КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

А.Ф. ГАЙНУТДИНОВА, М.Ф. АБЛАЕВ, А.И. ХАДИЕВА

КОММУНИКАЦИОННЫЕ ВЫЧИСЛЕНИЯ

Учебное пособие

КАЗАНЬ
2024

УДК 519.71
ББК 22.18

Рекомендовано к изданию
Учебно-методической комиссией ИВМиИТ
(протокол № 3 от 26 ноября 2024 года)

Рецензенты:
доктор технических наук, профессор **Шалагин С.В.**
кандидат физико-математических наук, доцент **Васильев А.В.**

Гайнутдинова А.Ф., Аблаев М.Ф., Хадиева А.И.
Коммуникационные вычисления: учебное пособие / А.Ф. Гайнутдинова, М.Ф. Аблаев, А.И. Хадиева – Казань: Казан. ун-т, 2024, – 74 с.

В учебном пособии излагаются основы теории коммуникационных вычислений. Рассматриваются различные варианты коммуникационной модели: детерминированная, недетерминированная, вероятностная. Определяется понятие протокола, вводится понятие сложности, приводятся примеры протоколов, вычисляющих булевы функции, рассматриваются методы доказательства нижних оценок коммуникационной сложности функций. Приводятся приложения методологии коммуникационных вычислений к другим областям. Предназначено для студентов ВУЗов, аспирантов, преподавателей и научных работников, ведущих исследования в области компьютерных наук.

УДК 519.71
ББК 22.18

Оглавление

Глава 1

Детерминированная коммуникационная модель

1.1 Коммуникационные протоколы, сложность коммуникационных вычислений

Под коммуникационными вычислениями понимаются распределенные вычисления. В коммуникационной модели имеются два (или более) участников (вычислителей), которые совместно пытаются решить некоторую задачу. При этом входные данные распределены между ними таким образом, что каждый из участников знает лишь часть входа. Для решения задачи участники вынуждены коммуницировать друг с другом, передавая информацию другу другу по каналу. Алгоритмы, определяющие коммуникационные вычисления называются протоколами.

Коммуникационные вычисления булевых функций определяют следующим образом. Пусть f булева функция следующего вида $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Даны два вычислителя с неограниченными вычислительными возможностями (как правило один из них называют A, т.е. Алиса , а другой B, т.е. Боб), каждый из них получает на вход последовательность σ и γ соответственно, длиной n бит каждая. Ни один из них не знает входного набора, который получил другой вычислитель, и они совместно хотят вычислить значение функции f на наборе (σ, γ). Для вычисления значения $f(\sigma, \gamma)$ A и B обмениваются сообщениями (двоичными последовательностями) согласно протоколу (алгоритму) Ф. Под сложностью протокола Ф на наборах σ и γ понимают количество бит, переда-

ваемых вычислителями. В теории коммуникационных вычислений исследуются различные варианты коммуникационных вычислений. Теория и техника коммуникационных вычислений имеет обширную область применений. В частности она используется при доказательстве нижних оценок времени и памяти реализации вычислений на машинах Тьюринга, в исследовании сложности вычислений для ряда других моделей вычислений.

Структуры данных, такие как частично упорядоченные полные бинарные деревья, массивы сортировки, списки и т. д. — основные объекты в алгоритмических конструкциях. Были исследованы многие разновидности схемы, описанной выше, такие как: вероятностные протоколы, недетерминированные, выборочные и т.д. Более того, нижняя оценка коммуникационной сложности используется в СБИС.

Дадим формальные определения протокола вычисления булевой функции и коммуникационной сложности протокола булевой функции. В данном пособии мы будем рассматривать коммуникационные протоколы, у которых ответ может выдавать только вычислитель B. То есть количество раундов в вычислениях нечетно.

Определение 1.1 *t-раундовым коммуникационным протоколом* Φ *для булевой функции $f(x,y)$ называется алгоритм:*

1. *Вычислитель A получает на вход $\sigma \in \{0,1\}^n$, вычислитель B получает $\gamma \in \{0,1\}^n$.*

2. *Первый раунд. Вычислитель A начинает вычисления: по σ определяет сообщение*

$$m^1(\sigma) = m^1 = \{m^1_1, m^1_2, ...m^1_{t_1}\} \in \{0,1\}^{t_1}, t_1 = t_1(\sigma)$$

   и передает его B.

3. *Вычислитель B получает m^1. Если вычислитель B по γ, m^1 может вычислить значение функции, то он выдаёт значение функции $f(\sigma,\gamma)$ и выполнение протокола завершается. В противном случае выполняется второй раунд протокола: вычислитель B формирует сообщение*

$$m^2(\gamma, m^1) = m^2 = {m^2}_1 ... {m^2}_{t_2} \in \{0,1\}^{t_2}$$

и отправляет вычислителю A сообщение m^2.

4. *Вычислитель A получает m^2, формирует сообщение m^3 и отправляет его вычислителю B и т.д.*

Определение 1.2 *Коммуникационным сообщением протокола Φ на входном наборе σ, γ называется двоичная последовательность:*

$$m = m^1 m^2 \dots m^t = m_\Phi(\sigma, \gamma)$$

Определение 1.3 *Сложностью коммуникационного протокола на наборе (σ, γ) называется количество бит, которыми обмениваются вычислители в протоколе Φ (количество бит в коммуникационном сообщении):*

$$C_\Phi(\delta, \gamma) = \left|m^1\right| + \left|m^2\right| + ... + \left|m^t\right| = \left|m_\Phi(\delta, \gamma)\right|$$

Определение 1.4 *Сложностью коммуникационного протокола Φ называется величина:*

$$C(\Phi) = \max_{\delta,\gamma \in \{0,1\}^n} C_\Phi(\delta, \gamma)$$

Определение 1.5 *Коммуникационный протокол Φ вычисляет функцию $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, если для любого $(\sigma, \gamma) \in \{0,1\}^n \times \{0,1\}^n$ значение, выдаваемое вычислителем B в качестве ответа совпадает со значением $f(\sigma, \gamma)$.*

Определение 1.6 *Коммуникационная сложность $C(f)$ для функции f определяется как сложность наилучшего протокола, вычисляющего функцию f:*

$$C(f) = \min_{\Phi \text{ вычисляет } f} C(\Phi)$$

Далее мы рассмотрим примеры протоколов.

EQ Рассмотрим функцию $EQ(x, y)$ и построим протокол Φ для неё. На наборе (σ, γ) функция EQ равна 1, если $\sigma = \gamma$ и 0 в противном случае. Построим протокол Φ: вычислитель A получает на вход σ — последовательность из n бит, вычислитель B получает последовательность γ. Вычислитель A передаёт вычислителю B всё сообщение σ, вычислитель B, зная σ, γ, производит их побитовое сравнение и выдаёт ответ. Сложность такого протокола $C(\Phi) = n$, поэтому $C(EQ) \leq n$.

NEQ Теперь рассмотрим функцию $NEQ(x, y)$ и построим протокол Φ для неё. На наборе (σ, γ) функция NEQ равна 1, если $\sigma \neq \gamma$ и 0 в противном случае. Построим протокол Φ: вычислитель A получает на вход σ — последовательность из n бит, вычислитель B получает последовательность γ. Вычислитель A передаёт вычислителю B всё сообщение σ, вычислитель B, зная σ, γ, производит их побитовое сравнение и выдаёт ответ. Сложность такого протокола $C(\Phi) = n$, поэтому $C(NEQ) \leq n$.

PARITY $Parity(z) = z_1 \oplus z_2 \oplus ... \oplus z_{2n}$. Рассмотрим два коммуникационных протокола для функции $Parity$.

1. Протокол Φ_1 для функции $Parity$: пусть вычислитель A получает на вход σ — последовательность из n бит, B получает γ. Вычислитель A передаёт вычислителю B всё сообщение σ, вычислитель B, зная σ, γ, производит их побитовое сложение и выдаёт ответ. Сложность такого протокола $C_{\Phi_1}(\sigma, \gamma) = n$, поэтому $C(Parity) \leq n$.

2. Для этой же функции построим ещё один протокол Φ_2, в котором оба вычислителя получают на вход те же последовательности, но вычислитель A сначала выполняет побитовое сложение последовательности σ и передаёт вычислителю B результат этого сложения. Вычислитель B, зная этот результат, выполняет побитовое сложение последовательности γ, к результату этого сложения прибавляет результат, полученный им от вычислителя A и выдаёт ответ. Сложность такого протокола $C_{\Phi_2}(\sigma, \gamma) = 1$. Таким образом, $C(Parity) \leq 1$.

Из протокола Φ_2 получаем верхнюю оценку $C(Parity) \leq 1$. Так как функция $Parity$ существенным образом зависит от всех переменных, то вычислитель A должен передать вычислителю B хотя бы один бит и значит $C(Parity) \geq 1$. Отсюда получаем $C(Parity) = 1$.

MODm Рассмотрим функцию $MOD_m(x, y)$ и построим протокол Ф. Пусть (σ, γ) входной набор. Обозначим $n(\sigma)$ $(n(\gamma))$ — целое неотрицательное число, двоичным представлением которого является последовательность σ (γ). На наборе (σ, γ) функция MOD_m равна 1, если $n(\sigma)$ равно $n(\gamma)$ по модулю m, то есть при делении на m оба числа дают одинаковый остаток. Значение функции равно 0 в противном случае. Построим протокол Ф, вычисляющий функцию $MOD_m(x, y)$: вычислитель A получает на вход σ — последовательность из n бит, B получает последовательность γ. Вычислитель A вычисляет $r(\sigma) = n(\sigma) \pmod m$ — остаток от деления $n(\sigma)$ на m и передаёт вычислителю B. Вычислитель B вычисляет $r(\gamma) = n(\gamma) \pmod m$, сравнивает $r(\sigma)$ и $r(\gamma)$ и выдаёт ответ. Сложность такого протокола $C(\Phi) = \log m$, количество бит необходимых для передачи числа меньшего m . Таким образом сложность функции будет $C(MOD_m) \leq \log m$.

Теорема 1.1 *Для произвольной булевой функции $f(x, y)$ выполняется $C(f) \leq n$.*

Доказательство: Для произвольной булевой функции $f(x, y)$ построим протокол Ф. Пусть на вход вычислителя A поступает набор σ, B получает γ. A генерирует следующее сообщение $m = \sigma_1\sigma_2 \ldots \sigma_n$. Тогда вычислитель B, получив это сообщение, будет знать оба входных набора σ и γ и исходя из определения коммуникационных вычислений, может выдать ответ. В связи с произвольностью выбора функции f, получаем что $C(f) \leq C(\Phi) = n$. $\square$

1.2 Методы доказательства нижних оценок коммуникационной сложности

Покажем, что построенный выше протокол для функции EQ является наилучшим и не существует протокола меньшей сложности, вычисляющий функцию EQ. Для этого воспользуемся одним из методов доказательства нижних оценок коммуникационной сложности — методом «полных множеств».

1.2.1 Метод полных множеств "fooling set"

Продемонстрируем схему доказательства, основанного на методе полных множеств, на примере функции EQ.

Теорема 1.2 $C(EQ) \geq n$.

Доказательство: Докажем от противного. Предположим, что существует протокол Ф, вычисляющий функцию EQ и имеющий сложность не более $n-1$. Это означает, что множество возможных сообщений, которыми вычислителя могут обмениваться друг с другом, содержит не более чем $2^0+2^1+...+2^{n-1} = 2^n-1$ элементов.

Рассмотрим множество всех пар вида (σ, σ):

$$S = \{(\sigma, \sigma) \,|\, \sigma \in \{0,1\}^n\}.$$

Заметим, что $|S| = 2^n$. Так как число различных сообщений, которые могут использоваться в протоколе Ф, строго меньше 2^n, то по принципу Дирихле, найдутся две пары (σ, σ) и (σ', σ'), для которых в протоколе используется одно и то же коммуникационное сообщение m. Понятно, что $EQ(\sigma, \sigma) = EQ(\sigma', \sigma') = 1$ и протокол на наборах (σ, σ) и (σ', σ') должен выдавать ответ 1. Теперь рассмотрим вычисление на входе (σ, σ'). Для неё коммуникационное сообщение будет таким же, как для (σ, σ) и для (σ', σ'). Действительно, если вычислитель A отправляет бит первым, тогда этот бит будет таким же, как и для σ, и для σ'. Если вычислитель B отправляет сообщение во втором раунде, тогда его бит должен быть таким же, как для σ и для σ' до тех пор, пока он получает такой же бит от вычислителя A. Таким образом, ответ протокола на наборе (σ, σ) должен совпадать с ответом протокола на наборе (σ, σ').

Но $EQ(\sigma, \sigma') = 0$, а $EQ(\sigma, \sigma) = 1$. Получили противоречие с тем, что протокол Ф правильно вычисляет функцию EQ. Следовательно, выполняется $C(EQ) \geq n$. $\square$

Определение 1.7 *Полное множество (fooling set или FS_f) для функции $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ — это множество входных наборов $S \subseteq \{0,1\}^n \times \{0,1\}^n$ такое, что существует $b \in \{0,1\}$ при котором выполняется:*

1. *для любой пары $(\sigma, \gamma) \in S$, $f(\sigma, \gamma) = b$,*
2. *для любых двух разных пар $(\sigma_1, \gamma_1), (\sigma_2, \gamma_2) \in S$ выполняется либо $f(\sigma_1, \gamma_2) \neq b$, либо $f(\sigma_2, \gamma_1) \neq b$.*

Теорема 1.3 *Для произвольной булевой функции f выполняется*

$$C(f) \geq \log |FS_f|.$$

Определим функцию $DISJ$ (функцию «непересечения»). Пусть σ, γ — характеристические вектора подмножеств $S_\sigma, S_\gamma \subseteq \{1, 2, ..n\}$.

$$DISJ(\delta, \gamma) = \begin{cases} 1, & \text{если } \delta \cap \gamma = 0, \\ 0, & \text{если } \delta \cap \gamma \neq 0. \end{cases}$$

Теорема 1.4 $C(DISJ) \geq n$

Теорема 1.5 $C(MOD_m) \geq \log m$

Теорема 1.6 $C(EQ) \geq n$

Теорема 1.7 $C(NEQ) \geq n$

Теорема 1.8 $C(Parity) \geq 1$

1.2.2 Метод монохроматических прямоугольников

Рассмотрим метод доказательства нижних оценок коммуникационной сложности, основанный на разбиении коммуникационной матрицы на монохроматические прямоугольники.

Коммуникационная матрица — это табличный способ задания булевой функции, отражающий распределение входных данных между двумя вычислителями. Формально, коммуникационная матрица $CM(f)$ булевой функции $f(x,y) : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ — это таблица с 2^n строками и 2^n столбцами. Строки пронумерованы всевозможными наборами σ, столбцы пронумерованы всевозможными наборами γ. На пересечении строки σ и столбца γ стоит 0 или 1 — значение функции $f(\sigma,\gamma)$.

