.1. Пусть язык $L \in$ AM, и ансамбль D является полиномиально ограниченным с носителем на дополнении L, тогда существует такой полином q, что для распределенной задачи доказательств $(pad_q(L), D \times U^{(q)})$ существует полиномиально ограниченная система доказательств.

Доказательство. Такой системой доказательств будет построенный в доказательстве теоремы алгоритм $A((y,r),\delta;n)$, доказательством будет являться недетерминированная подсказка. Из доказательства видно, что на для каждого элемента из языка всегда есть доказательство размера $poly(\frac{n}{\delta})$, а условие об ошибке на дополнении языка прямо следует из теоремы. $\square$

Заметим, что при любом полиноме q язык L сводится по Карпу к языку $pad_q(L)$ и наоборот. Из этого следует, что $pad_q(L) \in$ NP тогда и только тогда, когда $L \in$ NP.

На самом деле можно доказать и более сильное утверждение, чем существование обыкновенного сведения:

Предложение 5.4. Для каждого полиномиально ограниченного ансамбля D и полинома q, если $(pad_q(L), D \times U^{(q)}) \in$ HeurBPP, то $(L, D) \in$ HeurBPP.

Доказательство. Пусть алгоритм $B(x,\delta,n)$ решает задачу $(pad_q(L), D \times U^{(q)})$ в классе HeurBPP. Опишем алгоритм $A(x,\delta;n)$ для задачи (L,D). Выберем случайную строку r длины $q(n)$ и запустим алгоритм $B((x,r),\delta;n)$. $\square$

Следствие 5.2. Для каждого полиномиально ограниченного ансамбля D с носителем в дополнении L и полинома q, если $(pad_q(L), D \times U^{(q)})$ решается вероятностным полиномиально ограниченным аксептором, то для любого полиномиально ограниченного распределения D' с носителем на L задача $(L, \frac{D+D'}{2}) \in$ HeurBPP.

6 Доказательства, устойчивые относительно перестановок

Язык GNI состоит из пар неизоморфных графов с одинаковым числом вершин.

Определение 6.1. Инвариантной системой доказательств (или системой доказательств, инвариантной относительно перестановок) для языка GNI называется вероятностный алгоритм $\Pi(x,w)$, обладающий следующими свойствами

1. Время работы $\Pi(x,w)$ полиномиально от $|x|$ и $|w|$.

2. (полнота) Для любой пары графов $(G_0, G_1) \in GNI$ (каждый граф имеет n вершин) найдется строка w такая, что $\Pr_{\pi_0,\pi_1 \leftarrow S_n, \Pi}[\Pi(\pi_0(G_0), \pi_1(G_1), w) = 1] \geq \dfrac{3}{4}$, где вероятность берется по случайным равномерно распределенным на S_n перестановкам π_0, π_1 и по случайным битам алгоритма Π. Такая строка w называется Π-доказательством для x.

3. (корректность) Для любой пары $(G_0, G_1) \notin GNI$ (каждый граф имеет n вершин) и любой строки w, $\Pr_{\pi_0,\pi_1 \leftarrow S_n,\Pi}[\Pi(\pi_0(G_0), \pi_1(G_1), w) = 1] \leq \dfrac{1}{4}$.

Определение 6.2. Система доказательств Π_1 *моделирует* систему доказательств Π_2 ($\Pi_1 \leq \Pi_2$), если существует такой полином p, что для каждого $x \in L$ выполняется $l_{\Pi_1(x)} \leq p(l_{\Pi_2}(x) + |x|)$, где $l_{\Pi_i(x)}$ - длина кратчайшего Π_i-доказательства для x.

Система доказательств Π_1 *p-моделирует* систему доказательств Π_2 ($\Pi_1 \leq_p \Pi_2$), если существует полиномиальный по времени алгоритм, который по x и Π_2-доказательству для x выводит Π_1-доказательство для x.

Система доказательств Π называется *p-оптимальной*, если она p-моделирует любую другую систему доказательств.

