САНКТ-ПЕТЕРБУРГСКИЙ АКАДЕМИЧЕСКИЙ УНИВЕРСИТЕТ – НАУЧНО-ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР НАНОТЕХНОЛОГИЙ РАН

На правах рукописи

Диссертация допущена к защите

Зав. кафедрой

А.В.Омельченко

_____

“ ”_____2011 г.

Д И С С Е Р Т А Ц И Я

НА СОИСКАНИЕ УЧЕНОЙ СТЕПЕНИ

М А Г И С Т Р А

Тема: Оптимальный детерминированный эвристический алгоритм для образа инъективной функции.

Направление: 010600.68 – Прикладные математика и физика

Магистерская программа: "Математические и информационные технологии"

Выполнила студентка В.О. Николаенко

(подпись)

Руководитель:
к.ф.-м.н. Д.М. Ицыксон

(подпись)

Рецензент:
к.ф.-м.н., доцент Э.А. Гирш

(подпись)

Санкт-Петербург
2011 г.

Реферат

Работа выполнена на 35 страницах, в 4 главах. Использовано 18 источников.

Работа посвящена построению оптимального эвристического алгоритма не использующего случайных битов для проверки принадлежности образу инъективной функции. Эвристический алгоритм может ошибаться на не более, чем $\frac{1}{d}$ доле входов, где параметр d дается алгоритму в качестве дополнительного входа. В работе улучшается результат, полученный в 2010 году Э.А.Гиршем и Д.М.Ицыксоном, о существовании оптимального эвристического вероятностного аксептора (алгоритма, принимающего строки из языка и не останавливающегося на по крайней мере $1 - \frac{1}{d}$ доли других входов, где d дается алгоритму в качестве параметра) для дополнения образа инъективной полиномиально вычислимой функции — дается детерминированная версия и доказывается аналогичное утверждение для образа функции. Такие задачи представляют интерес, поскольку если в качестве функции взять инъективный псевдослучайный генератор, то для дополнения образа генератора не будет существовать полиномиально ограниченного эвристического аксептора. На основе оптимального эвристического алгоритма производится построение оптимальной автоматизируемой эвристической системы доказательств. Для построение оптимального эвристического алгоритма используется конструкция аналогичная Левинскому оптимальному алгоритму для задачи поиска выполняющего набора. Для тестирования алгоритмов в процессе их параллельного исполнения в работе используются графы с хорошими свойствами перемешивания.

Содержание

Введение

Аксептор для языка — это алгоритм, который принимает все строчки из языка и не останавливается на всех остальных строках. Аксепторами обладают все перечислимые языки. Понятие аксептора связано с понятием системы доказательства. *Системой доказательств* [1] для языка L называется полиномиально вычислимая сюръективная функция Π, отображающая множество строчек над некоторым конечным алфавитом на L. Мы говорим, что строка w является доказательством принадлежности строки x языку L, если $\Pi(w) = x$. Неформально на функцию Π можно смотреть как на алгоритм, проверяющий корректность доказательства. Каждый аксептор A для языка L задает систему доказательств для языка L, которая выдает строку x по принимающему протоколу алгоритма A на входе x и выдает какую-нибудь фиксированную строчку языка L, если строка не является принимающим протоколом. С практической точки зрения важным понятием является существование автоматизируемой системы доказательств. Система доказательств Π называется *автоматизируемой*, если существует алгоритм B (автоматизатор), который на строках из языка L выдает Π-доказательство, работая полиномиальное от длины кратчайшего доказательства время, и не останавливается на строках не из языка. Нетрудно видеть, что система доказательств, построенная по аксептору, является автоматизируемой, и каждый автоматизатор легко переделывается в аксептор. Таким образом, понятия автоматизируемой системы доказательств и аксептора эквивалентны со сложностной точки зрения.

Изучение аксепторов и систем доказательств приближает нас к решению важнейших задач теории сложности. $\mathrm{P} = \mathrm{NP}$ в том и только в том случае, когда существует полиномиальный по времени аксептор для языка булевых тавтологий. $\mathrm{NP} = \mathrm{coNP}$ в том и только в том случае, если существует система доказательств для языка булевых тавтологий, в которой все тавтологии имеют доказательства полиномиального размера.

Для многих систем доказательств известны экспоненциальные нижние оценки на длину кратчайших доказательств, для других нижние оценки еще не доказаны. Однако, для доказательства $\mathrm{NP} \neq \mathrm{coNP}$

необходимо и достаточно показать, что системы доказательств для тавтологий с полиномиальными нижними оценками не существует, или, что все системы доказательств для языка тавтологий имеют экспоненциальные нижние оценки. В таких случаях может оказаться полезным найти универсальный объект, в нашем случае это система доказательств с наикратчайшими доказательствами.

Говоря формально, на системах доказательств можно ввести предпорядок (рефлексивное транзитивное отношение) с помощью отношения моделирования (или p-моделирования). Система $\Pi : \Sigma_1 \to L$ моделирует систему $\Phi : \Sigma_2 \to L$, если существуют полином q и функция $h : \Sigma_1 \to \Sigma_2$ такие, что для любой строки $w \in \Sigma_1$ выполняется $\Pi(h(w)) = \Phi(w)$ и $|h(w)| \leq q(|w|)$. Т.е. для любого доказательства в системе Φ существует не слишком более длинное доказательство в системе Π. Если функция h вычислима за полиномиальное время, то говорят, что Π p-моделирует Φ.

Система доказательств Π называется оптимальной (p-оптимальной), если она моделирует (p-моделирует) любую другую. Оптимальная (p-оптимальная) система доказательств является наименьшим элементом относительно только что описанного отношения порядка (моделирования).

Понятие моделирования между системами доказательств схоже с понятием сводимости задач. По этой аналогии оптимальные системы доказательств соответствуют полным задачам.

Оптимальным аксептором для языка L называется такой аксептор, который на всех строках языка работает быстрее, чем любой другой аксептор с точностью до полинома. В 1989 году Крайчек и Пудлак [2] доказали, что существование p-оптимальной системы доказательств эквивалентно существованию оптимального аксептора. В 1999 году Месснер обобщил [3] этот результат на все языки, к элементам которого можно дописывать наполнитель (padding), не изменяя принадлежности языку.

Система доказательств называется полиномиально ограниченной, если для любой строки из языка существует доказательство, длина которого ограничена полиномом от этой строки. Если система Π моделирует систему Φ и система Φ полиномиально ограничена, то и система Π является полиномиально ограниченной.

Если существует оптимальная система доказательств Π, то полиномиально ограниченная система доказательств существует в том

и только в том случае, если Π полиномиально ограничена, а Π полиномиально ограничена в том и только в том случае, если $\mathrm{NP} = \mathrm{co{-}NP}$. Из равенства классов $\mathrm{NP} = \mathrm{co} - \mathrm{NP}$ вытекает существование оптимальной системы для булевых тавтологий, однако, обратное, возможно, не верно.

Обзор литературы. Об оптимальных системах доказательств известна масса условных утверждений. Так, если язык булевых тавтологий и язык выполнимых формул имеют p-оптимальные системы доказательств, то $\mathrm{NP} \cap \mathrm{coNP}$ имеет полное множество относительно m-сведений [4]. Язык A сводится к языку B относительно m-сведений, если существует всюду определенная вычислимая функция f, такая, что $w \in A$ в том и только в том случае, если $f(w) \in B$.

Также известно, что существование p-оптимальной системы доказательств для булевой формулы с кванторами влечет существование полного языка в $\mathrm{NP} \cap \mathrm{coNP}$ [5].

p-оптимальная система доказательств для языка булевых тавтологий существует только в том случае, если существует полная задача в классе UP относительно m-сведений [4]. Класс UP — это класс задач распознавания, лежащих в NP, и решаемых недетерминированными полиномиальными по времени машинами Тьюринга, которые для каждого входа принимают либо строго одну подсказку, либо ни одной.

С другой стороны, существование оптимальной системы доказательств для булевых тавтологий влечет существование полной задачи относительно полиномиальных m-сведений в классе $\mathrm{NP} \cap \mathrm{SPARSE}$ [4]. Класс $SPARSE$ — класс задач распознавания таких, что число принимаемых строк одной длины ограничено полиномом от этой длины.

Крайчек и Пудлак показали, что существование оптимальной ($p-$оптимальной) системы доказательств вытекает из $\mathrm{NExp} = \mathrm{coNExp}$ ($\mathrm{Exp} = \mathrm{NExp}$ соответственно) [2], а Кёблер, Месснер и Торьян [4] улучшили этот результат до двойной экспоненты. Здесь $\mathrm{Exp} = \mathrm{DTIME}(2^{\mathrm{O(n)}})$, $\mathrm{NExp} = \mathrm{NTIME}(2^{\mathrm{O(n)}})$, $\mathrm{ExpExp} = \mathrm{DTIME}(2^{\mathrm{O}(2^{\mathrm{n}})})$, $\mathrm{NExpExp} = \mathrm{NTIME}(2^{\mathrm{O}(2^{\mathrm{n}})})$.