Определение 1.8 *Комбинаторным прямоугольником $A \times B$ называется подматрица, составленная из элементов, находящихся на пересечении строк, входящих в множество A и столбцов, входящих в множество B, где $A, B \subseteq \{0,1\}^n$.*

Определение 1.9 *Комбинаторный прямоугольник $A \times B$ называется монохроматическим, если существует $b \in \{0,1\}$, такое что для любого $\sigma \in A$, и для любого $\gamma \in B$, значение $f(\sigma,\gamma) = b$.*

Пусть Ф — коммуникационный протокол. Если протокол начинает работу с сообщения от первого вычислителя, длиной в один бит, тогда $CM(f)$ разбивается на два прямоугольника типа $A_0 \times \{0,1\}^n, A_1 \times \{0,1\}^n$, где A_b — подмножество строк, для которых сообщения первого вычислителя равно b. При этом $A_0 \cup A_1 = \{0,1\}^n$. Если следующий бит послан вторым вычислителем, тогда каждый из двух прямоугольников разбивается на два меньших прямоугольника, зависящих от того, какой бит был послан. Если протокол отработал k шагов, матрица будет состоять из 2^k прямоугольников. После k шагов вычислитель В выдаёт правильный ответ, значит каждый такой прямоугольник соответствует подмножеству входных пар, для которых значение функции одинаково. Если протокол остановил свою работу, то значение f внутри каждого прямоугольника должно быть одинаковым для всех пар σ, γ в этом прямоугольнике. Значит, коммуникационный протокол должен привести к разбиению коммуникационной матрицы на монохроматические прямоугольники.

Определение 1.10 *Монохроматическое разбиение для матрицы $CM(f)$ — это разбиение $CM(f)$ на непересекающиеся монохрома-*

тические прямоугольники. Обозначим через $\chi(f)$ минимальное число монохроматических прямоугольников, на которые мы можем разбить $CM(f)$.

Следующая теорема следует непосредственно из рассуждений, сделанных выше.

Теорема 1.9 *Если f имеет коммуникационную сложность C, тогда её монохроматическое разбиение содержит не более чем 2^C прямоугольников. Т.е. $C \geq \log \chi(f)$.*

Лемма 1.1 *Для произвольной функции f выполняется*

$$\chi(f) \geq |FS_f|.$$

Доказательство: Пусть полное множество для функции f содержит m пар. Две различные пары (σ_1, γ_1) и (σ_2, γ_2) не могут принадлежать одному монохроматическому прямоугольнику. Следовательно, $\chi(f) \geq m$. □

1.2.3 Метод ранга коммуникационной матрицы

В данном разделе мы рассмотрим оценку коммуникационной сложности $\chi(f)$ в терминах ранга коммуникационной матрицы. Напомним, что ранг матрицы в поле F — это максимальное число линейно независимых строк или столбцов. Для ранга матрицы существует и другое определение:

Определение 1.11 *Ранг матрицы M размера $n \times n$ — это минимальное значение l такое, что M может быть представлена в виде $M = \sum_{i=1}^{l} \alpha_i B_i$, где $\alpha_i \in F \backslash \{0\}$ и каждая матрица B_i — это $n \times n$ матрица ранга 1.*

Отметим, что $0, 1$ являются элементами любого поля, поэтому мы можем вычислить ранг над любым полем (выбор поля может быть ключевым).

Теорема 1.10 *Для любой функции f выполняется*

$$\chi(f) \geq rank(CM(f)).$$

Доказательство: каждый монохроматический прямоугольник может быть представлен как матрица ранга не больше чем 1. □

1.3 Функция Inner Product

Определим функцию IP (Inner Product) скалярного (внутреннего) произведения по модулю 2 для двух векторов $x, y \in \{0,1\}^n$:

$$IP(x,y) = \bigoplus_{i=1}^{n} x_i y_i.$$

Задача состоит в том, чтобы два участника, Алиса и Боб, нашли значение функции $IP(x,y)$ при условии, что Алиса знает x, а Боб знает y.

Пример:

- Пусть $x = (1,0,1,1)$ и $y = (0,1,1,0)$.
- Тогда $IP(x,y) = (1\cdot 0 + 0\cdot 1 + 1\cdot 1 + 1\cdot 0) \mod 2 = (0+0+1+0) \mod 2 = 1$.

Теорема 1.11 $C(IP) = n$.

Доказательство: Неравенство $C(IP) \leq n$ очевидно. Докажем нижнюю оценку. Воспользуемся методом ранга коммуникационной матрицы.

Пусть $M = CM(IP)$ — коммуникационная матрица. Нужно оценить её ранг. Возведем матрицу M в квадрат: $N = M^2$. В матрице N первый столбец и первая строка состоят из нулей. Элемент, стоящий на пересечении строки x и столбца y матрицы N равен

$$N[x,y] = \sum_{z \in \{0,1\}^n} IP(x,z)IP(z,y).$$

Заметим, что $x_i z_i$ и $z_i y_i$ это либо 0 либо 1, так как x_i, z_i, y_i — булевские переменные.

Рассмотрим два случая:

1. $x = y$. Тогда $N[x, x]$ равно числу решений уравнения

$$x_1 z_1 \oplus \cdots \oplus x_n z_n = 1 \tag{1.1}$$

относительно переменной z для фиксированного x.

Уравнение невырождено при $x \neq 0$. Пусть $x \neq 0$ — произвольный набор. Будем строить наборы z, для которых выполняется (1.1). При поразрядном умножении набора x на набор z ненулевые биты могут получаться только если на соответствующих позициях в x и z стоят единицы. Зафиксируем произвольную позицию i, для которой $x_i = 1$ (такая позиция существует, так как $x \neq 0$). Для всех позиций $j \neq i$ в качестве значения z_j будем выбирать произвольное значение $\in \{0, 1\}$. Получим 2^{n-1} наборов z с незаполненной i-ой позицией. Для каждого построенного набора в качестве значения z_i будем выбирать такое, чтобы равенство (1.1) обратилось в истину, т. е.

$$z_i = 1 \oplus \bigoplus_{j \neq i} x_j z_j.$$

Получили, что для любого $x \neq 0$ значение $N[x, x] = 2^{n-1}$.

2. $x \neq y$. В этом случае $N[x, y]$ — число таких z, что скалярное произведение по модулю 2 вектора z и на вектор x и на вектор y равно 1. Имеем следующую систему уравнений над полем из двух элементов относительно переменных z_i $(i = 1, \ldots, n)$:

$$\begin{cases} x_1 z_1 \oplus \cdots \oplus x_n z_n = 1 \\ y_1 z_1 \oplus \cdots \oplus y_n z_n = 1 \end{cases} \tag{1.2}$$

То есть $N[x, y]$ равно числу наборов z для которых оба равенства обращаются в 1. Так как $x \neq y \neq 0$, то найдется номер i такой, что $x_i \neq y_i$. Пусть для определенности $x_i = 1$, $y_i = 0$. Найдем число наборов z, для которых $y_1 z_1 \oplus \cdots \oplus y_n z_n = 1$. Как было показано выше, число таких наборов равно 2^{n-1}. При построении наборов z в качестве z_i мы могли выбирать и 0 и 1, так как $y_i = 0$. Теперь добьёмся выполнения равенства

$x_1z_1 \oplus \cdots \oplus x_nz_n = 1$, выбирая в качестве z_i значение 0 или 1, при котором указанное равенство обращается в 1, т.е.

$$z_i = 1 \oplus \bigoplus_{j \neq i} x_j z_j.$$

Число наборов сократилось в два раза. Получили, что для любых x, y таких, что $x, y \neq 0, x \neq y$ значение $N[x, y] = 2^{n-2}$.

Таким образом

$$N = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 2^{n-1} & 2^{n-2} & \dots & 2^{n-2} \\ 0 & 2^{n-2} & 2^{n-1} & \dots & 2^{n-2} \\ \dots & & \ddots & \dots & \\ 0 & 2^{n-2} & 2^{n-2} & \dots & 2^{n-1} \end{pmatrix}$$

Ранг этой матрицы равен $2^n - 1$. Теперь найдем ранг исходной матрицы $M = CM(IP)$.

- $rank(M) < 2^n$, так как при возведении в квадрат получилась нулевая строка и нулевой столбец (при возведении в квадрат матрицы полного ранга получаем матрицу полного ранга).
- При возведении в квадрат ранг не возрастает.

Поэтому $rank(CM(IP)) = 2^n - 1$. По теореме о нижней оценке получаем $C(IP) \geq n$. □

1.4 Многораундовые коммуникационные вычисления

Определение 1.12 *Многораундовыми или $t-$раундовыми коммуникационными вычислениями булевой функции $f(X, Y)$ называется алгоритм:*

1. *Вычислитель A получает на вход $\sigma \in \{0, 1\}^n$,вычислитель B получает $\gamma \in \{0, 1\}^n$*
2. *Вычислитель A начинает вычисления: по σ определяет сообщение*

$$m^1(\sigma) = m^1 = \{m_1^1, m_2^1, ...m_{t_1}^1\} \in \{0, 1\}^{t_1}, t_1 = t_1(\sigma)$$

и передает его B.

3. *Вычислитель B получает m^1. Если по m^1 и γ вычислитель B может вычислить значение функции, то он выдаёт ответ — значение $f(\sigma, \gamma)$. В противном случае B формирует сообщение*
$$m^2 = m_1^2 ... m_{t_2}^2 \in \{0,1\}^{t_2}$$
и отправляет вычислителю A сообщение m^2.

4. *Вычислитель A получает m^2. Если вычислитель A по σ, m^1, m^2 может вычислить значение функции, то он выдаёт значение функции $f(\sigma, \gamma)$. В противном случае A формирует m^3 и отправляет его вычислителю B и т.д.*

Согласно рассматриваемой модели коммуникационных вычислений, булева функция $f(x, y)$ вычисляется с нечётным количеством раундов. В следующем разделе мы рассмотрим однораундовые коммуникационные вычисления.

1.4.1 Однораундовые коммуникационные вычисления

Определение 1.13 *Однораундовым (односторонним) коммуникационным протоколом вычисления булевой функции $f(x, y)$ называется алгоритм:*

1. *Вычислитель A получает на вход $\sigma \in \{0,1\}^n$, вычислитель B получает $\gamma \in \{0,1\}^n$.*

2. *Вычислитель A начинает вычисления: по σ определяет сообщение*
$$m^1(\sigma) = m^1 = m_1^1 m_2^1 \dots m_{t_1}^1 \in \{0,1\}^{t_1}, t_1 = t_1(\sigma)$$
и передает его B.

3. *Вычислитель B получает m^1 и по m^1 и γ выдаёт ответ — значение функции $f(\sigma, \gamma)$.*

Определение 1.14 *Односторонняя коммуникационная сложность функции f — это минимальная сложность однораундового протокола, вычисляющего f*
$$C_1(f) = \min_{\Phi \text{ вычисляет } f} C(\Phi).$$

Пусть $f(x,y)$ — произвольная булева функция, $CM(f)$ — коммуникационная матрица этой функции. Обозначим число различных строк матрицы $CM(f)$ через $nrow(CM(f))$.

Теорема 1.12 *Для любой булевой функции выполняется*

$$C_1(f) = \log nrow(CM(f)).$$

Доказательство: Для удобства рассуждений будем полагать, что $nrow(CM(f)) = 2^l$. Докажем, что $C_1 \leq \log nrow(CM(f))$. Рассмотрим коммуникационную матрицу, где $1, 2, \ldots, 2^l$ — группы, внутри каждой группы строки одинаковые (это возможно сделать, так как от перестановки строк матрица не изменится). Построим протокол Ф. Закодируем номер каждой группы:

$$\begin{aligned} &m_1 \to 1 \\ &m_2 \to 2 \\ &\vdots \\ &m_{2^l} \to 2^l \end{aligned}$$

Длина кода $\leq l$.

Вычислитель A определяет номер группы, в которую попал входной набор σ и передаёт закодированный номер вычислителю B. Вычислитель B получает на вход набор γ и, зная номер группы, выдает ответ $f(\sigma, \gamma)$. Сложность этого протокола Ф: $C_1(\Phi) \leq l$. Следовательно, $C_1(f) \leq l$.

Теперь докажем обратное неравенство $C_1(f) \geq l$. Покажем, что не существует протокола, сложности $< l$, вычисляющего функцию f. Воспользуемся принципом Дирихле и методом от противного. Предположим, что существует протокол Ф, такой, что $C_1(\Phi) < l$.

Протокол Ф может использовать $< 2^l$ различных сообщений. Следовательно, найдутся по крайней мере две строки σ и σ' коммуникационной матрицы, принадлежащие различным группам, для которых протокол использует одно и то же сообщение m. Но так σ и σ' принадлежат разным группам, то найдется такое γ, что $f(\sigma, \gamma) \neq f(\sigma', \gamma)$. Поскольку и для σ и для σ' вычислитель A передаёт одно и то же сообщение m, то вычислитель B будет выдавать один и тот же ответ и для (σ, γ) и для (σ', γ). На одном из этих

наборов выдаваемый ответ будет неверный. Получили противоречие с тем, что протокол верно вычисляет функцию. Следовательно, выполняется $C_1(f) \geq l$. $\square$

1.4.2 Сравнение трёхраундовых и одноpаундовых коммуникационных вычислений

Рассмотрим трёхраундовые коммуникационные вычисления.

Определение 1.15 *Трёхраундовым коммуникационным протоколом вычисления булевой функции $f(x, y)$ называется алгоритм:*

1. *Вычислитель A получает на вход $\sigma \in \{0,1\}^n$, вычислитель B получает $\gamma \in \{0,1\}^n$.*
2. *Вычислитель A начинает вычисления: по σ определяет сообщение*

$$m^1(\sigma) = m^1 = m_1^1 m_2^1 \dots m_{t_1}^1 \in \{0,1\}^{t_1}, t_1 = t_1(\sigma)$$

   и передает его B.

3. *Вычислитель B начинает вычисления: по сообщению m^1 и γ определяет сообщение*

$$m^2(\gamma, m^1) = m^2 = m_1^2 m_2^2 \dots m_{t_2}^2 \in \{0,1\}^{t_2}, t_2 = t_2(\gamma, m^1)$$

   и передает его A.

4. *Вычислитель A по сообщению m^2 и σ определяет сообщение*

$$m^3(\sigma, m^2) = m^3 = m_1^3 m_2^3 \dots m_{t_2}^3 \in \{0,1\}^{t_3}, t_3 = t_3(\sigma, m^2)$$

   и передает его B.

5. *Вычислитель B получает m^3 и по m^3, m^1 и γ выдаёт ответ — значение функции $f(\sigma, \gamma)$.*

Проверим даёт ли использование нескольких раундов передачи сообщений преимущество в коммуникационной сложности.