Лемма 6.1. Для любой корректной системы доказательств Π существует система доказательств $\hat{\Pi}$ такая, что $\hat{\Pi} \leq_p \Pi$, и время работы $\hat{\Pi}$ не превышает Cn^2, где n — это длина входа, а C — абсолютная константа.

Доказательство. Пусть время работы алгоритма Π ограничено полиномом p. Определим алгоритм $\hat{\Pi}$: если вход алгоритма имеет вид $x, w01^{p(|x|+|w|)}$, то просто запустить $\Pi(x, w)$, иначе выдать 0. Тогда $\hat{\Pi}$ p-моделирует Π и вычисляется за время не большее, чем Cn^2. □

Пусть Π - вероятностный алгоритм, $w \in \{0,1\}^*$, (G_0, G_1) - пара графов с n вершинами, δ - рациональное число. Если Π - инвариантная система доказательств, время работы которой ограничено квадратом от размера входа, то следующий алгоритм Amplify увеличивает вероятность успеха Π.

Алгоритм 6.1. Amplify$(G_0, G_1, \Pi, w, \delta)$

- Выполнить $m = max\{\lceil 8 \ln \frac{2}{\delta} \rceil, \lceil 32 \ln 2 \rceil\} + 1$ раз:
  - сгенерировать случайные перестановки $\pi_0, \pi_1 \in S_n$
  - запустить $\Pi((\pi_0(G_0), \pi_1(G_1)), w)$ на число шагов, не большее, чем Cn^2, где n — длина входа.
- Выдать 1, если алгоритм Π вернул 1 хотя бы $\frac{m}{2}$ раз.

Лемма 6.2. Пусть Π — инвариантная система доказательств, работающая не более Cn^2 шагов, w — это Π-доказательство для (G_0, G_1).

1. Если $G_0, G_1 \notin GNI$, то $\Pr[Amplify(G_0, G_1, \Pi, w) = 1] < \delta$.
2. Если $G_0, G_1 \in GNI$, то $\Pr[Amplify(G_0, G_1, \Pi, w] = 1] > 1 - \frac{1}{8}$

Доказательство. Следует из оценок Чернова. □

Следующий алгоритм исправляет плохие системы доказательств, чтобы они не доказывали неверные утверждения.

Алгоритм 6.2. SelfCorrect(G_0, G_1, Π, w, k)

- Запустить подряд k раз $\text{Amplify}(G_0, G_0, \Pi, w, \frac{1}{8k})$ и один раз $\text{Amplify}(G_0, G_1, \Pi, w, \frac{1}{8k})$, каждый раз используя независимые от предыдущих случайные биты. Если первые k запусков выдали 0, а последний 1, выдать 1, иначе выдать 0.

Лемма 6.3. 1. Для любого вероятностного алгоритма Π, любой пары графов $G_0, G_1 \notin GNI$ и любых w и k выполняется

$$\Pr[SelfCorrect(G_0, G_1, \Pi, w, k) = 1] \leq \frac{1}{k+1}.$$

2. Если Π — инвариантная система доказательств, время работы которой ограничено Cn^2, где n — длина входа, $G_0, G_1 \in GNI$, и w является Π-доказательством для (G_0, G_1), то $\Pr[SelfCorrect(G_0, G_1, \Pi, w, k) = 0] < \frac{1}{4}$.

Доказательство. 1. Т.к. алгоритм Amplify применяет к входу G_0, G_1 случайные перестановки, и $G_0 \cong G_1$, то каждый из $k+1$ запусков алгоритма Amplify, которые производит SelfCorrect, выдает 1 с одинаковой вероятностью. Следовательно, вероятность того, что SelfCorrect выдает 1 не превосходит $\frac{1}{k+1}$ (поскольку все $k+1$ событий, что на выходе появляется k нулей и одна единица, равновероятны).