Существование оптимальной системы доказательств, кроме того, имеет несколько интересных следствий и применений, как то: из существования оптимальной системы доказательств следует существование самой трудной для разделения дизъюнктной NP-пары [6, 7]. Дизъюнктная NP-пара — это два непересекающихся множества (U, V), такие что $U, V \in \mathrm{NP}$ и $U \cap V = \emptyset$. Разделить NP-пару, значит представить

такой полиномиальный по времени алгоритм, который возвращает 1, если вход принадлежит U, и 0, если вход принадлежит V, поведение алгоритма на остальных входах может быть произвольным. Существование неразделимой NP-пары является очень важным открытым вопросом в криптографии, поскольку не существует алгоритма шифрования с открытым ключом, если не существует неразделимой дизъюнктной NP-пары [8].

Отметим, что существует оптимальный алгоритм для задачи поиска выполняющего набора для булевой формулы [9]. Этот алгоритм состоит в том, что параллельно запускаются все алгоритмы, и как только какой-то алгоритм выдает подсказку, запускается алгоритм, проверяющий эту подсказку. Отметим, что оптимальный алгоритм Левина не является оптимальным аксептором. Проблема построения как оптимального аксептора, так и оптимальной системы доказательств состоит в том, что мы не знаем, как перечислить все корректные аксепторы или корректные системы доказательств.

Ослабление определений в некоторых случаях приводит к оптимальному аксептору или системе доказательств. Так, в работе Кука и Крайчека [10] строится система доказательств с одним битом подсказки, которая моделирует все системы доказательств с логарифмической длиной подсказки. Система доказательств с подсказкой — это полиномиальный по времени алгоритм $\Pi(x, w, \alpha)$ и последовательность строк длины $t(n)$ такие что для всех x, $x \in L$ в том и только в том случае, когда $\exists w$: $\Pi(x, w, \alpha_{|x|+|w|}) = 1$.

Другой пример ослабления определений представлен в работе [11]. Гирш и Ицыксон рассматривают распределенные задачи доказательства и определяют для них вероятностные эвристические аксепторы и системы доказательств. Распределенная задача доказательства — это язык и полиномиально моделируемое распределение с носителем на дополнении языка. Вероятностные эвристические аксепторы могут использовать случайные числа и, кроме того, могут принять маленькую долю (согласно некоторому полиномиально моделируемому распределению на дополнении языка) строк не из языка. В той же работе показано, что для любого перечислимого языка L и любого полиномиально моделируемого распределения с носителем в дополнении L существует оптимальный вероятностный эвристический аксептор. Оптимальный вероятностный аксептор удается построить за счет того, что можно вероятностно

протестировать, является ли алгоритм корректным аксептором, запустив его на случайных входах согласно распределению на дополнении языка. В работе [12] (которая является расширенной версией работы [11]) показывается, что на основе оптимального эвристического аксептора можно построить оптимальную слабо автоматизируемую эвристическую систему доказательств. Система называется слабо автоматизируемой, если существует алгоритм, работающий полиномиальное от длины кратчайшего доказательства время, и порождающий доказательство в другой системе доказательств. Доказательство эквивалентности существования оптимальных аксепторов и систем доказательств, к сожалению, перестает работать в эвристическом смысле. Существование оптимального аксептора интересно только в том случае, когда для задачи не существует полиномиально ограниченного аксептора, иначе он и является оптимальным. В [12] доказывается, что существование распределенных задач доказательств, для которых не существует полиномиально ограниченных вероятностных аксепторов, эквивалентно существованию бесконечно часто односторонних функций. Бесконечно часто односторонние функции были введены в [13], и предположение об их существовании более слабое, чем существование обычных односторонних функций. Строгое изложение определений и результатов статей [11, 12] дано в разделе 1.1.

Подробнее про оптимальные аксепторы и системы доказательств можно прочитать в обзоре [14].

Постановка задачи. Целью данной работы является дерандомизация вероятностного аксептора, предложенного в статьях [11, 12], для задачи распознавания $(\overline{\mathrm{Im}\, f}, U(\mathrm{Im}\, f))$ (f — инъективная полиномиально вычислимая функция, удлиняющая свой вход на один символ, а $U(S)$ — равномерное распределение на множестве S). Кроме того, ставится задача построения аналогичного детерминированного аксептора для задачи $(\mathrm{Im}\, f, U(\overline{\mathrm{Im}\, f}))$. Также, путем объединения полученных аксепторов, требуется построить алгоритм, распознающий язык $\mathrm{Im}\, f$ с ограниченной ошибкой. Наконец, требуется найти оптимальные автоматизируемые эвристические системы доказательств для задач распознавания $(\overline{\mathrm{Im}\, f}, U(\mathrm{Im}\, f))$ и $(\mathrm{Im}\, f, U(\overline{\mathrm{Im}\, f}))$.

Структура работы. В первой части главы 1 приведены определения, сопутствующие понятию вероятностных эвристических аксепторов, во второй части главы определения переносятся на понятие

детерминированного аксептора. В заключении главы, вводится понятие эвристического алгоритма.

Представленная в работе конструкция оптимального эвристического аксептора является дерандомизацией конструкции оптимального вероятностного эвристического аксептора из работы [11]. С помощью графов с хорошим свойством перемешивания мы заменяем использование случайных чисел на дополнительную небольшую ошибку по входам. Используемые графы и их свойства описаны в главе 2.

В главе 3 приведены конструкции оптимальных эвристических аксепторов для задач $(\operatorname{Im} f, U(\overline{\operatorname{Im} f}))$ и $(\overline{\operatorname{Im} f}, U(\operatorname{Im} f))$, где f — инъективная функция, которая входы длины n переводит во входы длины $n+1$, а $\operatorname{Im} f = f(\{0,1\}^*)$ ($U(S)$ — равномерное распределение на множестве S).

Несмотря на то, что мы существенно сужаем класс рассматриваемых распределенных задач, среди этих задач остаются интересные. В частности, в нашей работе показано, что если f — это инъективный псевдослучайный генератор, то для задачи $(\overline{\operatorname{Im} f}, U(\operatorname{Im} f))$ не существует полиномиально ограниченных аксепторов.

В главе 4 представлены основные следствия существования оптимальных аксепторов. В первой части главы на их основе осуществляется построение оптимального эвристического разрешающего алгоритма для языка $\operatorname{Im} f$, а во второй части вводится понятие эвристической системы доказательств и показывается существование оптимальных автоматизируемых систем доказательств для задач $(\operatorname{Im} f, U(\overline{\operatorname{Im} f}))$ и $(\overline{\operatorname{Im} f}, U(\operatorname{Im} f))$.

1. Основные определения

В первой части этой главы даются определения, связанные с понятием оптимальных вероятностных аксепторов, и описываются результаты, известные для них. Во второй части даются определения детерминированных аксепторов. В последней части определяется понятие эвристического алгоритма.

1.1. Вероятностные эвристические аксепторы

В определениях этого подраздела мы следуем терминологии введенной в статьях Э. Гирша, Д. Ицыксона, И. Монахова, А. Смаля [11, 12].

Определение 1.1. Пара (L, D) называется *распределенной задачей доказательства*, если D — это семейство вероятностных распределений D_n, причем носитель D_n содержится в $\overline{L} \cap \{0,1\}^n$.

Определение 1.2. *Вероятностным эвристическим аксептором* для распределенной задачи доказательств (L, D) называется вероятностный алгоритм A с двумя входами $x \in \{0,1\}^*$ и $d \in \mathbb{N}$, который удовлетворяет следующим свойствам:

1. $A(x, d)$ либо не останавливается, либо возвращает 1.
2. Для любого $x \in L$ и $d \in \mathbb{N}$, $A(x, d) = 1$.
3. Для любых $n, d \in \mathbb{N}$

$$\Pr_{A,\, x \leftarrow D_n}\{A(x, d) = 1\} < \frac{1}{d}.$$

Здесь вероятность берется по x и по случайным битам алгоритма A.

Основной сложностной характеристикой вероятностного эвристического аксептора является медианное время его работы:

Определение 1.3. Медианным временем работы алгоритма A на входе z называется величина

$$t_A(z) = min\{t \mid \Pr_A\{A(z) \text{ останавливается не более, чем за } t \text{ шагов}\} \geq \frac{1}{2}\}.$$

Определение 1.4. Функция $f : \{0,1\}^* \times \mathbb{N} \to \mathbb{N}$ *доминирует* над функцией $g : \{0,1\}^* \times \mathbb{N} \to \mathbb{N}$ на языке L (обозначим $f \succeq g$), если существуют полиномы p, q такие, что для всех $x \in L$ и $d \in \mathbb{N}$

$$g(x,d) \leq \max_{d' \leq q(|x|d)} \{p(f(x,d')d|x|)\}.$$

Определение 1.5. Пусть A и A' — вероятностные эвристические аксепторы для одного и того же языка L. Говорят, что A *моделирует* A', если $t_{A'} \succeq t_A$ на L. Эвристический аксептор называется *оптимальным*, если он моделирует любой другой.