INDEX Рассмотрим функцию $INDEX(x,y) : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, определяемую следующим образом

$$INDEX(\sigma,\gamma) = \begin{cases} \sigma_i, \text{если в } \gamma \text{ только одна единица на } i\text{-ом месте,} \\ 0, \text{иначе.} \end{cases}$$

Теорема 1.13 $C_1(INDEX) = n$

Доказательство: Построим коммуникационную матрицу M для функции $INDEX$, увидим что $nrow(M) = 2^n$, так как для всякого набора σ строки будут различны, таким образом $C_1(ISA) \geq n$. Равенство получаем из теоремы о верхней оценки коммуникационной сложности для произвольной функции. □

Теорема 1.14 $C_3(ISA) \leq \log n + 1$

Доказательство: Для трёхраундового коммуникационного вычисления можно построить протокол Φ с коммуникационной сложностью $C_3(ISA) = \log n + 1$. Принцип работы протокола Φ следующий. На первом раунде вычислитель A передаёт пустое сообщение, затем вычислитель B просматривает входной набор γ, если в нём присутствует ровно одна единица, то вычислителю A передаётся номер этого символа, иначе выдается ответ 0. Вычислитель A, получив i, находит соответствующий бит своего входного набора и отправляет его значение σ_i вычислителю B. Таким образом сложность этого протокола равна $C(\Phi) = \log n + 1$, а значит коммуникационная сложность функции $C_3(ISA) \leq \log n + 1$. □

Теорема 1.15 $C(INDEX) \geq \log n$

Доказательство: Воспользуемся одним из методов доказательства нижней оценки коммуникационной сложности — методом полных множеств. Построим множество

$$S = \{(\sigma,\sigma) \,:\, \sigma = 0^j 10^{n-j-1}, j = 0,\ldots,n-1\}.$$

Очевидно, что для любых $(\sigma,\sigma) \in S$ выполняется $INDEX(\sigma,\sigma) = 1$, и для любых различных пар (σ,σ), $(\sigma',\sigma') \in S$ выполняется

$INDEX(\sigma, \sigma') = 0$ и $INDEX(\sigma', \sigma) = 0$, то есть множество S — полное множество для функции $INDEX$. Заметим, что $|S| = n$, откуда следует $C(INDEX) \geq \log n$. □

Далее покажем, что увеличение количества раундов, не всегда даёт улучшение в коммуникационной сложности. Рассмотрим уже известную функцию $EQ(x, y)$.

Теорема 1.16 $C_1(EQ) = C_t(EQ) = n$, *для любого* $t \geq 1$.

Глава 2

Недетерминированная коммуникационная модель

2.1 Определение недетерминированной модели

В данном разделе будем рассматривать следующую модель коммуникационных вычислений. Имеются два вычислителя A и B, им на вход подаются два входных набора σ и γ, соответственно. Вычислитель A, на основе входного набора строит множество возможных сообщений $M(\sigma) = \{m^1, \ldots, m^t\}$, которые можно отправить B, где $t = t(\sigma)$ также зависит от входа. Далее вычислитель выбирает сообщение из этого множества $m \in M(\sigma)$ и отправляет его B. Механизм выбора отправляемого сообщения недетерминирован (не определён). Вычислитель B, получив сообщение m и зная входной набор γ, если может выдать ответ, выдаёт значение функции. Иначе строит своё множество возможных сообщений $M(m, \gamma)$ и также недетерминированно выбирает сообщение $m' \in M(m, \sigma)$ и отправляет его вычислителю B. Вычисления продолжаются до тех пор пока B не сможет выдать ответ. В следующем определении рассмотрим односторонний недетерминированный протокол.

Определение 2.1 *Будем говорить, что протокол* Φ *недетерминированно вычисляет функцию* $f(\sigma, \gamma)$*, если:*

1. *для любого входного набора* (σ, γ)*, такого что* $f(\sigma, \gamma) = 1$*, существует такое сообщение* m_i^σ*, что вычислитель* A *передает* m_i^σ *вычислителю* B*, а* B *выдаёт единицу;*

2. *для любого входного набора* (σ, γ)*, такого что* $f(\sigma, \gamma) = 0$ *и*

для любого сообщения $m_i^\sigma \in \{m_1^\sigma ... m_t^\sigma\}$, которое вычислитель A передает B, вычислитель B всегда выдаёт 0.

Обозначим через Ф протокол, который недетерминированно вычисляет функцию f. Из определения получаем, что на входных наборах, на которых функция принимает значение 1, у протокола Ф существует такое сообщение, на котором протокол Ф выдаёт 1. А на тех входных наборах, на которых функция принимает значение 0, протокол Ф всегда выдаёт 0.

Далее введём понятие сложности недетерминированного коммуникационного протокола. На произвольном входном наборе (σ, γ) сложность протокола будет вычисляться следующим образом. Пусть $M(\sigma, \gamma) = \{m^1(\sigma, \gamma), m^2(\sigma, \gamma), \ldots, m^t(\sigma, \gamma), \}$ множество всех сообщений протокола Ф на этом входном наборе. Тогда сложность протокола для данного входа

$$C(\Phi(\sigma, \gamma)) = \max_{m \in M(\sigma,\gamma)} |m|$$

Определение 2.2 *Сложностью недетерминированного коммуникационного протокола* Ф *называется величина:*

$$C(\Phi) = \max_{\sigma,\gamma} C(\Phi(\sigma, \gamma))$$

Определение 2.3 *Недетерминированной сложностью функции f называется минимальная сложность протокола, недетерминированно вычисляющего f:*

$$NC(f) = \min_{\Phi} C(\Phi)$$

Теорема 2.1 *Для произвольной булевой функции $f(x, y)$ верно*

$$NC(f) = NC_1(f)$$

Доказательство: Неравенство $NC_1(f) \geq NC(f)$ выполняется, так как как односторонние коммуникационные вычисления это частный случай коммуникационных вычислений.

Докажем, что $NC_1(f) \leq NC(f)$. Для этого возьмем недетерминированный протокол Ф, который вычисляет функцию f с минимальной (наилучшей) сложностью, то есть $NC(f) = C(\Phi)$. По

этому протоколу Φ построим односторонний недетерминированный протокол Φ_1 с той же коммуникационной сложностью, $C(\Phi) = C(\Phi_1)$.

Пусть $M(\Phi) = \{m^1, m^2, \ldots, m^t\}$ — множество различных сообщений, используемых протоколом Φ. Так как Φ недетерминированно вычисляет функцию f, то для любых входов (σ, γ) для которых $f(\sigma, \gamma) = 1$ существует $m \in M(\Phi)$, при использовании которого ответ вычислителя B равен 1 и для любых входов (σ, γ) для которых $f(\sigma, \gamma) = 0$ для любых $m \in M(\Phi)$. Недетерминированный односторонний протокол Φ_1 будет использовать то же множество сообщений $M(\Phi)$. На первом шаге вычислитель A недетерминированно выбирает сообщение $m \in M(\Phi)$ и передаёт его вычислителю B, после чего B выдаёт ответ. Несложно показать, что Φ_1 верно вычисляет функцию f. При этом $C(\Phi_1) = C(\Phi)$. Поэтому $NC_1(f) \leq NC(f)$.

Объединяя оба доказанные неравенства, получаем утверждение теоремы $NC_1(f) = NC(f)$. □

Теперь рассмотрим примеры недетерминированных протоколов для некоторых булевых функций.

Теорема 2.2 $NC_1(NEQ) \leq \log n + 1$

Доказательство: Построим для недетерминированный коммуникационный протокол для функции NEQ:

1. Вычислитель A недетерминированно выбирает i-тый бит σ_i набора σ и пересылает B номер i и значение σ_i.

2. Вычислитель B сравнивает $\sigma_i = \gamma_i$, если они не равны, то выдает 1, иначе 0.

Проверим корректность данного протокола, правильно ли вычисляет наш протокол функцию NEQ в недетерминированном смысле. В соответствии с определением функции и недетерминированных коммуникационных вычислений:

1. $f(\sigma, \gamma) = 1 \Rightarrow$ существует m_i^σ, такое что на наборе γ вычислитель B выдаст 1. Действительно, если $NEQ(\sigma, \gamma) = 1$, то $\sigma \neq \gamma \Rightarrow \exists i \in [1, n] : \sigma_i \neq \gamma_i$. Если A пошлёт i, σ_i, тогда B выдаст правильный ответ.

2. $f(\sigma, \gamma) = 0 \Rightarrow$ для любого m_i^σ на наборе γ вычислитель B всегда выдаёт 0. Действительно, если $NEQ(\sigma, \gamma) = 0$, то $\sigma = \gamma \Rightarrow \forall i \in [1, n] : \sigma_i = \gamma_i$. Вычислитель B в этом случае всегда будет выдавать 0, так как $\delta = \gamma$.

Значит алгоритм корректен и недетерминированная сложность функции $NC(NEQ) \leq \log n + 1$. □

Ранее было доказано, что детерминированная сложность функции NEQ $C(NEQ) = n$. То есть недетерминизм уменьшает сложность коммуникационных вычислений. Однако это выполняется не для всех функций. Для того, чтобы это понять, нам необходимо уметь доказывать нижние оценки недетерминированной коммуникационной сложности функций.

2.2 Методы доказательства нижних оценок недетерминированной коммуникационной сложности

Отличие недетерминированной модели от детерминированной состоит в том, что в недетерминированном коммуникационном протоколе участники могут передавать различные сообщения для одного и того же входа, и ответ вычислителя B для конкретного входа может отличаться от значения функции на данном входе. Для доказательства нижних оценок для этой модели необходимы свои методы. В данном разделе мы рассмотрим некоторые из них.

2.2.1 Метод 1-полных множеств

Метод 1-полных множеств является аналогом метода полных множеств для детерминированной модели.

Определение 2.4 *1-полное множество (1-fooling set или $FS1_f$) для функции $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ — это множество входных наборов $S \subseteq \{0,1\}^n \times \{0,1\}^n$ такое, что*

1. *для любой пары $(\sigma, \gamma) \in S$, $f(\sigma, \gamma) = 1$,*
2. *для любых двух разных пар $(\sigma_1, \gamma_1), (\sigma_2, \gamma_2) \in S$, выполняется либо $f(\sigma_1, \gamma_2) \neq 0$, либо $f(\sigma_2, \gamma_1) \neq 0$.*

Теорема 2.3 *Для произвольной булевой функции f выполняется*

$$NC_1(f) \geq \log |S|,$$

где S — 1-полное множество для функции f.

Доказательство: Пусть $S = \{(\sigma, \gamma)\}$ — 1-полное множество для функции f, Φ — односторонний недетерминированный протокол минимальной сложности, вычисляющий f. Каждой паре $(\sigma, \gamma) \in S$ должно соответствовать своё коммуникационное сообщение. Предположим, это не так, и для двух различных пар (σ, γ), (σ', γ') в протоколе используется одно и то же сообщение m. Так как для всех пар $\in S$ значение функции равно 1, а для пар (σ, γ') и (σ', γ) значение функции равно 0, то на входах (σ, γ') и (σ', γ) протокол будет выдавать неверный ответ. Следовательно, $NC_1(\Phi) \geq \log |S|$. $\square$

2.2.2 Метод 1-прямоугольников

Ранее мы рассматривали метод доказательства нижней оценки детерминированной коммуникационной сложности произвольной булевой функции f, основанный на разбиении коммуникационной матрицы $CM(f)$ на одноцветные прямоугольники. В таком разбиении каждый элемент матрицы мог принадлежать ровно одному прямоугольнику. Для доказательства нижних оценок в недетерминированной модели может быть использован похожий метод, в котором вместо разбиения на прямоугольники используется покрытие матрицы монохромными прямоугольниками, при этом один и тот же элемент матрицы может принадлежать сразу нескольким прямоугольникам.

Пусть $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ — произвольная булева функция, $CM(f)$ — коммуникационная матрица функции f.

Определение 2.5 *$cov_1(f)$ – минимальное число 1-прямоугольников, которыми можно покрыть все единицы матрицы $CM(f)$ (возможно с перекрытиями); $cov_0(f)$ – минимальное число 0-прямоугольников, которыми можно покрыть все нули матрицы $CM(f)$ (возможно с перекрытиями).*

Свойство 1 *Для любой функции* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$

$$cov_1(f) \leq \chi(f).$$

Теорема 2.4 *Для произвольной функции* $f : \{0,1\}^n \times \{0,1\} \to \{0,1\}$ *выполняется*

$$NC_1(f) \leq \lceil \log cov_1(f) \rceil + 1,$$

$$NC(f) \geq \log(cov_1(f).$$

Доказательство: Покажем $NC_1(f) \leq \lceil \log cov_1(f) \rceil + 1$. Построим протокол, недетерминированно вычисляющий функцию f. Пусть (σ, γ) — входной набор, Алиса получает на вход σ, Боб получает на вход γ.

- Алиса недетерминированным образом выбирает номер 1-прямоугольника, которому принадлежит её набор σ и отправляет этот номер Бобу. Если не существует 1-прямоугольника, которому принадлежит σ, Алиса отправляет Бобу 0.
- Боб проверяет, принадлежит ли его входной набор γ тому 1-прямоугольнику, номер которого он получил от Алисы. Если принадлежит, Боб выдаёт ответ 1, в противном случае Боб выдаёт ответ 0.

Проверим корректность построенного протокола. Если входной набор (σ, γ) такой, что $f(\sigma, \gamma) = 1$, то (σ, γ) принадлежит хотя бы одному 1-прямоугольнику, и существует вариант вычисления, при котором Алиса выбирет именно этот прямоугольник, отошлёт его номер Бобу и Боб выдает верный ответ 1. Если вход (σ, γ) такой, что $f(\sigma, \gamma) = 0$, то не существует 1-прямоугольника, которому бы принадлежали одновременно σ и γ, следовательно, Боб в этом случае всегда выдаёт ответ 0. Сложность построенного протокола $NC_1(\Phi) = \lceil \log cov_1(f) \rceil + 1$.

Покажем $NC(f) \geq \log(cov_1(f)$. Пусть Φ — недетерминированный протокол минимальной сложности, вычисляющий функцию f и пусть Φ использует l различных сообщений, и следовательно сложность протокола $NC(\Phi) \geq \log l$.