2.

$$\Pr[SelfCorrect(G_0, G_1, \Pi, w, k) = 0] = \Pr[Amplify(G_0, G_1, \Pi, w, \frac{1}{8k}) = 0] + \\ k \cdot \Pr[Amplify(G_0, G_0, \Pi, w, \frac{1}{8k}) = 1] < \frac{1}{8} + k \cdot \frac{1}{8k} = \frac{1}{4}$$

□

Обозначим $\Pi^*(G_0, G_1, \Pi, w) = SelfCorrect(G_0, G_1, \Pi, w, 3)$.

Теорема 6.1. Алгоритм Π^* является оптимальной инвариантной системой доказательств для GNI.

Доказательство. Π^*-доказательства - это пары (Π, w), где Π — описание алгоритма, w — Π-доказательство для (G_0, G_1).

Проверим, что Π^* является корректной системой доказательств. Пусть $(G_0, G_1) \in GNI$, Π — корректная инвариантная система доказательств, пусть система доказательств $\hat{\Pi}$ моделирует Π и работает время не большее, чем Cn^2, где n — длина входа (такая система строится в лемме 6.1) и w - корректное $\hat{\Pi}$-доказательство пары (G_0, G_1). Amplify применяет к G_0, G_1 случайные перестановки, поэтому по пункту 2 леммы 6.3 $\Pr_{\pi_0, \pi_1, \Pi^*}[\Pi^*(\pi_0(G_0), \pi_1(G_1), \hat{\Pi}, w) = 1] = \Pr_{\Pi^*}[\Pi^*(G_0, G_1, \hat{\Pi}, w) = 1] \geq \frac{3}{4}$. Значит, если w - корректное $\hat{\Pi}$-доказательство для пары (G_0, G_1), то $(\hat{\Pi}, w)$ — это корректное Π^* для пары (G_0, G_1). Значит мы доказали, что Π^* p-моделирует $\hat{\Pi}$, а следовательно и Π.

Из пункта 1 леммы 6.3 следует, что для пары изоморфных графов G_0, G_1 для любой строки w $\Pr_{\pi_0, \pi_1, \Pi^*}[\Pi^*(\pi_0(G_0), \pi_1(G_1), \Pi, w) = 1] \leq \frac{1}{4}$.

Система Π^* является инвариантной относительно перестановок, так как алгоритм Amplify применяет перестановки ко входу. Поскольку Amplify запускает алгоритмы лишь на квадратичное от длины входа время, то время работы Π^* ограничено полиномом относительно $(|x| + |w| + |\Pi|)$.

□

Список литературы

[FS04] Lance Fortnow and Rahul Santhanam. Hierarchy theorems for probabilistic polynomial time. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, pages 316–324, 2004.

[Gol11] Oded Goldreich. A sample of samplers: A computational perspective on sampling. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, volume 6650 of *Lecture Notes in Computer Science*, pages 302–332, 2011.

[HI12] Edward A. Hirsch and Dmitry Itsykson. On an optimal randomized acceptor for graph nonisomorphism. *Inf. Process. Lett.*, 112(5):166–171, 2012.

[HIMS12] Edward A. Hirsch, Dmitry Itsykson, Ivan Monakhov, and Alexander Smal. On optimal heuristic randomized semidecision procedures, with applications to proof complexity and cryptography. *Theory of Computing Systems*, 2012. Extended abstract appeared in the proceedings of *STACS-2010*. Preliminary version is available as ECCC TR10-193.

[HKN+05] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In *Proc. of EUROCRYPT-2005*, 2005.

[Its10] Dmitry M. Itsykson. Structural complexity of AvgBPP. *Annals of Pure and Applied Logic*, 162(3):213–223, 2010.

[KvM99] Adam R. Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, STOC '99, pages 659–667, New York, NY, USA, 1999. ACM.

[Mes99] Jochen Messner. On optimal algorithms and optimal proof systems. In *Proceedings of the 16th Symposium on Theoretical Aspects of Computer Science*, volume 1563 of *Lecture Notes in Computer Science*, pages 361–372, 1999.

[Sch76] Claus-Peter Schnorr. Optimal algorithms for self-reducible problems. In *ICALP'76*, pages 322–337, 1976.