Определение 1.6. Функция f: $\{0,1\}^* \times \mathbb{N} \to \mathbb{N}$ называется полиномиально ограниченной на языке L, если существует полином p такой, что для всех $x \in L$ и $d \in \mathbb{N}$ справедливо $f(x,d) \leq p(|x|d)$. Эвристический аксептор A для распределенной задачи (L,D) называется *полиномиально ограниченным* на L, если функция $t_A(x,d)$ является полиномиально ограниченной на L.

Определение 1.7. Семейство распределений D_n называется *полиномиально моделируемым*, если существует полиномиальный по времени алгоритм такой, что его распределение при равномерном распределении на входах совпадает с D_n.

Теорема 1.1. [11, 12] Для любой распределенной задачи (L,D) с перечислимым языком L и полиномиально моделируемым семейством распределений D существует оптимальный эвристический аксептор.

Определение 1.8. Пусть функция f: $\{0,1\}^* \to \{0,1\}^*$ — это сохраняющая длину полиномиально вычислимая функция. Мы называем f бесконечно часто односторонней, если для любого полиномиального по времени вероятностного алгоритма A и любого полинома p

$$\forall n_0 \; \exists n > n_0 \; \Pr_{x \leftarrow U_n} \{A(x) \in f^{-1}(f(x))\} < \frac{1}{p(n)}, \text{ где } U_n = U(\{0,1\}^n)$$

Теорема 1.2. [12] Бесконечно часто односторонние функции существуют тогда и только тогда, когда существует распределенная задача доказательств, для которой не существует полиномиально ограниченного вероятностного эвристического аксептора.

1.2. Детерминированные эвристические аксепторы

Определения, приведенные ниже являются естественным переносом определений предыдущего подраздела на детерминированные асепторы.

Определение 1.9. *Детерминированным эвристическим аксептором* для распределенной задачи доказательства $(L,\ D)$ называется детерминированный алгоритм A с двумя входами $x \in \{0,1\}^*$ и $d \in \mathbb{N}$, который удовлетворяет следующим условиям:

1. Для всех $x \in \{0,1\}^*$ и $d \in \mathbb{N}$ алгоритм $A(x,d)$ либо не останавливается, либо возвращает 1.
2. Для любого $x \in L$ и $d \in \mathbb{N}$, $A(x,d) = 1$.
3. Для любых $n, d \in \mathbb{N}$

$$\Pr_{r \leftarrow D_n}\{A(r,d) = 1\} < \frac{1}{d}.$$

Определение 1.10. Функция $f\colon \{0,1\}^* \times \mathbb{N} \to \mathbb{N}$ называется *почти везде полиномиально ограниченной*, если существуют такой полином $p(x)$, что для любого $d \in \mathbb{N}$, для любой длины входа $n \in \mathbb{N}$ выполняется неравенство

$$\Pr_{x \leftarrow U_n}\{f(x,d) \le p(n \cdot d)\} \ge 1 - \frac{1}{d}.$$

Определение 1.11. Функция $f\colon \{0,1\}^* \times \mathbb{N} \to \mathbb{N}$ *почти везде доминирует* над функцией $g\colon \{0,1\}^* \times \mathbb{N} \to \mathbb{N}$ (будем обозначать $f \succsim g$), если существуют два полинома p и q, таких, что $q(n,d) \ge 2d$ и для всех $d \in \mathbb{N}$, для любой длины входа $n \in \mathbb{N}$ выполняется неравенство:

$$\Pr_{x \leftarrow U_n}\{g(x,d) \le p(n \cdot d \cdot f(x, q(n \cdot d)))\} \ge 1 - \frac{1}{2d}.$$

Покажем, что почти везде полиномиально ограниченные функции замкнуты относительно введенного отношения доминирования.

Теорема 1.3. Пусть $f \succsim g$ и $f(x,d)$ является почти везде полиномиально ограниченной, тогда $g(x,d)$ также является почти везде полиномиально ограниченной.

Доказательство. Пусть p и q — полиномы из определения доминирования, а полином p' — из определения полиномиальной ограниченности функции

$f(x,d)$. Не умаляя общности можно считать, что полином p возрастает. Из условия моделирования получаем

$$\Pr_{x\leftarrow U_n}\{g(x,d)\le p(n\cdot d\cdot f(x,q(nd)))\}\ge 1-\frac{1}{2d}.$$

Благодаря почти везде полиномиальной ограниченности функции f:

$$\Pr_{x\leftarrow U_n}\{f(x,q(nd))\le p'(n\cdot q(nd))\}\ge 1-\frac{1}{q(nd)}\ge 1-\frac{1}{2d}.$$

Комбинируя два последних неравенства и используя тот факт, что полином p возрастает, получаем:

$$\Pr_{x\leftarrow U_n}\{g(x,d)\le p(n\cdot d\cdot p'(n\cdot q(nd)))\}\ge 1-\frac{1}{2d}-\frac{1}{2d}\ge 1-\frac{1}{d}.$$

Последнее неравенство говорит о почти везде полиномиальной ограниченности функции g. □

Определение 1.12. Для эвристических детерминированных аксепторов A и A' для одной и той же распределенной задачи доказательства (L,D) будем говорить, что A *моделирует* A', если $t_{A'}\gtrsim t_A$ на L. Здесь $t_A(z,d)$ — количество шагов, за которое A принимает (z,d), и ∞, если A на входе (z,d) не останавливается.

Определение 1.13. Детерминированный эвристический аксептор называется *оптимальным*, если он моделирует любой другой.

В этой работе мы будем рассматривать распределения специального вида. В качестве языков мы ограничимся рассмотрением образов и прообразов инъективной функции $f:\{0,1\}^n\to\{0,1\}^{n+1}$:

$$\operatorname{Im} f = f(\{0,1\}^*),$$
$$\overline{\operatorname{Im} f} = \{0,1\}^*\setminus \operatorname{Im} f.$$

Распределение на дополнении языка будем брать равномерным. Как показывает следующая теорема, эти классы представляют интерес для исследования.

Определение 1.14. Пусть $g:\{0,1\}^*\to\{0,1\}^*$ — полиномиально вычислимая функция такая, что $|g(r)|=|r|+1$ при всех $r\in\{0,1\}^*$.

Функция g называется псевдослучайным генератором, если для любого полиномиального вероятностного алгоритма A и любого полинома p,

$$\exists n_0 \forall n > n_0 | \Pr_{x \leftarrow U_n} \{A(g(x)) = 1\} - \Pr_{x \leftarrow U_{n+1}} \{A(x) = 1\}| < \frac{1}{p(n)}.$$

Инъективные псевдослучайные генераторы существуют, если существуют односторонние перестановки [15].

Теорема 1.4. Если g - инъективный псевдослучайный генератор, то для задачи $(\overline{\mathrm{Im}\, g}, U(\mathrm{Im}\, g))$, где $U(\mathrm{Im}\, g)$ — равномерное распределение на $\mathrm{Im}\, g$, не существует почти везде полиномиально ограниченного детерминированного эвристического аксептора.

Доказательство. Пусть существует почти везде полиномиально ограниченный эвристический аксептор A для $(\overline{\mathrm{Im}\, g}, U(\mathrm{Im}\, g))$. Это означает, что существует полином p такой, что

$$\Pr_{x \in U_{n+1}(\overline{\mathrm{Im}\, g})} \{t_A(x, d) \leq p(|x|d)\} \geq 1 - \frac{1}{d}.$$

Пусть $B(x)$ запускает $A(x, \frac{1}{5})$ на $p(|x|5)$ шагов и возвращает 1, если A принимает, и 0 иначе. Из того, что A — аксептор, следует, что

$$\Pr_{x \leftarrow U_n} \{B(g(x)) = 1\} < \frac{1}{5}. \tag{1.1}$$

Из того, что A почти везде полиномиально ограничен, следует

$$\Pr_{x \leftarrow U_{n+1}} \{B(x) = 1\} \geq \frac{1}{2} \Pr_{x \leftarrow U_{n+1}(\overline{\mathrm{Im}\, g})} \{B(x) = 1\} \geq \frac{1}{2}(1 - \frac{1}{5}) \geq \frac{2}{5}. \tag{1.2}$$

Неравенства (1.1) и (1.2) дают противоречие с тем, что g — псевдослучайный генератор. □

1.3. Детерминированные эвристические алгоритмы

Определение 1.15. Алгоритм $A(x, d)$ называется *эвристическим детерминированным разрешающим алгоритмом* для распределенной задачи (L, D), если $A(x, d)$ останавливается на всех входах с ответом либо 0, либо 1, и $\Pr_{x \leftarrow D_n}\{A(x, d) \neq L(x)\} < \frac{1}{d}$, где $L(x)$ — характеристическая функция языка L, т.е. $L(x) = 1$ в том и только в том случае, если $x \in L$, иначе $L(x) = 0$. Здесь распределение D_n может быть сосредоточено на любом подмножестве $\{0, 1\}^n$.