Покажем, что l не может быть меньше $cov_1(f)$. Предположим это не так, и $l < cov_1(f)$. Тогда по принципу Дирихле, найдется сообщение m, которое будет соответствовать двум различным 1-прямоугольникам. Обозначим эти прямоугольники $A \times B$ и $A' \times B'$. Так как эти прямоугольники разные, то найдется пара (σ, γ) такая что $\sigma \in A$ и $\gamma \in B'$ либо $\sigma \in A'$ и $\gamma \in B$ и при этом $f(\sigma, \gamma) = 0$. На наборе (σ, γ) будет существовать вариант вычисления, при котором Боб выдаст ответ 1. Это означает что протокол неверно вычисляет функцию f. Следовательно, $l \geq cov_1(f)$ и значит $NC(f) \geq \log(cov_1(f)$. □

Рассмотрим пример функции, для которой $cov_1(f)$ значительно меньше $\chi(f)$. Проблема пересечения множеств (Set Intersection) формулируется следующим образом: для двух множеств $\subseteq [1, n]$ требуется определить, имеют ли эти множества непустое пересечение. На основе данной проблемы определим булеву функцию SI:

$$SI(x, y) = \bigvee_{i=1}^{n} (x_i \wedge y_i).$$

Утверждение 2.1

$$cov_1(SI) \leq n.$$
$$\chi(SI) \geq 2^n - 1.$$

Доказательство: 1-прямоугольники $A_i \times B_i$ $(i = 1, \ldots, n)$, где $\sigma \in A_i$ если $\sigma_i = 1$ и $\gamma \in B_i$ если $\gamma_i = 1$, покрывают все единицы матрицы $CM(SI)$. Следовательно, $cov_1(SI) \leq n$.

Мы знаем, что $C(f) \geq \chi(f) \geq rank(CM(f))$ для любой функции f. При этом $rank(CM(SI)) = 2^n - 1$, следовательно выполняется $\chi(SI) \geq 2^n - 1$. □

Теорема 2.5 *Недетерминированная сложность функции SI:*

$$NC_1(EQ) = \log n + 1$$

Теорема 2.6 *Недетерминированная сложность функции EQ:*

$$NC_1(EQ) \geq n$$

Теорема 2.7 *Недетерминированная сложность функции ISA:*

$$NC_1(INDEX) \geq \log n + 1$$

2.2.3 Классы сложности и отношения между ними

Обозначим через $F_n = \{f_n(x_1, \ldots, x_n, y_1, \ldots, y_n)\}$ множество булевых функций от $2n$ переменных. Определим несколько классов сложности булевых функций.

Определение 2.6 *Класс $P-CC_1 = \{f \in F_n$: существует детерминированный односторонний коммуникационный протокол Φ, вычисляющий f и имеющий сложность $C(\Phi) \leq (\log n)^k$, где $k \geq 0\}$.*

Определение 2.7 *Класс $NP-CC_1 = \{f \in F_n$: существует недетерминированный односторонний коммуникационный протокол Φ, вычисляющий f и имеющий сложность $NC(\Phi) \leq (\log n)^k$, где $k \geq 0\}$.*

Определение 2.8 *Класс $co-NP-CC_1 = \{g \in F_n$: отрицание этой функции $\neg g \in NP-CC_1\}$.*

Теорема 2.8

$$P-CC_1 \subset NP-CC_1$$

Доказательство: $P-CC_1 \subseteq NP-CC_1$, так как любой детерминированный коммуникационный протокол, есть частный случай недетерминированного коммуникационного протокола. С другой стороны в предыдущем разделе было показано, что $NEQ \in NP-CC_1 \setminus P-CC_1$. □

2.3 Обобщения модели $k-$ вычислителей

Существует несколько способов обобщить рассмотренную ранее коммуникационную модель на модели, в которую входит более двух вычислителей. Рассмотрим наиболее интересную: “number on the forehead”. Она заключается в математической головоломке, в ходе которой людей собирают в одной комнате, у каждого человека

на голове есть последовательность бит, которую могут видеть все остальные, а он сам не видит. Фактически: существует функция $f : (\{0,1\}^n)^k \to \{0,1\}$ и входной вектор $(x_1, x_2...x_k)$, где $x_i \in \{0,1\}^n$, i-ый игрок может видеть все x_j, $i \neq j$. Как и в случае двух игроков, у игроков есть фиксированный коммуникационный протокол, общение согласно которому основано на принципе "public blackboard". В завершении протокола все участники должны знать $f(x_1...x_k)$.

Пример 2.1 *Рассмотрим вычисление функции*

$$f(x_1, x_2, x_3) = \bigoplus_{i=1}^{n} maj(x_{1i}, x_{2i}, x_{3i})$$

в модели с тремя участниками, где x_1, x_2, x_3 вектора размерности n бит. Коммуникационная сложность этой функции =3: каждый игрок, читая число i, может определить то значение, которое составляет большинство из x_{1i}, x_{2i}, x_{3i}, рассматривая биты доступные ему. Он записывает $\oplus$ сумму этих чисел на доску, и конечный ответ это $\oplus$ из битов игроков. Этот протокол верен так как большинство из каждого ряда известно 1-ому или 3-ему игроку (для нечётного номера).

Пример 2.2 *Функция «inner product» (IP)*

$$IP_{n,k} = \bigoplus_{i=1}^{n} \bigwedge_{j=1}^{k} x_{ij} \tag{2.1}$$

Заметим, что при $= 2$ получим функцию IP.

В модели с двумя вычислителями мы ввели понятие монохромотических прямоугольников, чтобы доказать нижнюю оценку. Для модели с k участниками мы будем использовать цилиндрические пересечения.

Определение 2.9 *Цилиндр в i измерениях — это множество S входных значений, таких что если $(x_1...x_k) \in S$, тогда для всех x_i мы получим $(x_1, ..., x_{i-1}, x_i, x_{i+1}, ..., x_k)$ также $\in S$.*

Определение 2.10 *Цилиндрическое пересечение — это $\bigcap_{i=1}^{k} T_i$, где T_i — цилиндр в i измерениях.*

Как было замечено в случае двух вычислителей, коммуникационный протокол может быть рассмотрен как процесс разбиения матрицы $CM(f)$ на монохроматические прямоугольники. Для модели с k участниками $CM(f)$ это $k-$размерный куб, и сообщение, переданное i-ым участником, не зависит от x_i. Тем не менее можно показать, что если у f есть "multiparty" протокол, который передает c бит, тогда у $CM(f)$ есть разбиение, которое использует не более 2^c монохроматических цилиндрических пересечений.

Лемма 2.1 *Если любое разбиение $CM(f)$ на монохроматические цилиндрические пересечения требует не менее R цилиндрических пересечений, то тогда $k-$вычислительная коммуникационная сложность $\geq \log_2 R$.*

Глава 3

Вероятностная коммуникационная модель

3.1 Определение модели

В данном разделе будем рассматривать следующую модель коммуникационных вычислений. Даны два вычислителя A и B, на вход которым подаются два входных набора σ и γ, соответственно. Вычислитель A, на основе входного набора строит множество возможных сообщений $M(\sigma) = \{m^1, \ldots, m^t\}$, которые можно отправить B, где $t = t(\sigma)$ также зависит от входа. Далее вычислитель выбирает сообщение из этого множества $m \in M(\sigma)$ и отправляет его B. Механизм выбора отправляемого сообщения вероятностный, то есть имеется генератор случайных чисел, и вычислитель A, в соответствии с полученной случайной величиной, выбирает одно из сообщений. Вычислитель B, получив сообщение m и зная входной набор γ, если может выдать ответ, выдаёт значение функции. Иначе строит своё множество возможных сообщений $M(m, \gamma) = \ldots$ и также вероятностно выбирает сообщение $m' \in M(m, \sigma)$, затем отправляет его вычислителю B. Вычисления продолжаются до тех пор, пока B не сможет выдать ответ. Если для обоих вычислителей используется общий генератор случайных чисел, тогда эта модель называется вероятностными коммуникационными вычислениями с открытым ключом “public coin”. Если у каждого вычислителя свой генератор, то эта модель называется коммуникационной моделью с закрытым ключом “private coin”.

Определение 3.1 *Будем говорить, что протокол* Φ *вычисляет функцию* $f(\sigma,\gamma)$ *с неограниченной ошибкой (с вероятностью* $\frac{1}{2}$*), если выполняется следующее:*

1. *Для любого входного набора* (σ,γ)*:* $f(\sigma,\gamma)=1$*, вероятность принять входной набор* $\Pr^{\Phi}_{accept}(\delta,\gamma)>\frac{1}{2}$*,*
2. *Для любого входного набора* (σ,γ)*:* $f(\sigma,\gamma)=0$*, вероятность принять входной набор* $\Pr^{\Phi}_{accept}(\delta,\gamma)\leq\frac{1}{2}$*.*

Определение 3.2 *Будем говорить, что протокол* Φ *вычисляет функцию* $f(\sigma,\gamma)$ *с ограниченной ошибкой (с вероятностью* $\frac{1}{2}+\varepsilon$*), если существует такое* $\varepsilon\in\left(0,\frac{1}{2}\right]$*, что:*

1. *Для любого входного набора* (σ,γ)*:* $f(\sigma,\gamma)=1$*, вероятность принять входной набор* $\Pr^{\Phi}_{accept}(\delta,\gamma)\geq\frac{1}{2}+\varepsilon$*,*
2. *Для любого входного набора* (σ,γ)*:* $f(\sigma,\gamma)=0$*, вероятность принять входной набор* $\Pr^{\Phi}_{accept}(\delta,\gamma)\leq\frac{1}{2}+\varepsilon$*.*

Определение 3.3 *Вероятностной коммуникационной сложностью функции* $f(\sigma,\gamma)$ *называется величина*

$$PC(f)=\min C(\Phi)$$

сложность наилучшего протокола, который вычисляет функцию $f(x,y)$ *с вероятностью* $\frac{1}{2}$*,*

$$RC(f)=\min C(\Phi)$$

сложность наилучшего протокола, который вычисляет функцию $f(x,y)$ *с вероятностью* $\frac{1}{2}+\varepsilon$*.*

Свойство 2

$$C(f)\geq RC_{\frac{1}{2}+\varepsilon}(f)\geq PC_{\frac{1}{2}}(f)$$

3.1.1 Вероятностная коммуникационная сложность функции «Равенство»

Рассмотрим пример вероятностного протокола для известной нам функции EQ. Построим вероятностный протокол Φ, который вычисляет функцию $EQ(x,y)$ с большой вероятностью правильного ответа, используя при этом $O(\log n)$ битов при коммуникации.

Идея построения подобного вероятностного алгоритма принадлежит Р.В. Фрейвалду.

Входы трактуются как два натуральных числа x и y, $0 \leq x, y \leq 2^n - 1$. Вычислитель A выбирает (равновероятностно) простое число $p \leq n^2$, вычисляет $x' = x \pmod{p}$ и пересылает пару (x', p) вычислителю B. Вычислитель B вычисляет $y' = y \pmod{p}$ и сравнивает y' с x'. Если $y' \neq x'$, то вычислитель B выдает ответ $x \neq y$. Если $y' = x'$, то вычислитель B выдает ответ $x = y$.

Теперь посчитаем вероятности возможных ошибок.

- Если два числа x, y равны, то очевидно x', y' также равны, и протокол Ф выдаст правильный ответ.
- Если два числа x, y различны, то тем не менее может оказаться, что $y' = x'$, и протокол Ф выдаст в этом случае неверный ответ. Это может произойти, если p является делителем числа $x - y$.

Заметим, что $|x - y| < 2^n$, следовательно $x - y$ может иметь не более n различных простых делителей. С другой стороны вычислитель P_x выбирает простые числа среди $O(\frac{n^2}{\log n})$ простых чисел, таким образом вероятность выбора делителя числа $x - y$ очень мала.

Таким образом, доказали следующее утверждение:

Утверждение 3.1 $RC(EQ) = O(\log n)$

3.2 Сравнение моделей “public coin” и “private coin”

В определении вероятностной коммуникационной модели “private coin” вычислители Алиса и Боб по определению имеют свои датчики случайных чисел и следовательно, каждый использует своё вероятностное распределение случайных чисел. Алиса и Боб не видят случайные распределения друг друга. В модели “public coin” Алиса и Боб имеют общий датчик случайных чисел и общее вероятностное распределение случайных чисел. Более формально, существует вероятностная строка r (выбираемая в соответствии с распределением вероятностей П). При этом выбор передаваемого сообщения от Алисы к Бобу зависит от входного набора σ Алисы и вероятностного набора r, выбор передаваемого сообщения от Боба к Алисе зависит от входного набора γ Боба и вероятностного набора r.

Мы можем также смотреть на данную модель как на распределение $\{P_r\}_{r\in\Pi}$ детерминированных протоколов. Алиса и Боб совместно выбирают строку r (это выбор зависит только от распределения вероятностей Π и не зависит от значений σ и γ) и затем следуют детерминированному протоколу Φ_r.

Определение 3.4 *Вероятностный протокол Φ^{pub} с общим датчиком случайных чисел для функции $f(X,Y)$ — это вероятностное распределение над детерминированными протоколами $\{\Phi_r\}_{r\in\Pi}$. Вероятность правильного результата на входе (σ,γ) — это есть вероятность выбора детерминированного протокола Φ_r (основываясь на распределении вероятностей Π), такого, что Φ_r правильно вычисляет функцию f на наборе (σ,γ).*

Определение 3.5 *$RC^{public}(f)$ — сложность наилучшего протокола с общим датчиком случайных чисел, вычисляющего функцию f с ограниченной ошибкой.*

Покажем, что произвольный "private coin" протокол может быть промоделирован как "public coin" протокол.

Утверждение 3.2

$$RC^{public}(f) \le RC(f)$$

Доказательство: Пусть Φ — произвольный "private coin" протокол и пусть r_A и r_B — последовательности случайных чисел Алисы и Боба (каждая последовательность выбрана опираясь на собственное распределение и независимо друг друг от друга). Данный протокол может быть промоделирован "public coin" протоколом, в котором общая строка случайных чисел r образована как конкатенация строк r_A и r_B и рассматривается как public строка. □

Теорема 3.1

$$RC_1^{pub}(EQ) = O(1)$$

Доказательство: Опишем следующий "public coin" протокол для функции EQ на наборе σ,γ.

1. Алиса и Боб совместно выбирают случайную строкуr, состоящую из n бит.
2. Алиса вычисляет скалярное произведение $b = \langle \sigma, r \rangle$ по модулю 2 и передаёт вычисленное значение (один бит) Бобу.
3. Боб проверяет равенство $b = \langle \gamma, r \rangle$ и выдает ответ 1, если $b = \langle \gamma, r \rangle$, и 0, если $b \neq \langle \gamma, r \rangle$.

Очевидно, что если выполняется $\sigma = \gamma$, то ответ всегда 1 и протокол не ошибается.

Пусть $\sigma \neq \gamma$. Оценим вероятность ошибки в этом случае. Поскольку $\sigma \neq \gamma$, то существует позиция i такая что $\sigma_i \neq \gamma_i$. Пусть для определенности $\sigma_i = 1$, $\gamma_i = 0$. Существует 2^{n-1} наборов r длины n, в которых на позициях $j \neq i$ могут стоять произвольные значения, а значение r_i выбрано таком образом, чтобы обеспечить $\langle \sigma, r \rangle \neq \langle \gamma, r \rangle$. Поэтому

$$Pr_r[\langle \sigma, r \rangle \neq \langle \gamma, r \rangle] = 1/2,$$

и поэтому Боб выдает правильный ответ 0 с вероятностью $1/2$ и с вероятностью $1/2$ ошибается. Повторяя эту процедуру дважды (с двумя независимыми выборами r) и выдавая ответ 0 только в том случае, если оба повтора выдадут ответ 0, вероятность ошибки уменьшается до $1/4$. Применяя данную технику мы можем понизить вероятность ошибки до сколь угодно малого значения ϵ. Таким образом, мы показали, что $RC_1^{public}(EQ) = O(1)$. □

Теорема 3.1 показывает, что разрыв между $C(f)$ и $RC^{public}(f)$ может быть произвольно большим. В то же время мы знаем, что для модели "private coin" выполняется $RC(EQ) \leq O(\log n)$. Следовательно, данный пример демонстрирует, что "public coin" может быть лучше, чем "private coin". Далее мы покажем, что преимущество модели "public coin" над моделью "private coin", полученное для функции EQ — наибольшее, которое может быть достигнуто. То есть "public coin" протокол может быть преобразован в "private coin"протокол с небольшим увеличением сложности протокола вероятности ошибки вычисления. При этом полученный протокол также будет вычислять функцию f с ограниченной ошибкой.

Теорема 3.2 *Пусть $f(X,Y)$ — булева функция. Для любого $\delta > 0$ и любого $\epsilon > 0$ выполняется следующее:*

$$RC_{\epsilon+\delta}(f) \leq RC^{pub}_{\epsilon}(f) + O(\log n + \log(\frac{1}{\delta})).$$

Доказательство: Нам достаточно доказать, что произвольный "public coin" протокол Φ, использующий произвольное число вероятностных бит, может быть преобразован в другой "public coin" протокол Φ' такой, что протокол Φ', использующий только $O(\log n + \log \frac{1}{\delta})$ вероятностных бит, и вероятность ошибки протокола Φ' увеличится не более чем в δ раз. Если мы докажем это, то доказательство Теоремы будет следовать из того, что Алиса может произвести это количество вероятностных бит сама и передать их Бобу, и далее вычислители просто следуют "public" протоколу Φ'.

Пусть Φ — "public coin" протокол, использующий произвольное число вероятностных бит. Обозначим через $Z(\sigma, \gamma, r)$ — вероятностную переменную, которая принимает значение 1, если протокол Φ выдаёт на наборе σ, γ и вероятностной строке r неверный ответ (отличный от значения $f(\sigma, \gamma)$). Соответственно, $Z(\sigma, \gamma, r)$ принимает значение 0, если протокол Φ на наборе σ, γ и вероятностной строке r выдаёт правильный ответ (равный $f(\sigma, \gamma)$). Так как Φ вычисляет $f(X,Y)$ с ошибкой ϵ, то выполняется $E_{r \in \Pi}[Z(\sigma, \gamma, r)] \leq \epsilon$ для любых входов $(\sigma, \gamma) \in \{0,1\}^{2n}$.

Будем строить новый "public coin" протокол Φ', который использует меньшее число вероятностных бит, используя «вероятностный метод».

Доказательство при помощи вероятностного метода. Идея «вероятностного метода» доказательства следующая. Пусть $R = \{r_1, \ldots, r_d\}$ — множество вероятностных строк, используемых в протоколе Φ. Для каждого входного набора (σ, γ) существует подмножество случайных строк $R(\sigma, \gamma) \subseteq R$, при использовании которых протокол не ошибается. При этом для каждого (σ, γ) это подмножество $R(\sigma, \gamma)$ своё. Назовем его «подходящим множеством» для входа (σ, γ). Надо показать, что существует подмножество $R' \subseteq R$, которое является «подходящим» для всех (σ, γ). Мы докажем, что ве-

роятность существования такого «подходящего множества» не равна нулю. Тем самым доказывается, что такое множество существует.

Для доказательства нам понадобится оценка Чернова.

Оценка Чернова. Пусть $X_1, \ldots, X_t$ — независимые случайные величины, принимающие значения из множества $\{0, 1\}$ с вероятностью $Pr(X_i = 1) = p, Pr(X_i = 0) = 1 - p$ для любого $i = 1, \ldots, t$. Тогда для любого $\delta \geq 0$ выполняется

$$Pr(|\frac{1}{t}\sum_{k=1}^{t} X_k - p| \geq \delta) \leq 2e^{-2t\delta^2}.$$

Зафиксируем параметр t (значение t определим позднее). Пусть $R = \{r_1, \ldots, r_d\}$ — множество всех вероятностных строк, используемых в протоколе Ф. Определим "public coin" протокол Φ_R следующим образом:

1. На входном наборе (σ, γ) Алиса и Боб равновероятно выбирают значение i $(1 \leq i \leq t)$ и запускают вычисления согласно протоколу Ф, используя r_i в качестве вероятностной строки.

Далее необходимо показать, что существует «подходящее» множество $R_t = \{r_1, \ldots, r_t\}$ такое, что

$$E_i[Z(\sigma, \gamma, r_i)] \leq \epsilon + \delta$$

для **любых** входов $(\sigma, \gamma) \in \{0, 1\}^{2n}$. Для такого выбора множества R_t построенный протокол Φ_R и будет искомым протоколом Φ'.

Чтобы построить такое «подходящее» множество R_t будем выбирать t значений $r_1, \ldots, r_t$ вероятностно (в соответствии с вероятностным распределением П).

Рассмотрим **фиксированную** пару σ, γ и посчитаем вероятность того, что $E_i[Z(\sigma, \gamma, r_i)] > \epsilon + \delta$ (где i имеет равномерное распределение) (то есть множество R для набора (σ, γ) — «плохое»). Это есть в точности вероятность того, что $\frac{1}{t}\sum_{i=1}^{t} Z(\sigma, \gamma, r_i) > \epsilon + \delta$.

Согласно неравенству Чернова, так как $E_r[Z(\sigma, \gamma, r)] \leq \epsilon$, то

$$Pr_R[(\frac{1}{t}\sum_{i=1}^{t} Z(\sigma, \gamma, r_i) - \epsilon) > \delta] \leq 2e^{-2\delta^2 t}.$$

При $t = O(n/\delta^2)$ это значение строго меньше чем 2^{-2n}. Поэтому для конкретного вероятностного выбора $R = \{r_1, \ldots, r_t\}$ вероятность того, что множество R окажется «неподходящим» хотя бы для одного входа (σ, γ) , то есть $E_i[Z(\sigma, \gamma, r_i)] > \epsilon + \delta$, строго меньше чем $2^n 2^{-n} = 1$. Следовательно, существует выбор множества R, которое является «подходящим» для всех наборов $(\sigma, \gamma) \in \{0,1\}^{2n}$.

Отметим теперь, что число вероятностных бит, используемых протоколом Φ' равно $\log t = O(\log n + \log \frac{1}{\delta})$. □

3.3 Методы доказательства нижних оценок для вероятностной коммуникационной модели

Проблема доказательства нижних оценок для вероятностных моделей требует новой техники. В случае детерминированных протоколов методом доказательства существования функций с высокой (асимптотически линейной) коммуникационной сложностью мог служить метод (неконструктивный) подсчета. Данный метод основан на следующем. Существует всего $2^{2^{2n}}$ различных булевских функций $f(X, Y), X, Y \in \{0,1\}^n$. С другой стороны, существует только $2^{2^{O(l)}}$ различных детерминированных протоколов длины l.

Если же мы имеем дело с вероятностной моделью, то существует несчётно много вероятностных протоколов длины l, так как вероятности могут быть произвольны. В случае ограниченных вероятностных вычислений (вычислений с ограниченной ошибкой) доказательство как линейных, так и логарифмических нижних оценок существенным образом используют в качестве аргумента факт, что ошибка вычисления ограничена константой.

3.3.1 Топологический метод

В данном разделе рассмотрим метод доказательства нижней оценки вероятностной коммуникационной сложности для вычисления с ограниченной ошибкой.

Теорема 3.3 *Пусть* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ — *произвольная булева функция,* $\varepsilon \in (0, 1/2]$, $p = 1/2 + \varepsilon$. *Тогда выполняется*

$$RC_p(f) \geq \log(DC(f) - 1) - \log\log(1 + 1/\varepsilon).$$

Введем необходимые понятия из теории множеств и метрических пространств.

Пусть $\mathcal{S}$ — это метрическое пространство с метрикой ρ. Конечное множество элементов $s_1, s_2, \ldots, s_t$ пространства $\mathcal{S}$ называется ε-цепью, если $\rho(s_i, s_{i+1}) < \varepsilon$ для $i \in \{1, 2, \ldots, t-1\}$. Говорят, что элементы s_1 и s_t соединимы ε-цепью.

Подмножество $\mathcal{C}_\varepsilon$ пространства $\mathcal{S}$ называется ε-компонентой пространства $\mathcal{S}$, если два любых элемента $s, s' \in \mathcal{C}_\varepsilon$ соединимы ε-цепью. Метрическое подпространство $\mathcal{S}'$ пространства $\mathcal{S}$ называется ограниченным, если существует такая константа c, что для произвольных двух элементов $s, s' \in \mathcal{S}'$ выполняется $\rho(s, s') \leq c$. Для произвольного $\varepsilon > 0$ ограниченное подпространство конечномерного векторного пространства разбивается на конечное число своих ε-компонент.

В d-мерном векторном пространстве R^d определим метрику ρ следующим образом. Для элементов $\mu = (p_1, p_2, ..., p_d)$ и $\mu' = (p'_1, p'_2, ..., p'_d)$ пространства R^d положим

$$\rho(\mu, \mu') = \sum_{i=1}^{d} \mid p_i - p'_i \mid .$$

Пусть $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ — произвольная булева функция, а Φ — односторонний вероятностный протокол, p-вычисляющий функцию f, $p = 1/2 + \varepsilon$, $\varepsilon \in (0, 1/2]$. Пусть $dim(\Phi) = d$. Обозначим R^d_Φ подпространство пространства R^d, состоящее из всевозможных распределений вероятностей сообщений протокола Φ.

$$R^d_\Phi = \left\{\mu \in R^d \;/\; \mu = \mu(x), x \in \{0,1\}^n\right\}.$$

По определению, метрическое подпространство R^d_Φ ограничено.

Мы докажем нижнюю оценку для p-коммуникационной размерности булевой функции, которая связана с величиной p-коммуникационной сложности.

Теорема 3.4 *Пусть* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ — *произвольная булева функция,* $\varepsilon \in \left(0, \frac{1}{2}\right]$, $p = 1/2 + \varepsilon$. *Тогда выполняется*

$$dim_p(f) \geq \frac{\log dim(f)}{\log(1 + 1/\varepsilon)}$$

Доказательство: Пусть Ф — это произвольный односторонний вероятностный протокол, p-вычисляющий функцию f. Для произвольных двух слов x, x' из множества представителей X функции f точки $\mu(x) = \{p_1(x), p_2(x), \ldots, p_d(x)\}$ и $\mu(x') = \{p_1(x'), p_2(x'), \ldots, p_d(x')\}$ принадлежат различным 2ε-компонентам пространства R^d_Φ.

Действительно, предположим, что существует 2ε-компонента $\mathcal{C}_{2\varepsilon} \in R^d_\Phi$ такая, что $\mu(x), \mu(x') \in \mathcal{C}_\varepsilon$. Положим $x_1 = x$, а $x_2 = x'$. Пусть точки $\mu(x_1), \mu(x_2), \ldots, \mu(x_t)$ образуют 2ε-цепь. Последнее означает, что для $i \in \{1, 2, \ldots, t-1\}$ выполняется

$$\rho(\mu(x_i), \mu(x_{i+1})) < 2\varepsilon. \tag{3.1}$$

Применяя последнее неравенство, получаем, что для произвольного слова y из множества Y тест функции f выполняется

$$\begin{aligned} \mu(x)\nu(y) - \mu(x')\nu(y) &\leq \sum_{i=1}^{d} \mid p_i(x) - p_i(x') \mid q_i(y) \\ &\leq \rho(\mu(x), \mu(x')) < 2\varepsilon. \end{aligned} \tag{3.2}$$

В силу условия теоремы 3.3 вероятностный протокол имеет надежность ε, поэтому для произвольных последовательностей $u, v \in \{0,1\}^n$ выполняется либо $\mu(u)\nu(v) \geq \frac{1}{2} + \varepsilon$, либо $\mu(u)\nu(v) \leq \frac{1}{2} + \varepsilon$. Содержательно $\mu(u)\nu(v)$ — это вероятность выдачи 1 протоколом Ф на входе uv.

Из определения вероятностного коммуникационного протокола и соотношений следует, что $f(x, y) = f(x', y)$ для всех последовательностей $y \in \{0,1\}^n$. Это противоречит тому, что $x, x' \in X$.

Оценим теперь число K 2ε-компонент пространства R^d_f. Нам достаточно оценить K следующим образом. В каждую 2ε-компоненту $\mathcal{C}_{2\varepsilon}$ пространства R^d_f поместим сферу радиуса ϵ с центром в соответствующей точке $\mu(x)$, $x \in X$. Все эти сферы могут пересекаться разве что по границе. Пространство R^d_f вместе со своими

сферами радиуса ϵ поместим в большую сферу радиуса $1+\epsilon$ с центром в точке $(0, 0, \ldots, 0)$.

Объем сферы радиуса r в пространстве R^d равен cr^d, где константа c зависит от используемой метрики ρ. Таким образом справедлива оценка

$$K \leq \frac{c(1+\epsilon)^d}{c\epsilon^d} = \left(1+\frac{1}{\epsilon}\right)^d.$$

Так как $K \geq dim(f)$, то теорема доказана. □

Так как для произвольной функции f ее вероятностная p-размерность $dim_p(f)$ связана с величиной $PC_p(f)$ p-коммуникационной сложности соотношением $2^{PC_p(f)} \geq dim_p(f)$, то из теоремы и того, что $\log dim(f) > DC(f) - 1$ следует утверждение теоремы.

3.3.2 Геометрический метод

Теорема 3.5 *Пусть* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *— произвольная булева функция,* $p = 1/2$. *Тогда выполняется*

$$PC_p(f) \geq \log(DC(f) - 1) - \log\log ts(f).$$

В последующей части излагается доказательство теоремы, которое состоит из двух утверждений. Введем необходимые определения и обозначения.

Обозначим $\mathbf{R}^d$ d-мерное Евклидово пространство. $(d-1)$-мерная гиперплоскость

$$a_1z_1 + a_2z_2 + \cdots + a_dz_d = b$$

делит $\mathbf{R}^d$ на две связные области. Условимся, что эти две связные области определяются следующими двумя неравенствами

$$a_1z_1 + a_2z_2 + \cdots + a_dz_d > b,$$

$$a_1z_1 + a_2z_2 + \cdots + a_dz_d \leq b.$$

Обозначим $k(d,t)$ максимальное число связных областей, которые могут быть образованы в d-мерном Евклидовом пространстве $\mathbf{R}^d$ t различными $(d-1)$-мерными гиперплоскостями. Следующий факт установлен О.Б.Лупановым.