Определение 1.16. Эвристический детерминированный разрешающий алгоритм $A(x, d)$ для задачи (L, D) называется *оптимальным*, если для любого другого эвристического детерминированного разрешающего алгоритма A', $t_{A'} \gtrsim t_A$ на $\{0,1\}^n$ для любого натурального n.

Определение 1.17. Эвристическим детерминированным алгоритмом $A(x, d)$ для языка L будем называть эвристический детерминированный алгоритм для распределенной задачи $(L, U(\overline{L}))$.

2. Перемешивающий граф

Теорема 2.1. Для любого $n \in \mathbb{N}$, для любого $\delta \geq 2^{-n}$, существует двудольный граф G_n с кратными ребрами такой, что количество вершин в левой и правой долях равно 2^n, и он удовлетворяет следующим условиям:

1. Существует полином $poly(n)$ такой, что все вершины графа G имеют одинаковую степень $s = poly(\frac{1}{\delta})$, и i-й сосед вершины u может быть найден за время, полиномиальное от $\frac{1}{\delta}$.
2. Для любого множества вершин правой доли $B \subseteq R$ множество вершин левой доли $A = \{u \in L : \left|\frac{|E(\{u\},B)|}{s} - \rho(B)\right| > \delta\}$ имеет плотность $\rho(A) < \delta$.

Здесь $E(S_1, S_2)$ обозначает множество ребер, ведущих из множества S_1 в множество S_2, $\rho(S) = \frac{|S|}{2^n}$ обозначает плотность множества S в соответствующей доли.

Доказательство. Мы будем использовать графы Рамануджана R_n степени $s = poly(\frac{1}{\delta})$ со вторым собственным числом λ, множеством вершин $\{0,1\}^n$ и такие, что $\frac{\lambda}{s} \leq \epsilon$, где $\epsilon = \frac{\delta^2}{2}$ [16],[17]. Графы R_n являются эффективно вычислимыми при $\delta \geq 2^{-n}$, т.е. для любой вершины можно получить множество смежных с ней вершин за полиномиальное от $\frac{1}{\delta}$ время.

Здесь также можно использовать конструкцию О. Голдрейха, А. Вигдерсона [18], но ее авторы используют те же графы Рамануджана, которые будут использоваться здесь.

Превратим граф R_n в двудольный, продублировав каждую вершину, т.е. заменив v на v_1 и v_2 и аналогичным образом продублировав ребра, превратив (u, v) в (u_1, v_2) и (u_2, v_1), полученный граф обозначим G_n. Покажем, что граф G_n удовлетворяет свойствам 1, 2 теоремы.

Свойство 1 выполнено благодаря эффективному построению графа.

Поскольку граф G_n является регулярным графом степени s со вторым собственным числом λ, для него верна лемма о перемешивании:

$$\left|\frac{|E(A,B)|}{|E|} - \rho(A)\rho(B)\right| \leq \frac{\lambda}{s}\sqrt{\rho(A)\rho(B)} < \epsilon.$$

Доказательство этой леммы можно найти, например, в статье [18].

Здесь $|E|$ - это количество ребер графа G_n и $|E| = s \cdot 2^n$.

Для доказательства пункта 2 введем обозначение:

$$C_+ = \left\{ x \in \{0,1\}^n \mid \frac{|E(\{x\}, B)|}{s} > \rho(B) + \delta \right\}.$$

По лемме о перемешивании: $\frac{|E(C_+,B)|}{|E|} - \rho(C_+)\rho(B) < \epsilon$, следовательно $\rho(C_+)\rho(B) > \frac{|E(C_+,B)|}{|E|} - \epsilon \geq \rho(C_+) \cdot (\rho(B) + \delta) - \epsilon$, значит $\rho(C_+) < \frac{\epsilon}{\delta}$.

Аналогичное рассуждение можно провести для

$$C_- = \left\{ x \in \{0,1\}^n | \frac{|E(\{x\}, B)|}{s} < \rho(B) - \delta \right\}.$$

По лемме о перемешивании: $\frac{|E(C_-,B)|}{|E|} - \rho(C_-)\rho(B) > -\epsilon$, следовательно $\rho(C_-)\rho(B) < \frac{|E(C_-,B)|}{|E|} + \epsilon \leq \rho(C_-) \cdot (\rho(B) - \delta) + \epsilon$, значит $\rho(C_-) < \frac{\epsilon}{\delta}$.

Значит $\rho(C_+ \cup C_-) < 2\frac{\delta^2}{2\delta} = \delta$, что и требовалось. □

Следствие 2.1. Рассмотрим граф G_n из теоремы 2.1. Для любого множества вершин правой доли $B \subseteq R$ такого, что $\rho(B) > 3\delta$, множество вершин левой доли $A = \{u \in L : |E(\{u\}, B)| < 2\delta s\}$ имеет плотность $\rho(A) < \delta$.

Доказательство. Множество вершин A можно эквивалентно записать таким образом:

$$A = \{u \in L : \frac{|E(\{u\}, B)|}{s} - \rho(B) < 2\delta - \rho(B)\}.$$

Тогда множество

$$A' = \{u \in L : \frac{|E(\{u\}, B)|}{s} - \rho(B) < -\delta\}$$

является надмножеством множества A, поскольку $\rho(B) > 3\delta$. В силу пункта 2 теоремы 2.1, плотность множества A' меньше δ, значит и плотность множества A меньше δ. □

Следствие 2.2. Рассмотрим граф G_n из теоремы 2.1. Для любого множества вершин правой доли $B \subseteq R$, такого что $\rho(B) < \delta$, множество вершин левой доли $A = \{u \in L : |E(\{u\}, B)| > 2\delta s\}$ имеет плотность $\rho(A) < \delta$.

Доказательство. Множество вершин A можно эквивалентно записать таким образом:

$$A = \{u \in L : \frac{|E(\{u\}, B)|}{s} - \rho(B) > 2\delta - \rho(B)\}.$$

Тогда множество вершин

$$A' = \{u \in L : \frac{|E(\{u\}, B)|}{s} - \rho(B) > \delta\}$$

является надмножеством множества A, поскольку $\rho(B) < \delta$. В силу пункта 2 теоремы 2.1, плотность множества A' меньше δ, значит и плотность множества A меньше δ. □

3. Оптимальные аксепторы

3.1. Оптимальный эвристический аксептор для $(\overline{\mathrm{Im}\, f}, U(\mathrm{Im}\, f))$

В этой главе мы приведем конструкцию оптимального детерминированного аксептора для задачи $(\overline{\mathrm{Im}\, f}, U(\mathrm{Im}\, f))$ — дополнения множества значений инъективной полиномиально вычислимой функции, f, удлиняющей свой вход на один символ.

Напомним, что аксептором распределенной задачи (L, D) называется детерминированный алгоритм A, который либо говорит 1, либо не останавливается, и такой, что вероятность его ошибки согласно распределению D мала, т.е. для всех $d \in \mathbb{N}$ и для всех $n \in \mathbb{N}$:

$$\Pr_{x \leftarrow D_n} \{A(x, d) = 1\} < \frac{1}{d}.$$

Данная конструкция является дерандомизацией конструкции вероятностного эвристического аксептора, предложенного в работах [11, 12]. Заметим, что распределение $U(\mathrm{Im}\, f)$ является полиномиально моделируемым с сэмплером f.

Оптимальный аксептор будет работать по следующей схеме: он будет параллельно запускать некоторое количество машин Тьюринга — потенциальных аксепторов. Это количество будет явным образом зависеть от его параметров. Если какая-то машина остановится с ответом 1, алгоритм запустит проверку этой машины. Алгоритм проверки принимает машину, если ее поведение похоже на поведение аксептора. Если алгоритм проверки принял машину, мы остановимся с ответом 1, иначе продолжим работу.

3.1.1. Конструкция оптимального аксептора

Пусть $f : \{0,1\}^* \to \{0,1\}^*$ — инъективная полиномиально вычислимая функция такая, что $|f(r)| = |r| + 1$ для всех $r \in \{0,1\}^*$.

Пусть A_i — это нумерация всех алгоритмов, от двух параметров.

Пусть $\overline{\mathcal{L}}(x)$ — экспоненциальный алгоритм, распознающий язык $\overline{\mathrm{Im}\, f}$.