Лемма 3.1 *Если* $d = 1$ *и* $t \geq 1$, *тогда* $k(d,t) = t+1$. *Если* $d \geq 2$ *и* $t \geq 2$, *тогда* $k(d,t) \leq t^d$.

Для полноты изложения приведём доказательство леммы.

Доказательство: Если $d = 1$, то очевидно, что прямая разбивается t различными точками (0-мерными гиперплоскостями) на $k(d,t) = t+1$ частей.

Доказательство второй части проведем методом индукции по числу t гиперплоскостей. Легко видеть, что для произвольного $d \geq 2$ выполняется $k(d,2) \leq 2^d$.

Пусть $t > 2$ и $d > 2$. Рассмотрим t произвольных $(d-1)$-мерных гиперплоскости α_1, α_2, $\ldots$, α_t. $t-1$ гиперплоскость α_1, α_2, $\ldots$, α_{t-1} могут определить в пространстве $\mathbf{R}^d$ самое большее $k(d,t-1)$ различных связных областей. Обозначим эти области через $D_1, D_2, \ldots, D_l$ $(l \leq k(d,t-1))$.

Гиперплоскость α_t можно рассматривать как $(d-1)$-мерное Евклидово пространство. Число связных областей в пространстве α_t, определяемое гиперплоскостями α_1, α_2, $\ldots$, α_{t-1} равно числу связных областей в пространстве α_t, образованное пересечением α_t с гиперплоскостями α_1, α_2, $\ldots$, α_{t-1} (каждое такое пересечение является $(d-2)$-мерной гиперплоскостью). Следовательно это число не превосходит $k(d-1,t-1)$.

Каждое из этих связных областей пространства α_t лежит в некоторой области D_i. Следовательно

$$k(d,t) \leq k(d,t-1) + k(d-1,t-1).$$

По предположению индукции для произвольного $d \geq 2$ имеем $k(d,t-1) \leq (t-1)^d$. Поэтому, для произвольного $d \geq 3$ выполняется

$$k(d,t) \leq (t-1)^d + (t-1)^{d-1} \leq t^d.$$

В случае $d = 2$ мы имеем $k(d,t) \leq (t-1)^2 + (t-1) + 1 \leq t^2 = t^d$. Таким образом для $d \geq 2$ мы получаем $k(d,t) \leq t^d$. $\square$

3.4 Вероятностные коммуникационные вычисления с неограниченной ошибкой

Неограниченная вероятностная модель не является основой для теории «разумной, используемой в практике передачи информации». Скорее, мы заинтересованы в понимании мощи неограниченных вероятностных вычислений для сопоставления её с ограниченными вероятностными вычислениями. Сравнение показывает, что эта мощь значительна.

Рассмотрим для примера функцию равенства $EQ(X, Y)$. Ранее было показано

$$C(EQ) \geq n,$$
$$RC(EQ) \leq O(\log n).$$
$$PC(EQ) = O(1).$$

Возникает естественный вопрос, не является ли модель неограниченных вероятностных вычислений слишком тривиальной и не является ли оценка вида $O(1)$ общей оценкой для всех функций. Как мы увидим далее, это не так.

3.4.1 Нижняя оценка. Метод гиперплоскостей

Метод гиперплоскостей основан на геометрических свойствах многомерных пространств.

Гиперплоскость h в R^d характеризуется при помощи вектора $\tilde{a} = (a_1, \ldots, a_{d+1}) \in R^{d+1}$. Точка $b \in R^d$ принадлежит гиперплоскости h, если

$$\langle (a_1, \ldots, a_d), b \rangle = a_{d+1}.$$

Гиперплоскость $(a_1, \ldots, a_d)$ — гиперплоскость, проходящая через начало координат.

Определение 3.6 *Размещением $Arr(H)$ называется конечное множество гиперплоскостей $H = \{h_1, \ldots, h_m\}$ в пространстве R^d для некоторого d. Данные гиперплоскости делят пространство R^d на* области. *Область размещения $Arr(H)$ — это непустая связанная компонента в R^d, образованная гиперплоскостями $Arr(H)$.*

Поскольку каждая гиперплоскость делит пространство R^d на два полупространства — положительное и отрицательное, то каждая область r размещения $Arr(H)$ может быть однозначно характеризована m-битной строкой $s = s_1, \ldots, s_m$, где $s_i = 1$, если область r находится в положительном полупространстве относительно гиперплоскости h_i, и $s_i = 0$, если область r находится в отрицательном полупространстве относительно гиперплоскости h_i. Будем называть строку s *сигнатурой* области r.

Будем говорить, что размещение $Arr(H)$ реализует множество $S_H \subseteq \{0,1\}^m$ сигнатур, если $S_H = \{w \in \{0,1\}^m \mid w$ есть сигнатура некоторой области r в $Arr(H)\}$.

Будем называть каждое $w \in \{0,1\}^m$ *требованием*. Требование $w \in \{0,1\}^m$ удовлетворяет размещению $Arr(H)$ в R^d для некоторого d, если $w \in S_H$. Будем говорить, что булевская матрица M порядка $k \times m$ удовлетворяет размещению $Arr(H)$ m гиперплоскостей H в R^d, если каждая строка матрицы M может быть рассмотрена как требование, принадлежащее S_H.

Теорема 3.6 *Пусть M — коммуникационная матрица функции f. Пусть d — наименьшая размерность пространства, в котором существует размещение $Arr(H)$ 2^n гиперплоскостей, которое удовлетворяет матрице M. Тогда выполняется*

$$\log d \leq PC(f) \leq \log d + 1.$$

$$PC(f) = O(\log d).$$

Доказательство: Будем называть длиной протокола Φ количество различных сообщений, используемых Φ . Ясно, что если длина протокола d, то $PC(\Phi) = \log d$.

Доказательство в одну сторону: Пусть для функции f существует протокол, вычисляющий f с неограниченной ошибкой и имеющий длину d. Покажем, что в этом случае существует размещение $Arr(H)$ в пространстве R^d, удовлетворяющее $CM(f)$.

Предположим для простоты, что все строки коммуникационной матрицы $CM(f)$ различны. Пусть Φ — протокол, вычисляющий f с неограниченной ошибкой и имеющий длину d и пусть $M =$

$\{m_1, \ldots, m_d\}$ — множество всех различных сообщений, используемых протоколом Φ . Обозначим $\mu(\sigma) = (p_1, \ldots, p_d)$ — вероятностный вектор, где p_i — вероятность того, что Алиса на входе σ передаст Бобу сообщение m_i. Ясно, что выполняется $p_1 + \cdots + p_d = 1$. Обозначим $\nu(\gamma) = (q_1, \ldots, d_d)$, $0 \leq q_I \leq 1$ — вектор, такой, что q_i — вероятность того, что Боб при получении от Алисы сообщения m_i и имея на входе набор γ, выдаст ответ 1. Поскольку Φ вычисляет функцию f с неограниченной ошибкой, то выполняется следующее:

- для любого входного набора $(\sigma, \gamma) \in \{0,1\}^{2n}$: $f(\sigma, \gamma) = 1$ выполняется $\langle \mu(\sigma)\nu(\gamma) \rangle > 1/2$;
- для любого входного набора $(\sigma, \gamma) \in \{0,1\}^{2n}$: $f(\sigma, \gamma) = 0$ выполняется $\langle \mu(\sigma)\nu(\gamma) \rangle < 1/2$.

Образуем новый вектор $\nu'(\gamma) = \nu(\gamma) - (1/2, \ldots, 1/2)$. Тогда $\mu(\sigma)\nu'(\gamma) = \mu(\sigma)(\nu(\gamma) - (1/2, \ldots, 1/2)) = \mu(\sigma)\nu(\gamma) - 1/2$. Поэтому верно следующее:

- для любого входного набора $(\sigma, \gamma) \in \{0,1\}^{2n}$: $f(\sigma, \gamma) = 1$ выполняется $\langle \mu(\sigma)\nu'(\gamma) \rangle > 0$,
- для любого входного набора $(\sigma, \gamma) \in \{0,1\}^{2n}$: $f(\sigma, \gamma) = 0$ выполняется $\langle \mu(\sigma)\nu'(\gamma) \rangle < 0$.

Вектора $\mu(\sigma)$, $\nu'(\gamma)$ можно трактовать

1. как точки в пространстве R^d,
2. как гиперплоскости в R^d, содержащие начало координат.

Точка $\mu(\sigma)$ лежит в положительном полупространстве относительно гиперплоскости $\nu'(\gamma)$, если выполняется $\langle \mu(\sigma), \nu'(\gamma) \rangle > 0$, то есть если $f(\sigma, \gamma) = 1$. Аналогично, точка $\mu(\sigma)$ лежит в отрицательном полупространстве относительно гиперплоскости $\nu'(\gamma)$, если выполняется $\langle \mu(\sigma), \nu'(\gamma) \rangle < 0$, то есть если $f(\sigma, \gamma) = 0$.

Пусть H — множество 2^n полупространств, P — множество 2^n точек. Легко видеть, что для любого σ сигнатура области r — это есть σ-строка матрицы $CM(f)$.

Доказательство в обратную сторону: Пусть существует размещение $Arr(H)$ в пространстве R^d, удовлетворяющее коммуникационной матрице $CM(f)$. Каждый столбец матрицы $CM(f)$ задаёт гиперплоскость. Каждая строка матрицы $CM(f)$ задаёт область (или точку внутри области) в размещении $Arr(H)$.

Будем строить протокол Ф, вычисляющий f с неограниченной ошибкой, имеющий длину $d+2$.

Для размещения $Arr(H)$ пусть $p = (p_1, \ldots, p_d)$ — точка внутри области, $q = (q_1, \ldots, q_d)$ - гиперплоскость.

Преобразуем вектор p следующим образом.

1. Построим по вектору p вектор p':

$$p' = (p_1, \ldots, p_d, 0, -\sum_{i=1}^{d} p_i).$$

   Сумма компонент вектора p' равна нулю. Компоненты вектора p' могут быть как положительные так и отрицательные.

2. Умножим каждый элемент вектора p' на положительную константу c такую, чтобы каждый элемент вектора p' стал по модулю меньше $\frac{1}{d+2}$.

3. Построим вектор $a = cp' + (\frac{1}{d+2}, \ldots, \frac{1}{d+2})$. Очевидно, что вектор a — стохастический.

Преобразуем вектор q следующим образом

1. Построим по вектору q вектор q':

$$q' = (q_1, \ldots, q_d, -\sum_{i=1}^{d} q_i, 0)$$

2. Умножим каждый элемент вектора q' на положительную константу c' такую, чтобы каждый элемент вектора q' стал по модулю меньше $\frac{1}{2}$.

3. Построим вектор $b = cq' + (\frac{1}{2}, \ldots, \frac{1}{2})$. Для каждого элемента вектора выполняется $0 \leq b_i \leq 1$.

Построим протокол Ф следующим образом. На входе (σ, γ) вероятностный вектор $\mu(\sigma)$, соответствующий набору σ — это вектор $a(\sigma)$, вектор, соответствующий набору γ — это вектор $b(\gamma)$.

Тогда

$$a(\sigma)b(\gamma) = \big(c(p_1, \ldots, p_d, 0, -\sum_{i=1}^{d} p_i) + (\frac{1}{d+2}, \ldots, \frac{1}{d+2})\big) \times$$
$$c\big(q_1, \ldots, q_d, -\sum_{i=1}^{d} q_i, 0\big) + (\frac{1}{2}, \ldots, \frac{1}{2}) = \qquad .$$
$$cc'(p_1, \ldots, p_d)(q_1, \ldots, q_d) + 0 + 0 + \frac{d+2}{(d+2)2}$$

Нетрудно убедиться, что это значение $> 1/2$, если $f(\sigma, \gamma) = 1$ и $< 1/2$, если $f(\sigma, \gamma) = 0$. □

3.4.2 Применение метода гиперплоскостей

Пример 1. Функция «Равенство». Рассмотрим функцию EQ:

$$EQ(\sigma, \gamma) = \begin{cases} 1, & \text{если } \sigma = \gamma; \\ 0, & \text{если } \sigma \neq \gamma. \end{cases}$$

Теорема 3.7

$$PC(EQ) = 1$$

Доказательство: Согласно теореме 3.6, протокол для вычисления функции EQ с неограниченной ошибкой может быть построен, если возможно построить размещение 2^n гиперплоскостей с 2^n областями, удовлетворяющее коммуникационной матрице $CM(EQ)$ в пространстве R^d для некоторого d. Так как $CM(EQ)$ содержит единицы на главной диагонали и нули во всех остальных позициях, любое множество гиперплоскостей $H = \{h_1, \ldots, h_{2^n}\}$ и любое множество точек $P = \{p_1, \ldots, p_{2^n}\}$, удовлетворяющее условию, что любая точка $p \in P$ отделяется от всех остальных точек из множества

P какой то гиперплоскостью $h \in H$ будет являться размещением, удовлетворяющим коммуникационной матрице $CM(EQ)$. Покажем, что такое размещение может быть достигнуто в пространстве размерности 2. Согласно теореме 3.6 это будет означать что существует коммуникационный протокол сложности 1, вычисляющий функцию EQ с неограниченной ошибкой.

Расположим точки $p \in P$ равномерно в первой четверти R^2 на окружности радиуса 1 с центром в начале координат. Чтобы отделить каждую точку от остальных, возьмём в качестве гиперплоскости касательную к этой точке, слегка смещённую к центру окружности. Формально:

- $\beta = \pi/2^n + 1$, $\delta = \beta/2$,
- $P = \{(\cos i\beta, \sin i\beta) \mid 0 \le i \le 2^n - 1\}$,
- $H = \{(\cos i\beta, \sin i\beta, \cos\delta) \mid 0 \le i \le 2^n - 1\}$. □

Пример 2. Функция INDEX. Рассмотрим функцию $INDEX$:

$$INDEX(\sigma, \gamma) = \begin{cases} \sigma_i, & \text{если в наборе } \gamma \text{ одна единица в позиции } i; \\ 0, & \text{если в наборе } \gamma \text{ число единиц не равно 1.} \end{cases}$$

Теорема 3.8

$$\lceil \log n \rceil \le PC(INDEX) \le \lceil \log n \rceil + 1.$$

Доказательство: Рассмотрим коммуникационную матрицу функции $INDEX$. На пересечении строки $\sigma = \sigma_1 \dots \sigma_n$ и столбца $\gamma = \gamma_1 \dots \gamma_n$ стоит значение σ_i, если в наборе γ ровно одна единица в позиции i, и стоит значение 0, если в наборе γ нет единиц или более одной единицы. Переупорядочим столбцы в коммуникационной матрице $CM(INDEX)$ таким образом, чтобы сначала шли столбцы, пронумерованные γ с одной единицей, потом — все остальные. После переупорядочивания вторая половина матрицы состоит только из нулей, а на пересечении строк и первых n столбцов находятся всевозможные двоичные последовательности длины n. Таким образом, в размещении, удовлетворяющем коммуникационной матрице

функции $INDEX$, должно быть n гиперплоскостей (соответствующим столбцам), и 2^n областей.