Алгоритм 3.1 (Алгоритм $S(x,d)$)**.**

1. Пусть $n = |x|$. Если $\frac{1}{8dn^2} < 2^{-n}$, запустим $\overline{\mathcal{L}}(x)$ и вернем ответ этой процедуры.
2. Иначе, будем запускать параллельно все алгоритмы: $A_1(x, 8dn^2), A_2(x, 8dn^2), ..., A_{\lfloor \frac{n}{7} \rfloor}(x, 8dn^2)$ и алгоритм $\overline{\mathcal{L}}(x)$.
3. Если какой-то алгоритм $A_i(x, 8dn^2)$ остановился с ответом 1 за T_i шагов, запустим процедуру $Verify(A_i, 2^{\lceil log(T_i) \rceil}, x, \frac{1}{8dn^2}, 8dn^2)$. Если процедура $Verify$ ответила 1, остановимся с ответом 1. Иначе, продолжим.
4. Если же $\overline{\mathcal{L}}$ выдает 1, то остановиться и выдать 1.

Процедура $Verify(A, T, x, \delta, 8dn^2)$ будет проверять, похож ли алгоритм на корректный аксептор для задачи $(\overline{\mathrm{Im}\, f}, U(\mathrm{Im}\, f))$.

Алгоритм 3.2 (Алгоритм $Verify(A, T, x, \delta, d')$)**.**

1. Пусть $n = |x|$, возьмем граф G_n из теоремы 2.1[1], соответствующий числам n и $\delta = \frac{1}{8dn^2}$. Пусть s - степень вершин левой доли.
2. Пусть $y|_m$ обозначает строку, состоящую из первых m битов строки y. Пусть E_n — множество ребер графа G_n.
3. Положим
$$\Delta = \frac{\sum_{(x,z) \in E_n} A^{\leq T}(f(z|_{n-1}), d')}{s}.$$
Если $\Delta \leq 2\delta$, вернуть 1, иначе 0.

3.1.2. Доказательство корректности

Теорема 3.1. $S(x,d)$ является корректным детерминированным эвристическим аксептором для задачи доказательства $(\overline{\mathrm{Im}\, f}, U(\mathrm{Im}\, f))$.

Доказательство. Пусть A_i — некоторый алгоритм из нашей нумерации. Для упрощения записи введем ряд обозначений:

- $S_i(x,T) = 1$ в том и только в том случае, если $A_i(x, 8dn^2)$ останавливается с ответом 1 ровно за T шагов, иначе положим $S_i(x,T) = 0$;

[1] Заметим, что алгоритм S вызывает процедуру $Verify$ в том и только в том случае, если $\delta \geq 2^{-n}$. Последнее необходимо для существования графа G_n.

- $C_i(x,T) = 1$ в том и только в том случае, если $Verify(A,T,x,\delta,8dn^2) = 1$, иначе положим $C_i(x,T) = 0$.

Оценим долю входов не из языка, на которых $S(x,d)$ останавливается с неверным ответом:

$$\sum_{i=1}^{\lfloor \frac{n}{7} \rfloor} \Pr_{x \leftarrow U(\operatorname{Im} f)} \{\exists T : S_i(x,T) = 1 \wedge C_i(x,T) = 1\} \leq \tag{3.1}$$

$$\sum_{i=1}^{\lfloor \frac{n}{7} \rfloor} \sum_{k=1}^{n} \Pr_{x \leftarrow U(\operatorname{Im} f)} \{S_i(x,2^k) \wedge C_i(x,2^k)\} \leq \tag{3.2}$$

$$\sum_{i=1}^{\lfloor \frac{n}{7} \rfloor} n \cdot \Pr_{x \leftarrow U(\operatorname{Im} f)} \{S_i(x,\widetilde{T}) \wedge C_i(x,\widetilde{T})\} \leq \tag{3.3}$$

$$\sum_{i=1}^{\lfloor \frac{n}{7} \rfloor} n \cdot min\{\Pr_{x \leftarrow U(\operatorname{Im} f)} \{S_i(x,\widetilde{T})\}, \Pr_{x \leftarrow U(\operatorname{Im} f)} \{C_i(x,\widetilde{T})\}\}. \tag{3.4}$$

Здесь $\widetilde{T}$ — время, для которого вероятность под суммой 3.2 максимальна.

Если $\Pr_{x \leftarrow U(\operatorname{Im} f)}\{S_i(x,\widetilde{T})\} > 3\delta$, то плотность множества $B = \{x \in R \mid S_i(f(x|_{n-1}),\widetilde{T}) = 1\}$ правой доли графа G_n, $\rho(B) > 3\delta$. Тогда по следствию 2.1 теоремы 2.1 для графа G_n, множество $A = \{u \in L : |E(\{u\},B)| > 2\delta s\}$ имеет плотность $\rho(A) < \delta$, тогда $\Pr_{x \leftarrow U(\operatorname{Im} f)}\{C_i(T)\} = 2 \cdot \rho(A) < 2\delta$. Таким образом, неравенство 3.4 можно продолжить оценкой $\frac{n}{7} \cdot n \cdot 3\delta = \frac{3}{7}n^2 \cdot \frac{1}{8dn^2} < \frac{1}{d}$.

Заметим также, что $S(x,d)$ будет всегда возвращать 1 на словах из языка $\overline{\operatorname{Im} f}$ благодаря алгоритму $\overline{\mathcal{L}}$, распознающему язык $\overline{\operatorname{Im} f}$. □

3.1.3. Доказательство оптимальности

Теорема 3.2. $S(x,d)$ является оптимальным детерминированным эвристическим аксептором для $(\overline{\operatorname{Im} f}, U(\operatorname{Im} f))$.

Доказательство. Если $\frac{1}{8dn^2} < 2^{-n}$, тогда время работы алгоритма $S(x,d)$ это $poly(|x|d)$ для $x \in \overline{\operatorname{Im} f}$ и условие оптимальности в этом случае выполнено; поэтому рассмотрим случай, когда $\frac{1}{8dn^2} \geq 2^{-n}$.

Пусть A_i — произвольный корректный эвристический детерминированный аксептор. Покажем, что алгоритм S будет моделировать A_i.

Выберем n такое, что $n > 7i$. В этом случае алгоритм $S(x, d)$ будет запускать алгоритм A_i.

Согласно определению корректного детерминированного эвристического аксептора:

$$\Pr_{x \leftarrow U_n(\operatorname{Im} f)}\{A_i(x, 8dn^2) = 1\} < \frac{1}{8dn^2} = \delta.$$

Тогда плотность множества вершин правой доли $B = \{x \in R \mid A_i(f(x|_{n-1}), 8dn^2) = 1\}$ равна $\frac{|B|}{2^n} < \delta$.

Согласно следствию 2.2 для графа G_n множество вершин левой доли $A = \{v \in L : |E(\{v\}, B)| > 2\delta s\}$ имеет плотность $\frac{|A|}{2^n} < \delta$.

Поэтому для любого T алгоритм $Verify(A_i, T, x, \delta, 8dn^2)$ принимает по крайней мере $1 - \delta$ долю всех входов и по крайней мере $1 - 2\delta > 1 - \frac{1}{2d}$ долю входов из $\overline{\operatorname{Im} f} \cap \{0,1\}^n$. Поэтому

$$\Pr_{x \leftarrow U_n(\overline{\operatorname{Im} f})}\{t_U(x, d) \le p(nd \cdot t_{A_i}(x, 8dn^2))\} \ge 1 - \frac{1}{2d},$$

где p — это некоторый фиксированный полином, который ограничивает сверху время, затрачиваемое на моделирование алгоритма A_i и на моделирование параллельного запуска алгоритмов. □

3.2. Оптимальный эвристический аксептор для $(\operatorname{Im} f, U(\overline{\operatorname{Im} f}))$

В этой главе мы приведем конструкцию оптимального детерминированного аксептора для задачи распознавания $(\operatorname{Im} f, U(\overline{\operatorname{Im} f}))$, где f — инъективная полиномиально вычислимая функция, удлиняющая свой вход на один символ. Пусть $\mathcal{L}(x)$ — это экспоненциальный алгоритм, распознающий язык $\operatorname{Im} f$.

Основная сложность построения аксептора такого вида, в отличие от аксептора для задачи $(\overline{\operatorname{Im} f}, U(\operatorname{Im} f))$ состоит в том, что распределение на дополнении языка $U(\overline{\operatorname{Im} f})$ не является полиномиально моделируемым, а потому для проверки аксептора на корректность у нас не получится так же просто, как и в предыдущем случае, сэмплировать входы.

В следующем пункте мы приведем формальную конструкцию оптимального аксептора, затем конструкцию проверяющего алгоритма и докажем одну вспомогательную лемму, которая понадобится нам в дальнейшем.

3.2.1. Конструкция оптимального аксептора

Алгоритм 3.3 (Оптимальный аксептор $R(x,d)$)**.**

1. Пусть $n = |x|$. Если $\frac{1}{8dn^2} < 2^{-n}$, запустим $\mathcal{L}(x)$ и вернем ответ этой процедуры, иначе перейдем к шагу 2.
2. Пусть A_i это нумерация всех детерминированных алгоритмов, от двух параметров.
3. Будем запускать параллельно все алгоритмы: $A_1(x, 8dn^2), A_2(x, 8dn^2), ..., A_{\lfloor \frac{n}{7} \rfloor}(x, 8dn^2)$ и алгоритм $\mathcal{L}(x)$.
4. Если какой-то алгоритм $A_i(x, 8dn^2)$ остановился с ответом 1 за T_i шагов, запустим процедуру $Verify(A_i, 2^{\lceil log(T_i) \rceil}, \frac{1}{8dn^2}, 8dn^2)$. Если процедура $Verify$ ответила 1, остановимся с ответом 1. Иначе, продолжим.
5. Если же $\mathcal{L}$ остановилась с ответом 1, остановимся и выдадим 1.

Процедура $Verify(A, T, x, \delta, 8dn^2)$ будет проверять похож ли алгоритм на аксептор для задачи $(\operatorname{Im} f, U(\overline{\operatorname{Im} f}))$.

Алгоритм 3.4 (Алгоритм $Verify(A_i, T, x, \delta, d)$)**.**

Введем обозначение $S_i(y, T) = 1$ в том и только в том случае, когда $A_i(y, d)$ останавливается и выдает 1 не более, чем за T шагов, иначе $S_i(y, T) = 0$.

1. Возьмем граф G_n с параметром $\delta = \frac{1}{8dn^2}$ из теоремы 2.1[2], пусть E_n — множество ребер G_n.
2. Положим

$$H_1(x) = \frac{\sum_{(x,z) \in E_n} S_i(z, T)}{s} \text{ и}$$

$$H_2(x) = \frac{\sum_{(x,z) \in E_n} S_i(f(z|_{n-1}), T)}{s},$$

[2]Заметим, что алгоритм R вызывает процедуру $Verify$ в том и только в том случае, если $\delta \geq 2^{-n}$. Последнее необходимо для существования графа G_n.

где $y|_m$ обозначает строку состоящую из первых m битов строки y. Заметим, что величины $H_1(x)$ и $H_2(x)$ зависят не только от x, но и от всех параметров алгоритма $Verify$, однако для упрощения изложения мы будем опускать эти параметры там, где они будут ясны из контекста.

3. Если $2H_1(x) - H_2(x) < 4\delta$, принять, иначе — отвергнуть.

Докажем лемму, которая понадобится нам в дальнейшем для доказательства свойств полученного алгоритма:

Лемма 3.1.

Если $\Pr\limits_{x \leftarrow U_n(\overline{\mathrm{Im}\, f})}\{S_i(x,T)=1\} > 7\delta$, то

$$\Pr_{x \leftarrow U_n(\overline{\mathrm{Im}\, f})}\{Verify(A_i,T,x,\delta,8dn^2)=1\} < 4\delta.$$

Если $\Pr\limits_{x \leftarrow U_n(\overline{\mathrm{Im}\, f})}\{S_i(x,T)=1\} < \delta$, то

$$\Pr_{x \leftarrow U_n(\mathrm{Im}\, f)}\{Verify(A_i,T,x,\delta,8dn^2)=1\} > 1 - 4\delta.$$

Доказательство. Благодаря тому, что мощность языка $\mathrm{Im}\, f$ совпадает с мощностью его дополнения, для любой длины входа имеет место следующее равенство:

$$\Pr_{x \leftarrow U_n(\overline{\mathrm{Im}\, f})}\{S_i(x,T)=1\} =$$
$$2\Pr_{x \leftarrow U_n}\{S_i(x,T)=1\} - \Pr_{x \leftarrow U_n(\mathrm{Im}\, f)}\{S_i(x,T)=1\}.$$

Обозначим $\Pr_{x \leftarrow U_n}\{S_i(x,T)=1\} = V_1$, $\Pr_{x \leftarrow U_n(\mathrm{Im}\, f)}\{S_i(x,T)=1\} = V_2$. Заметим, что если граф G_n был бы случайным, то имели мы место равенства $E[H_1(x)] = V_1$ и $E[H_2(x)] = V_2$.

Воспользовавшись условием теоремы $2V_1 - V_2 > 7\delta$, оценим нужную нам величину:

$$\Pr_{x \leftarrow U_n(\overline{\mathrm{Im}\, f})}\{Verify(A_i,T,x,8dn^2)=1\} =$$
$$\Pr_{x \leftarrow U_n(\overline{\mathrm{Im}\, f})}\{2H_1(x) - H_2(x) < 4\delta\} <$$
$$\Pr_{x \leftarrow U_n(\overline{\mathrm{Im}\, f})}\{2H_1(x) - H_2(x) - 2V_1 + V_2 < 4\delta - 7\delta\} <$$
$$\Pr_{x \leftarrow U_n(\overline{\mathrm{Im}\, f})}\{2|H_1(x) - V_1| + |H_2(x) - V_2| > 3\delta\}.$$

Воспользуемся свойством 2 графа G_n, показанном в теореме 2.1. Это свойство гласит, что для любого множества вершин правой доли $B \subseteq R$ множество вершин левой доли $A = \left\{u \in L : \left|\frac{|E(u,B)|}{s} - \rho(B)\right| > \delta\right\}$ имеет плотность $\rho(A) < \delta$ в левой доле графа и плотность, меньшую 2δ в подмножестве $\operatorname{Im} f$ левой доли. Воспользуемся этим свойством для множеств правой доли $B_1 = \{x \in \{0,1\}^n \mid S_i(x,T) = 1\}$ и $B_2 = \{x \in \{0,1\}^n \mid S_i\left(f(x|_{n-1}), T\right) = 1\}$, тогда $\rho(B_i) = V_i$, $\frac{|E(x,B_i)|}{s} = H_i(x)$, $i = 1, 2$ и справедливы неравенства:

$$\Pr_{x \leftarrow U_n(\overline{\operatorname{Im} f})}\{2|H_1(x) - V_1| < 2\delta\} > 1 - 2\delta.$$

$$\Pr_{x \leftarrow U_n(\overline{\operatorname{Im} f})}\{|H_2(x) - V_2| < \delta\} > 1 - 2\delta.$$

Объединяя эти два неравенства, получаем:

$$\Pr_{x \leftarrow U_n(\overline{\operatorname{Im} f})}\{2|H_1(x) - V_1| + |H_2(x) - V_2| < 3\delta\} > 1 - 4\delta.$$

Для второго условия теоремы доказательство аналогично, а именно:

$$\Pr_{x \leftarrow U_n(\operatorname{Im} f)}\{Verify(A_i, T, x, 8dn^2) = 1\} =$$
$$1 - \Pr_{x \leftarrow U_n(\operatorname{Im} f)}\{2H_1(x) - H_2(x) > \delta\} >$$
$$1 - \Pr_{x \leftarrow U_n(\operatorname{Im} f)}\{2H_1(x) - H_2(x) - 2V_1 + V_2 > 4\delta - \delta\} >$$
$$1 - \Pr_{x \leftarrow U_n(\operatorname{Im} f)}\{2|H_1(x) - V_1| + |H_2(x) - V_2| > 3\delta\} > 1 - 4\delta.$$

□

3.2.2. Доказательство корректности

Теорема 3.3. $R(x,d)$ является корректным детерминированным эвристическим аксептором для задачи доказательства $(\operatorname{Im} f, U(\overline{\operatorname{Im} f}))$.

Доказательство. Построенный алгоритм либо останавливается с ответом 1, либо не заканчивает работу. Благодаря параллельному запуску экспоненциального разрешающего алгоритма, аксептор всегда принимает входы из $\operatorname{Im} f$. Таким образом, для доказательства корректности необходимо только доказать, что доля ошибок на входах из $\overline{\operatorname{Im} f}$ ограничена константой $\frac{1}{d}$. Дальнейшие рассуждения практически

полностью повторяют рассуждения предыдущего раздела с поправкой на константы.

Для упрощения записи, введем ряд обозначение:

- $C_i(x,T) = 1$ в том и только в том случае, если $Verify(A,T,x,\delta,8dn^2) = 1$, иначе положим $C_i(x,T) = 0$.