Известно, что число областей, образованных n гиперплоскостями в пространстве R^d не превышает

$$\sum_{i=0}^{d} \binom{n}{i} = \sum_{i=0}^{d} \frac{n!}{i!(n-i)!} = 2^d.$$

Следовательно, $d \geq n$. Применяя теорему 3.6, завершаем доказательство теоремы. □

Пример 3. Функция ISA. Функция $ISA(x, y)$ (Indirect Storage Access) определяется следующим образом:

$$ISA(\sigma, \gamma) = \sigma_{bin(\gamma) \mod n}$$

Теорема 3.9

$$\log n \leq PC(ISA) \leq \log n + 1.$$

Доказательство: Нижняя оценка. Известно (может быть показано), что число различных областей в любом размещении n гиперплоскостей в d-мерном пространстве R^d ограничено сверху величиной $\sum_{i=0}^{d} \binom{n}{i}$ что равно 2^n, так как

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k} = 2^n \text{ при } x = y = 1.$$

Следовательно, d не может быть меньше n ($d \leq n$), что даёт нижнюю оценку.

Верхняя оценка. Существует размещение d гиперплоскостей в R^d, следовательно, мы может достигнуть верхнюю оценку. □

Теорему можно легко расширить, чтобы получить иерархию сложности для $0 \leq C \leq [\log n]$.

Глава 4

Приложения коммуникационных вычислений

В данной главе мы покажем, каким образом могут применяться результаты и техники теории коммуникационных вычислений для доказательства нижних оценок для различных вычислительных моделей: машин Тьюринга, схем из функциональных элементов и конечных автоматов.

4.1 Машины Тьюринга

Машина Тьюринга — является наиболее универсальной вычислительной моделью с потенциально бесконечной памятью в виде ленты, неограниченной в обе стороны. Машина работает в соответствии со своей программой, выполняя на каждом шаге простейшие действия, заключающиеся в смене текущего состояния, записи символа в текущую ячейку на ленте (лентах) и сдвиге на ленте (лентах) на одну ячейку влево/вправо. Модели машины Тьюринга различаются числом используемых лент. Известно, что использование большего количества лент не даёт существенного преимущества. Мы будем рассматривать модель с двумя лентами: входной лентой, предназначенной только для чтения, где располагаются входные данные, и рабочей лентой, предназначенной для чтения и записи, представляющей из себя рабочую память. Машины Тьюринга также различаются тем, решают ли они задачу распознавания (desicion problem) или задачу преобразования входа в выход. Машина Тьюринга-преобразователь преобразует входное слово в

выходное, являющееся результатом её работы. Машина Тьюринга-распознаватель выдает в качестве ответа «Да» или «Нет», в зависимости от того, удовлетворяет ли вход заданным свойствам (принадлежит ли он заданному языку). Мы будем рассматривать машины-распознаватели. Мерами сложности модели машины Тьюринга являются время — число тактов работы машины, и память — число использованных ячеек рабочей ленты. В данном разделе мы покажем, как с помощью теории коммуникационных вычислений доказывается нижняя оценка для этих сложностных мер.

4.1.1 Определение машины Тьюринга

Определение 4.1 *Детерминированная машина Тьюринга (МТ) — это вычислительная модель, которую можно представить как*

$$M = \langle \Sigma, Q, \delta, q_0, q_{acc}, q_{rej} \rangle,$$

где Σ — конечный входной алфавит, Q — конечное множество состояний устройства управления, q_0 — начальное состояние, q_{acc} — финальное принимающее состояние, q_{rej} — финальное отвергающее состояние, $\delta : Q \times \Gamma \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$ — функция перехода, где $\Gamma = \Sigma \cup \Lambda$, Λ — пустой символ, который содержат все пустые ячейки ленты.

Машина M состоит из входной ленты, предназначенной только для считывания, рабочей ленты, предназначенной для считывания и записи, потенциально бесконечных в обе стороны и разбитых на ячейки и устройства управления, которое в каждый момент времени может находиться в одном из состояний множества Q. Машина работает в дискретные моменты времени $t = 0, 1, 2, \ldots$. В начальный момент времени $t = 0$ на входной ленте записано входное слово w, каждая ячейка содержит одну букву входного слова, остальные ячейки пусты, читающая головка обозревает первую букву входного слова. Рабочая лента пуста. Машина начинает работу в начальном состоянии q_0. На каждом шаге работы в соответствии с функцией перехода δ машина меняет свое состояние, меняет символ в текущей ячейке на рабочей ленте и сдвигается на входной и

рабочей ленте на одну ячейку влево, вправо или остается на месте:

$$\delta : Q \times \Sigma \times \Gamma \to Q \times \Gamma \times \{L, R, S\} \times \{L, R, S\}.$$

Когда машина переходит в состояние основа, работа машины прекращается и машина соответственно принимает или отвергает слово w в зависимости от того, является ли конечное состояние состоянием q_{acc} или q_{rej}.

Число шагов, которая делает машина до момента останова, определяет время обработки слова w. Количество ячеек рабочей ленты, которое использует машина в процессе обработки, определяет память, используемую при обработке слова w.

Определение 4.2 *Будем говорить, что машина Тьюринга M принимает язык L, если машина принимает все слова из L, а на словах не из L либо не останавливается, либо не принимает их.*

Определение 4.3 *Будем говорить, что машина Тьюринга M распознает язык L, если машина принимает все слова из L и отвергает все слова не из L.*

Мы рассматриваем машины Тьюринга «распознаватели» (для каждого входного слова машина останавливается через конечное число шагов).

Определение 4.4 *Если язык L распознается некоторой машиной Тьюринга M, то L называется рекурсивным языком.*

4.1.2 Меры сложности

Для заданной машины Тьюринга M определим функции $time_M(w)$ и $space_M(w)$, где $time_M(w) : \Sigma^* \to \mathbb{N}$ равна числу шагов машины M на входе w и $space_M(w) : \Sigma^* \to \mathbb{N}$ равна числу ячеек рабочей ленты, которые использовались машиной M на входе w.

Нас интересует, как ведет себя сложность по времени и сложность по памяти с ростом длины входа. Определим функции, зависящие от параметра $n \in \mathbb{N}$, являющимся длиной входа ($n = |w|$). Обозначим их так же: $time$ и $space$. При этом нас будет интересовать асимптотическое поведение этих функций.

Определение 4.5 *Пусть M — машина Тьюринга. Временна́я сложность определяется как $time_M(n) : \mathbb{N} \to \mathbb{N}$, где*

$$time_M(n) = \max_{\substack{w \in \Sigma^* \\ |w|=n}} time_M(w).$$

Определение 4.6 *Пусть M — машина Тьюринга. Пространственная сложность определяется как $space_M(n) : \mathbb{N} \to \mathbb{N}$, где*

$$space_M(n) = \max_{\substack{w \in \Sigma^* \\ |w|=n}} space_M(w).$$

Как правило, временная и пространственная сложность решения одной и той же задачи взаимосвязаны между собой. Часто при существенном ограничении одного из ресурсов второй начинает расти: пытаясь уменьшить время работы нам придётся увеличить память (произвести предварительные расчеты, сохранить их и впоследствии использовать), пытаясь сократить память, придётся заплатить за это увеличением времени работы (вместо хранения промежуточных результатов придётся их перевычислять). Ниже мы продемонстрируем данный эффект на примере распознавания языка Палиндром.

4.1.3 Язык Палиндром

Язык Палиндром определяется следующим образом:

$$PALINDROM = \{ww^R : w \in \{0,1\}^*\},$$

где w^R представляет из себя строку w, записанную в обратном порядке: $w^R = w_n w_{n-1} \dots w_1$ для $w = w_1 w_2 \dots w_n$.

Мы можем построить разные алгоритмы распознавания данного языка для модели машины Тьюринга, которые различаются сложностными характеристиками: необходимым временем работы и требуемой памятью.

Алгоритм 1.

1. Перепишем входное слово w на рабочую ленту.

2. Установим читающую головку на входной ленте на первый символ входного слова w (слева), читающую/пишущую головку на рабочей ленте — на последний символ слова w (справа).

3. Будем сравнивать текущий символ на входной ленте и текущий символ на рабочей ленте. Если символы совпадают, передвинем головки на входной ленте на одну ячейку вправо, на рабочей ленте на одну ячейку влево.

4. Если на очередном шаге работы текущая пара символов не совпадает, машина переходит в состояние q_{rej} и останавливает работу. Если головки доходят до конца слова — машина переходит в состояние q_{acc} и также останавливает работу.

5. Если слово w является палиндромом, все сравниваемые символы будут равны и машина примет такое слово. Если слово w — не палиндром, обязательно найдутся два символа w_i, w_{n-i+1} такие, что $w_i \neq w_{n-i+1}$, и машина отвергнет такое слово.

Оценим сложность машины M_1, работающей по Алгоритму 1:

$$time_{M_1}(n) = O(n),$$

$$space_{M_1}(n) = O(n).$$

Алгоритм 2.

1. Пусть $n = |w|$. На каждом этапе i работы алгоритма ($i = 1, \ldots, n$) будем сравнивать i-ый символ слова w с соответствующим ему символом w_{n-i+1}.

2. Для этого будем хранить на рабочей ленте номер текущего символа i, увеличивая его на 1 после сравнения очередной пары.

3. При выполнении очередного сравнения машина устанавливается на левую границу слова w, сдвигается на i шагов, считая количество сделанных сдвигов, уменьшая счетчик на рабочей ленте, запоминает i-ый символ входного слова при помощи состояния, и сдвигается на правую границу слова. Затем проделывает аналогичные действия, сдвигаясь от правой границы

слова влево на i шагов, считая количество сделанных сдвигов, уменьшая счетчик на рабочей ленте, и затем сравнивает символ w_{n+1-i} с символом w_i.

4. Если на очередном шаге работы текущая пара символов не совпадает, машина переходит в состояние q_{rej} и останавливает работу. Если на очередном этапе сдвигов головки доходят до конца слова — машина переходит в состояние q_{acc} и также останавливает работу.

5. Если слово w является палиндромом, все сравниваемые символы будут равны и машина примет такое слово. Если слово w — не палиндром, обязательно найдутся два символа w_i, w_{n-i+1} такие, что $w_i \neq w_{n-i+1}$, и машина отвергнет такое слово.

Оценим сложность машины M_2, работающей по Алгоритму 2:

$$time_{M_2}(n) = O(n^2),$$

$$space_{M_2}(n) = O(\log n).$$

Как мы видим, Алгоритм 1 лучше по времени, а Алгоритм 2 лучше по памяти. Встает вопрос: можем ли мы построить алгоритм, эффективный и по времени и по памяти, т.е. который имел бы временную сложность $O(n)$ и пространственную сложность $O(\log n)$?

Теорема 4.1 *Пусть $f : \{0,1\}^n \to \{0,1\}$ — произвольная функция и пусть M — машина Тьюринга, которая принимает слова из множества*

$$\{x0^n y \,:\, |x| = |y| = n, f(x,y) = 1\}$$

и отвергает слова из множества

$$\{x0^n y \,:\, |x| = |y| = n, f(x,y) = 0\}.$$

Тогда выполняется

$$time_M(n) \cdot space_M(n) = \Omega(n \cdot C(f)).$$

Доказательство: По машине M построим коммуникационный протокол Ф для вычисления функции f следующим образом.

Алиса и Боб получают на вход x и y, соответственно. Алиса выполняет вычисления согласно программе машины M до тех пор, пока читающая готовка на входной ленте находится в зоне $x0^n$. Как только головка переходит в зону y Алиса передаёт управление Бобу. Соответственно, Боб выполняет вычисления согласно программе машины M до тех пор, пока читающая готовка на входной ленте находится в зоне $0^n y$. Как только головка переходит в зону x Боб передаёт управление Алисе. Когда машина M останавливается, то участники выдают ответ 1, если финальное состояние q_{acc}, и выдают ответ 0, если финальное состояние q_{rej}.

Оценим коммуникационную сложность полученного протокола Ф. При передаче управления от Алисы к Бобу и наоборот, участник передаёт другому описание текущей конфигурации машины, чтобы тот смог продолжить вычисление. Описание текущей конфигурации включает: текущее состояние управляющего устройства, полное описание рабочей ленты, положение головки на рабочей ленте, что составляет не более $c \cdot space_M(n)$ бит, где c — некоторая константа. Для того чтобы один участник передал управление другому участнику, он должен преодолеть зону между x и y, состоящую из n нулей. Следовательно, такая передача управления может произойти не менее чем через n шагов после того, как участник начал процесс вычисления. Поэтому количество передач управления от одного участника к другому составит не более $time_M(n)/n$ раз. Сложность протокола оценивается как количество переданных бит на раунде умноженное на количество раундов. Получаем, что

$$C(\Phi) = c \cdot space_M(n) \cdot time_M(n)/n.$$

Отсюда получаем

$$c \cdot space_M(n) \cdot time_M(n)/n \geq C(f).$$

И следовательно

$$space_M(n) \cdot time_M(n) = \Omega(n \cdot C(f)).$$

□

Следствие 1 *Для любой машины Тьюринга M, распознающей язык $PALINDROM$ выполняется*

$$space_M(n) \cdot time_M(n) = \Omega(n^2).$$

Доказательство: Пусть M — машина Тьюринга, распознающая язык $PALINDROM$. Рассмотрим множество входов $S \subset \{0,1\}^*$ такое, что

$$S = \{w = xy^R : |x| = |y|, \text{ и } 1/3 \text{ последних бит в } x \text{ и } y \text{ равна } 0\}.$$

Каждая строка из множества S представляется в виде $x0^{n'}y^R$, где $n' = 2n/3$ и $x, y \in \{0,1\}^{n'}, n \in \mathbb{N}$.

Проверка $x0^{n'}y^R \in PALINDROM$ эквивалентна проверке, верно ли, что $EQ_{2n/3}(x,y) = 1$. Так как $C(EQ_{2n/3}) = \Omega(n)$, то по Теореме 4.1 выполняется $space_M(n) \cdot time_M(n) = \Omega(n^2)$. □

4.2 Схемы из функциональных элементов

Схемы из функциональных элементов (circuits в анлг. литературе) — это неоднородная вычислительная модель без памяти, которая предназначена для вычисления булевых функций или булевых операторов.

4.2.1 Определение схемы из функциональных элементов

Определение 4.7 *Схема из функциональных элементов (далее — СФЭ) — это ациклический ориентированный граф, вершины которого бывают трёх видов:*

- *вершины без входящих ребер (входные вершины);*
- *вершины без исходящих ребер (выходные вершины);*
- *внутренние вершины с входящими и выходящими рёбрами.*

Каждая входная вершина помечена переменной из множества $X = \{x_1, \ldots, x_n\}$. Каждая внутренняя вершина (называется функциональным элементом) помечена булевой функцией g из используемого базиса и имеет входную степень, соответствующее арности

соответствующей функции g. Для определённости, далее будем рассматривать схемы в базисе $\{\wedge, \vee, \neg\}$.