Оценим долю входов не из языка, на которых $R(x,d)$ останавливается с неверным ответом:

$$\sum_{i=1}^{\lfloor \frac{n}{7} \rfloor} \Pr_{x \leftarrow U(\overline{\mathrm{Im}\, f})} \{\exists T : S_i(x,T) = 1 \wedge C_i(x,T) = 1\} \leq \tag{3.5}$$

$$\sum_{i=1}^{\lfloor \frac{n}{7} \rfloor} \sum_{k=1}^{n} \Pr_{x \leftarrow U(\overline{\mathrm{Im}\, f})} \{S_i(x,2^k) \wedge C_i(x,2^k)\} \leq \tag{3.6}$$

$$\sum_{i=1}^{\lfloor \frac{n}{7} \rfloor} n \cdot \Pr_{x \leftarrow U(\overline{\mathrm{Im}\, f})} \{S_i(x,\widetilde{T}) \wedge C_i(x,\widetilde{T})\} \leq \tag{3.7}$$

$$\sum_{i=1}^{\lfloor \frac{n}{7} \rfloor} n \cdot min\{\Pr_{x \leftarrow U(\overline{\mathrm{Im}\, f})} \{S_i(x,\widetilde{T})\}, \Pr_{x \leftarrow U(\overline{\mathrm{Im}\, f})} \{C_i(x,\widetilde{T})\}\}. \tag{3.8}$$

Здесь $\widetilde{T}$ — время, для которого вероятность под суммой 3.6 максимальна.

Поскольку если $\Pr_{x \leftarrow U(\overline{\mathrm{Im}\, f})}\{S_i(x,\widetilde{T}) = 1\} > 7\delta$, то по лемме 3.1 $\Pr_{x \leftarrow U(\overline{\mathrm{Im}\, f})}\{C_i(\widetilde{T}) = 1\} < 4\delta$, серию неравенств можно продолжить оценкой $7\delta \cdot \frac{n}{7} \cdot n = \frac{1}{8d} < \frac{1}{d}$, что завершит доказательство корректности ($\delta = \frac{1}{8dn^2}$). □

3.2.3. Доказательство оптимальности

Теорема 3.4. $R(x,d)$ является оптимальным детерминированным эвристическим аксептором для задачи $(\mathrm{Im}\, f, U(\overline{\mathrm{Im}\, f}))$.

Доказательство. Если $\delta < 2^{-n}$, то время работы алгоритма $R(x,d)$ составляет $poly(|x|d)$ для $x \in \mathrm{Im}\, f$ и условие оптимальности в этом случае выполнено; поэтому рассмотрим случай, когда $\delta \geq 2^{-n}$. Пусть A_i произвольный корректный эвристический детерминированный аксептор. Покажем, что алгоритм R будет моделировать A_i.

Выберем n такое, что $\lfloor \frac{n}{7} \rfloor > i$. В этом случае алгоритм $R(x,d)$ будет запускать алгоритм A_i.

Согласно определению корректного детерминированного эвристического аксептора:

$$\Pr_{x \leftarrow U_n(\overline{\operatorname{Im} f})} \{A_i(x, 8dn^2) = 1\} < \frac{1}{8dn^2} = \delta.$$

Тогда по лемме 3.1 процедура Verify примет аксептор A_i по крайней мере на $1 - 4\delta$ доле всех входов из $\operatorname{Im} f$. Значит, по крайней мере с вероятностью $1 - 4\delta$, $R(x, d)$ будет тратить только полиномиальное время от времени работы A_i. То есть

$$\Pr_{x \leftarrow U_n(\operatorname{Im} f)} \{t_U(x, d) \leq p(nd \cdot t_{A_i}(x, 8dn^2))\} \geq 1 - 4\delta = 1 - \frac{1}{2dn^2} \geq 1 - \frac{1}{2d},$$

где p — полиномиальное время работы, требующееся для моделирования алгоритма A_i. □

4. Следствия основных результатов

4.1. Оптимальный эвристический алгоритм для $(\mathrm{Im}\, f, U)$

В двух предыдущих секциях мы построили детерминированные эвристические аксепторы для задач $(\mathrm{Im}\, f, U(\overline{\mathrm{Im}\, f}))$ и $(\overline{\mathrm{Im}\, f}, U(\mathrm{Im}\, f))$, где f — инъективная функция, увеличивающая свой вход на один символ.

В этой секции мы объединим две предложенные конструкции и получим оптимальный эвристический детерминированный алгоритм для языка $\mathrm{Im}\, f$ или, говоря иначе, алгоритм, решающий распределенную задачу $(\mathrm{Im}\, f, U)$.

Будем строить оптимальный детерминированный эвристический алгоритм в общем виде: для распределенной задачи (L, D), в предположении, что для задач $(L, D(\overline{L}))$ и $(\overline{L}, D(L))$ существуют оптимальные детерминированные эвристические аксепторы, где $D(L)$ $(D(\overline{L}))$ — это часть распределения D, сосредоточенная на L (на $\overline{L}$ соответственно).

Алгоритм 4.1 (Алгоритм $Opt(x, d)$)**.** Запустить параллельно оптимальный аксептор — $R(x, 16d)$ для $(L, D(\overline{L}))$ и оптимальный аксептор — $S(x, 16d)$ для $(\overline{L}, D(L))$. Если первым остановился алгоритм R, остановиться и выдать 1, если же первым остановился алгоритм S, остановиться и выдать 0.

Теорема 4.1. *Opt* является корректным эвристическим разрешающим алгоритмом.

Доказательство. Заметим, что поскольку процедура R всегда завершает свою работу на L, а процедура S всегда завершает свою работу на $\overline{L}$, наш алгоритм $Opt(x, d)$ также всюду останавливается. Тогда имеет место

цепочка неравенств:

$$\Pr_{x \leftarrow D_n}\{Opt(x,d) \neq \operatorname{Im} f(x)\} <$$
$$\Pr_{x \leftarrow D_n(L)}\{Opt(x,d) = 0\} + \Pr_{x \leftarrow D_n(\overline{L})}\{Opt(x,d) = 1\} <$$
$$\Pr_{x \leftarrow D_n(L)}\{S(x,16d) = 1\} + \Pr_{x \leftarrow D_n(\overline{L})}\{R(x,16d) = 1\} < \frac{1}{16d} + \frac{1}{16d} = \frac{1}{8d}.$$

□

Теорема 4.2. *Opt* является оптимальным эвристическим разрешающим алгоритмом.

Доказательство. Пусть $\mathcal{L}$ — процедура разрешающая язык L. $\mathcal{L}(x) = 1$, если $x \in L$ и $\mathcal{L}(x) = 0$, иначе.

Рассмотрим некоторый эвристический разрешающий алгоритм $A(x,d)$ для задачи (L,D). По определению этот алгоритм отвечает либо 0, либо 1.

По алгоритму $A(x,d)$ построим алгоритмы R_A и S_A: $R_A(x,d)$ запускает $A(x,16d)$ и, если $A(x,16d)$ возвращает 1, тоже останавливается и возвращает 1, иначе запускает алгоритм $\mathcal{L}(x)$. Если $\mathcal{L}(x)$ вернул 1, останавливается и выдает 1, иначе зацикливается.

Алгоритм S_A будет действовать также, только дожидаться от запускаемых процедур ответа 0 и возвращает также только 0.

Алгоритмы R_A и S_A являются корректными детерминированными эвристическими аксепторами для задач $(L, D(\overline{L}))$ и $(\overline{L}, D(L))$ соответственно.

В алгоритме *Opt* используются оптимальные аксепторы $S(x,16d)$ и $R(x,16d)$, значит имеет место отношение моделирования для некоторых полиномов q_R, p_R:

$$\Pr_{x \leftarrow D_n(L)}\{t_R(x,16d) \leq p_R(ndt_{R_A}(x, q_R(nd)))\} \geq 1 - \frac{1}{32d}. \tag{4.1}$$

По определению доминирования $q_R(nd) > 2d$.

Из конструкции оптимального алгоритма, а точнее, из доказательства корректности, следует, что:

$$\Pr_{x \leftarrow D_n(L)}\{t_{Opt}(x,d) = t_R(x,16d)\} \geq \Pr_{x \leftarrow D_n(L)}\{t_{Opt}(x,d) = 1\} \geq 1 - \frac{1}{4d}. \tag{4.2}$$

Из определения процедуры R_A:

$$\Pr_{x \leftarrow D_n(L)}\{t_A(x, 16q_R(nd)) = t_{R_A}(x, q_R(nd))\} \geq 1 - \frac{2}{16q_R(nd)} > \frac{1}{16d}. \quad (4.3)$$

Последнее неравенство верно благодаря тому, что $q_R(nd) > 2d$.

Объединяя соотношения (4.1), (4.2) и (4.3), получаем:

$$\Pr_{x \leftarrow D_n(L)}\{t_{Opt}(x, d) \leq p_R(ndt_A(x, 16q_R(nd)))\} \geq 1 - \frac{1}{16d} - \frac{1}{4d} - \frac{1}{32d} > 1 - \frac{1}{2d}. \quad (4.4)$$

Аналогичное неравенство справедливо для входов из $\overline{L}$ для некоторорых полиномов p_S, q_S:

$$\Pr_{x \leftarrow D_n(\overline{L})}\{t_{Opt}(x, d) \leq p_S(ndt_A(x, 8q_S(nd)))\} \geq 1 - \frac{1}{2d}. \quad (4.5)$$

Объединяя (4.4) и (4.5), получаем требуемое:

$$\Pr_{x \leftarrow D_n}\{t_{Opt}(x, d) \leq max\{p_S(ndt_A(x, 8q_S(nd))), p_R(ndt_A(x, 8q_R(nd)))\}\} \geq 1 - \frac{1}{2d}.$$

Последнее неравенство говорит о том, что алгоритм Opt почти везде моделирует алгоритм A, что завершает доказательство оптимальности.

□

Следствие 4.1. Для любой инъективной функции f, увеличивающей свой вход на один символ для задачи $(\operatorname{Im} f, U)$ существует оптимальный эвристический детерминированный алгоритм.

4.2. Эвристические системы доказательств

Определение 4.1. Детерминированная машина Тьюринга называется *эвристической системой доказательств* для языка L, если она удовлетворяет следующим условиям

1. Время работы $\Pi(x, w, d)$ ограничено полиномом от d, $|x|$, $|w|$, т.е. $t_\Pi(x, w, d) \leq q(d \cdot |x| \cdot |w|)$
2. (Полнота) Для любого $x \in L$ и любого $d \in \mathbb{N}$, существует строка w такая, что $\Pi(x, w, d) = 1$
3. (Корректность) $\Pr_{x \leftarrow U_n(\overline{L})}\{\exists w : \Pi(x, w, d) = 1\} < \frac{1}{d}$

Для эвристической системы доказательств введем обозначение для длины кратчайшего доказательства в этой системе: $l_\Pi(x,d) = min\{|w| : \Pi(x,w,d) = 1\}$.

Определение 4.2. Эвристическая система доказательств Π называется автоматизируемой, если существует детерминированный алгоритм A и полином p, такие что

1. Для любого $x \in L$ и любого $d \in \mathbb{N}$, $A(x,d) = w$, где w — корректное доказательство в Π и $|w| \leq p(d \cdot |x| \cdot l_\Pi(x,d))$
2. Время работы алгоритма $A(x,d)$ ограничено полиномом от $|x|$, d и размера собственного выхода для $x \in L$.

Теорема 4.3. Пусть (Π, A) — эвристическая автоматизируемая система доказательств для рекурсивно перечислимого языка L, тогда алгоритм $B(x,d)$, запускающий параллельно $\Pi(x, A(x,d), d)$ и процедуру $\mathcal{L}(x)$, возвращающую 1, если $x \in L$, и не останавливающуюся в противном случае, и выдающий первый полученный ответ, является эвристическим аксептором для языка L, и время работы алгоритма B почти везде доминирует над длиной кратчайшего доказательства в системе Π, т.е. $l_\Pi(x,d) \succsim t_B(x,d)$ для $x \in L$.

Доказательство. Алгоритм $B(x,d)$ всегда возвращает 1 на языке L благодаря запуску полуразрешающей процедуры $\mathcal{L}(x)$.

Рассмотрим $\Delta_n = \{x \in \overline{L} \mid \exists w : \Pi(x,w,d) = 1\}$. Согласно определению эвристической системы доказательств $\rho_n(\Delta_n) < \frac{1}{d}$ $(\rho_n(S) = \frac{|S|}{2^n}) \Rightarrow \Pr_{x \leftarrow U_n(\overline{L})}\{\Pi(x, A(x,d), d) = 1\} < \frac{1}{d}$ тогда $\Pr_{x \leftarrow U_n(\overline{L})}\{B(x,d) = 1\} < \frac{1}{d}$. Значит, $B(x,d)$ является корректным детерминированным эвристическим аксептором.

Осталось доказать факт доминирования на L:

$$\begin{aligned} t_B(x,d) &\leq t_\Pi(x, A(x,d), d) \leq \\ &q(|x| \cdot A(x,d) \cdot d) \leq \\ &q(|x| \cdot p(|x| \cdot d \cdot l_\Pi(x,d))) \Rightarrow l_\Pi(x,d) \succsim t_B(x,d). \end{aligned}$$

□

Теорема 4.4. Пусть C — эвристический аксептор для языка L. Тогда существует автоматизируемая эвристическая система доказательств Π_c для L такая что $t_c \succsim l_{\Pi_c}$.

Доказательство. Рассмотрим детерминированный алгоритм $\Pi_c(x, 1^T, d)$, который запускает $C(x, d)$ и возвращает 1, если $C(x, d)$ остановился через T шагов. Π_c является автоматизируемой эвристической системой доказательств. Действительно, полнота очевидна, поскольку аксептор всегда принимает $x \in L$, поэтому для некоторого T, 1^T будет доказательством строки x. Корректность системы доказательств следует из определения аксептора. Время работы $\Pi_c(x, 1^T, d)$ ограничено полиномом от T. □

Определение 4.3. Система доказательств Π_1 *моделирует* систему доказательств Π_2, если $l_{\Pi_2} \succsim l_{\Pi_1}$. Система доказательств называется оптимальной, если она моделирует любую другую.

Теорема 4.5. Для языков $\operatorname{Im} f$ и $\overline{\operatorname{Im} f}$ существует оптимальная эвристическая автоматизируемая система доказательств.

Доказательство. По теореме 4.3 каждая автоматизируемая система Π для $\operatorname{Im} f$ ($\overline{\operatorname{Im} f}$) порождает эвристический аксептор A такой, что $l_\Pi(x, d) \succsim t_A(x, d)$. Аксептор A моделируется оптимальным аксептором R для $\operatorname{Im} f$ ($\overline{\operatorname{Im} f}$), т.е. $t_A(x, d) \succsim t_R(x, d)$. По теореме 4.4 эвристический аксептор R порождает автоматизируемую эвристическую систему доказательств Π_R такую, что $t_R(x, d) \succsim l_{\Pi_R}(x, d)$. По транзитивности Π_R моделирует Π. □

Заключение

В работе удалось выполнить все поставленные задачи: перенести результат, полученный в статьях [11, 12] на детерминированный случай для задачи распознавания $(\overline{\mathrm{Im}\, f}, U(\mathrm{Im}\, f))$ (где f — инъективная полиномиально вычислимая функция, удлинняющая свой вход на один символ, а $U(S)$ — равномерное распределение на множестве S), построить аналогичный алгоритм для задачи $(\mathrm{Im}\, f, U(\overline{\mathrm{Im}\, f}))$. Объединив полученные конструкции, удалось построить эвристической алгоритм для языка $\mathrm{Im}\, f$ с ограниченной ошибкой. На основе полученных результатов удалось построить оптимальную автоматизируемую эвристическую систему доказательств.

Интерес для дальнейших исследований представляет обобщение полученных результатов на произвольные функции f (необязательно инъективные).

Список литературы

[1] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, March 1979.

[2] Jan Krajíček and Pavel Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54(3):1063–1079, September 1989.

[3] 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.

[4] Johannes Köbler, Jochen Messner, and Jacobo Torán. Optimal proof systems imply complete sets for promise classes. *Information and Computation*, 184(1):71–92, 2003.

[5] Zenon Sadowski. On an optimal quantifed propositional proof system and a complete language for NP$\cap$co$-$NP. *In Proceedings of FCS'97*, 1279:423–428, 1997.

[6] Alexander A. Razborov. On provably disjoint NP-pairs. *Electronic Colloquium on Computational Complexity*, 94-006, 1994.

[7] Pavel Pudlák. On reducibility and symmetry of disjoint NP-pairs. *Theoretical Computer Science*, 295(1-3):323–339, 2003.

[8] S. Even, A. Selman, and J. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61:159–173, 1984.

[9] Leonid A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9:265–266, 1973.

[10] Stephen A. Cook and Jan Krajíček. Consequences of the provability of NP $\subseteq$ P/poly. *The Journal of Symbolic Logic*, 72(4):1353–1371, 2007.

[11] Edward A. Hirsch and Dmitry Itsykson. On optimal heuristic semidecion procedure with applications in proof complexity. In *Proceedings of 27th International Symposium on Theoretical Aspects of Computer Science*, pages 316–324, 2010.

[12] Edward A. Hirsch, Dmitry Itsykson, Ivan Monakhov, and Alexander Smal. On optimal heuristic semidecion procedure with applications in proof com-

plexity and cryptography. *Electronic Colloquium on Computational Complexity*, TR10-193, 2010.

[13] Э. А. Гирш и Д. М. Ицыксон. Бесконечно часто односторонняя функция, основанная на предположении о сложности в среднем. *Алгебра и анализ*, 21(3):130–144, 2009.

[14] Edward A. Hirsch. Optimal acceptors and optimal proof systems. In *Proceedings of the TAMC*, volume 6105 of *Lecture Notes in Computer Science*, pages 28–39, 2010.

[15] Oded Goldreich. *Foundation of Cryptography: Basic Tools*. Cambridge University Press, 1995.

[16] P. Sarnak A. Lubotzky, R. Phillips. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

[17] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22:407–420, 1981.

[18] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Structures and Algorithms*, 11(4):315–343, 1997.