Схема C с n входными вершинами представляет вычисление на входной строке длины n. При подаче на вход схемы значений 0 или 1, на выходе схемы генерируется выходное значение. Будем говорить, что схема C вычисляет булеву функцию $f : \{0,1\}^n \to \{0,1\}$ (булев оператор $F : \{0,1\}^n \to \{0,1\}^m$), если для любого $\sigma \in \{0,1\}^n$ результат, генерируемый на выходе (выходах) схемы, совпадает со значением $f(\sigma)$ $(F(\sigma))$.

4.2.2 Меры сложности

Определение 4.8 *Глубиной $Depth(C)$ схемы C будем называть длину самого длинного пути от входной вершины до выходной.*

Определение 4.9 *Сложностью $Size(C)$ схемы C будем называть количество функциональных элементов (внутренних вершин) схемы C.*

Определим схемную сложность функции f:

$$Depth(f) = \min_{\text{схема } C \text{ вычисляет функцию } f} Depth(C)$$

$$Size(f) = \min_{\text{схема } C \text{ вычисляет функцию } f} Size(C)$$

4.2.3 Определение формулы

Формулы — это частный вид схем. Будем рассматривать формулы, составленные из пропозициональных переменных с помощью связок $\wedge, \vee, \neg$ (конъюнкция, дизъюнкция и отрицание). Каждую формулу можно описать в виде схемы определённого вида, где каждая внутренняя вершина имеет выходную степень равную 1, операция отрицания может применяться только к входным переменным.

Определение 4.10 *Формулой над множеством переменных $\{x_1, \ldots, x_n\}$ называется схема следующего вида:*

- *Граф, представляющий схему, является деревом.*

- *Внутренние вершины помечены* $\wedge$ *или* $\vee$.
- *Входные вершины помечены* x_i *или* $\neg x_i$.

Минимальную глубину и сложность формулы, реализующей функцию f, будем обозначать также $Depth(f)$ и $Size(f)$, соответственно. Минимальная сложность $Size(f)$ схемы, представляющей формулу, соответствует (с точностью до умножения на некоторую константу) числу символов в обычной записи булевой формулы, $Depth(f)$ соответствует «глубине вложенности» скобок в стандартной линейной записи формулы.

4.2.4 Задача Карчмера-Вигдерсона

Для произвольной функции $f : \{0,1\}^n \to \{0,1\}$ рассмотрим следующую коммуникационную задачу. Пусть Алисе дано слово $x \in \{0,1\}^n$ такое что $f(x) = 1$, Бобу дано слово $y \in \{0,1\}^n$, такое что $f(y) = 0$. Алиса и Боб должны найти номер $i \in \{1,\ldots,n\}$ такой, что $x_i \neq y_i$. Назовём эту задачу KW_f. Поскольку $f(x) \neq f(y)$, то существует по крайней мере один ответ для этой задачи. Отметим, что для пар слов x, y, которые различаются в нескольких позициях, корректными ответами могут быть несколько значений i; каждое из них допускается в виде ответа.

Задача KW_f интересна тем, что детерминированная коммуникационная сложность решения этой задачи в точности равна минимальной глубине схемы, вычисляющей функцию f.

Теорема 4.2 *Для любой функции* $f : \{0,1\}^n \to \{0,1\}$ *выполняется*

$$C(KW_f) = Depth(f).$$

Доказательство: Докажем $C(KW_f) \leq Depth(f)$. Для этого по схеме C, вычисляющая функцию f, построим коммуникационный протокол с коммуникационной сложностью $Depth(C)$.

Предположим, для определённости, что Алисе дан набор битов $x = (x_1,\ldots,x_n)$, для которого $f(x) = 1$, а Бобу — набор битов $y = (y_1,\ldots,y_n)$, для которого $f(y) = 0$. И у Алисы и у Боба есть схема, в соответствии с которой они строят протокол. Коммуникационный протокол будет работать следующим образом. Алиса и

Боб двигаются по дереву формулы, начиная с корня по направлению к листьям вдоль пути, на котором функциональные элементы для входов Алисы и Боба выдают разные значения.

На первом шаге Алиса и Боб находятся в корневой вершине дерева, на вход которой подаются выходы двух поддеревьев. По условию задачи на выходе генерируются разные значения: 1 у Алисы и 0 у Боба. Предположим для определённости, что корневой элемент помечен операцией $\vee$. Тогда для входного набора x хотя бы на один из входов этого элемента должно подаваться значение 1, а на входе y на оба входа корневого элемента подаются значения 0. Входы корневого элемента являются выходами поддеревьев T_1 и T_2. Значит для входа x выход по крайней мере одного из этих поддеревьев должен возвращать значение 1, а на входе y оба поддерева должны возвращать значение 0. Алиса выбирает одно из двух поддеревьев (T_0 или T_1), на выходе которого для входа x генерируется значение 1 и отправляет номер выбранного поддерева (0 или 1) Бобу. Далее игроки рекурсивно применяют данную стратегию для выбранного поддерева, на выходе которого значения для входов Алисы и Боба различаются. Если же корневая вершина помечена операцией $\wedge$, то для входа x оба поддерева T_1 и T_2 должны генерировать значение 1, а для входа y по крайней мере одно из этих двух поддеревьев должно генерировать значение 0. В этом случае Боб выбирает поддерево, на выходе которого генерируется значение 0, и также отправляет Алисе номер выбранного поддерева, после чего игроки продолжают вычисление для данного поддерева. Когда Алиса и Боб попадают в лист дерева, они тем самым узнают литерал, для которого значения наборе x и в наборе y различны, что и требовалось. Ясно, что коммуникационная сложность построенного протокола совпадает с глубиной формулы. Следовательно $C(KW_f) \leq Depth(f)$.

Докажем, что $C(KW_f) \geq Depth(f)$. Пусть Φ — оптимальный протокол, вычисляющий KW_f, то есть протокол, который для любых x, y, являющихся входами Алисы и Боба, соответственно, таких, что $f(x) \neq f(y)$, выдает i такое, что $x_i \neq y_i$. Преобразуем Φ в дерево-формулу. Будем строить дерево, начиная с корня. Каждую вершину, в которой Алиса передает Бобу бит, пометим операцией $\vee$,

а каждую вершину, в которой Боб передает бит Алисе, пометим операцией $\wedge$. Каждый лист дерева пометим индексом $\in \{1, \ldots, n\}$, являющимся результатом протокола. Рассмотрим множество входов $S \times T \subseteq f^{-1}(0) \times f^{-1}(1)$ таких, для которых протокол Φ приводит в одну и ту же листовую вершину дерева, помеченную $i \in \{1, \ldots, n\}$. Тогда выполняется один из следующих пунктов:

1. либо $x_i = 1$ $\forall x \in S$ и $y_i = 0$ $\forall y \in T$,
2. либо $x_i = 0$ $\forall x \in S$ и $y_i = 1$ $\forall y \in T$.

Действительно, предположим, это не так, и существует две пары (x, y), $(x', y') \in S \times T$ такие, что $x_i = 1$, $y_i = 0$ и $x'_i = 0$, $y'_i = 1$. Из этого следует, что (x, y') и (x', y) также принадлежат $S \times T$. Следовательно, протокол на этих входах также выдает ответ i, что противоречит тому, что протокол правильно вычисляет KW_f. Пометим эту листовую вершину литералом z_i, если она соответствует случаю 1, и литералом $\neg z_i$, если она соответствует случаю 2.

Глубина построенной формулы в точности равна $C(KW_f)$. Осталось показать, что построенная формула вычисляет функцию f.

Лемма 4.1 *Для каждой внутренней вершины v построенной формулы, функция f', соответствующая этой вершине, удовлетворяет свойству: $f'(x) = 1$, для всех $x \in A$, и $f'(y) = 0$, для всех $y \in B$, где $A \times B$ — множество входов, на которых достижима вершина v в дереве протокола.*

Доказательство: Проведем доказательство индукцией по глубине d формулы.

База индукции: Формула глубины $d = 0$ состоит только из литерала. Очевидно, что в этом случае условие леммы выполняется.

Индукционный шаг: Пусть вершина v является выходом формулы глубины d, вычисляющей функцию f'. На вход вершины v подаются выходы вершин v_0 и v_1, являющихся выходами формул глубины не более $d - 1$, вычисляющих функции f'_0 и f'_1, соответственно. Пусть $A \times B$ — множество входов, на которых достижима вершина v. Предположим, что в вершине v Алиса передает бит

Бобу, и, следовательно, вершина v помечена операцией $\vee$. В зависимости от значения передаваемого бита множество A разбивается на две части A_0 и A_1 таким образом, что на входах $A_0 \times B$ достижима вершина v_0, а на входах $A_1 \times B$ достижима вершина v_1. По предположению индукции, $\forall y \in B$ выполняется $f'_0(y) = f'_1(y) = 0$, и $\forall x \in A_0$ выполняется $f_0(x) = 0$ и $\forall x \in A_1$ выполняется $f'_1(x) = 1$. Так как $f' = f'_0 \vee f'_1$, то $\forall y \in B$ выполняется $f'(y) = 0$ и $\forall x \in A$ выполняется $f'(x) = 1$. Случай, если в вершине v Боб передает бит Алисе, и, следовательно, вершина v помечена операцией $\wedge$, рассматривается аналогично. $\square$

Из доказанной леммы следует, что на выходе формулы вычисляется функция f, так как на выходе генерируется значение 1 для всех $x \in f^{-1}(1)$ и генерируется значение 0 для всех $y \in f^{-1}(0)$. $\square$

4.3 Конечные автоматы

Конечные автоматы — вычислительная модель с конечной памятью. Автомат-преобразователь преобразует входное слово в выходное, автомат-распознаватель для входного слова выдает ответ «Да» или «Нет» в зависимости от того, удовлетворяет ли вход заданному правилу. Кроме классической модели одностороннего конечного автомата также рассматриваются и другие ее разновидности: двусторонние конечные автоматы, многоголовочные конечные автоматы и т.д. Мы будем рассматривать наиболее простую модель — односторонние конечные автоматы-распознаватели.

Определение 4.11 *Конечный односторонний детерминированный автомат A (автомат без выхода) — это пятерка*

$$A = \langle \Sigma,\, Q,\, \delta,\, q_0,\, Accept \rangle,$$

где Σ — конечный входной алфавит, Q — конечное множество состояний, $\delta : Q \times \Sigma \to Q$ — функция переходов, $q_0 \in Q$ — начальное состояние, $Accept \subseteq Q$ — множество принимающих состояний.

Функция δ естественным образом доопределяется на множестве $Q \times \Sigma^*$. Автомат A распознает язык $L \subseteq \Sigma^*$, если $\delta(q_0, \sigma) \in Accept$ для любого $\sigma \in L$, и $\delta(q_0, \sigma') \notin Accept$ для любого $\sigma' \notin L$.

Сложностью $Size(A)$ конечного автомата A называется число его состояний.

Покажем, как с помощью теории коммуникационных вычислений можно доказывать нижние оценки сложности конечных автоматов, распознающих заданный язык L. Для простоты будем рассматривать двоичный входной алфавит $\Sigma = \{0, 1\}$.

Лемма 4.2 *Пусть $L \subseteq \{0,1\}^*$ — язык. Пусть $f : \{0,1\}^{2n} \to \{0,1\}$ такая, что $f(x,y) = 1 \Leftrightarrow xy \in L$. Тогда для любого автомата A, распознающего язык L выполняется*

$$\log Size(A) \geq C(f).$$

Доказательство: Пусть A — автомат с наименьшим числом состояний, распознающий L. По автомату A построим коммуникационный протокол Ф. На входе xy, где $x, y \in \{0,1\}^n$ Алиса моделирует работу автомата на части x входа, Боб — на части y. При переходе читающей головки с зоны x на зону y Алиса передает Бобу текущее состояние q автомата, чтобы он мог продолжить обработку. Совместно, Алиса и Боб, моделируют работу автомата на входе xy. По построению, если $xy \in L$, то $f(x,y) = 1$, если $xy \notin L$, то $f(x,y) = 0$. Сложность построенного протокола $C(\Phi) = \log |Q|$. Следовательно, $C(f) \leq \log Size(A)$. $\square$

Литература

[1] *Аблаев Ф.М.* Возможности вероятностных машин по представлению языков в реальное время / Ф.М. Аблаев // Известия ВУЗов, Математика, 1985. – Вып. 7. – С. 32–40.

[2] *Аблаев Ф.М.* Влияние степени изолированности точки сечения на число состояний вероятностного автомата / Ф.М. Аблаев // Математические заметки, 1988. – Т. 44. – Вып. 3. – С. 289–297.

[3] *Аблаев Ф.М.* Сравнительная сложность представления языков в вероятностных автоматах / Ф.М. Аблаев // Кибернетика, 1989. – С. 21–26.

[4] *Аблаев Ф.М.* О сравнительной сложности вероятностных и детерминированных автоматов / Ф. М. Аблаев // Дискретная математика, 1991. – Т. 3. – Вып. 2. – С. 114–120.

[5] *Аблаев Ф.М.* Нижние оценки для вероятностной односторонней коммуникационной сложности булевых функцийв / Ф.М. Аблаев // Методы и системы технической диагностики. Тезисы докладов X-ой международной конференции по проблемам теоретической кибернетики, 1993. – Вып. 18. – С. 3.

[6] *Агафонов В.Н.* Сложность алгоритмов и вычислений / В.Н. Агафонов. – Новосибирск, изд. Новосибирского университета, 1975, – 145 с.

[7] *Александров П.С.* Введение в теорию множеств и общую топологию / П.С. Александров. – М.: Наука, 1977. – 368 с.

[8] *Асарин Е.А.* О сложности равномерных приближений непрерывных функций / Е.А. Асарин. Успехи математических наук, 1984. – Т. 39. – Вып. 3. – С. 157–170.

[9] Проблемы Гильберта / Сб. под общей редакцией П.С. Александрова. – М.: Наука, 1969. – 240 с.

[10] *Арнольд В.И.* О функциях трех переменных / В.И. Арнольд // Доклады Академии наук, 1957. – Т. 114. – Вып. 4. – С. 679–681.

[11] *Ахо А.* Сложность вычислений в СБИС / А. Ахо, Д. Ульман – М: Мир, 1991.

[12] *Ахо А.* Построение и анализ вычислительных алгоритмов / А. Ахо, Д. Хопкрофт, Д. Ульман. – М: Мир, 1979. – 537 с.

[13] *Барздинь Я.М.* Сложность распознавании симметрии на машинах Тьюринга / Я.М. Барздинь // Проблемы кибернетики, 1965. – Вып. 15. – С. 245–248.

[14] *Барздинь Я.М.* Конечные автоматы. Поведение и синтез / Я.М. Барздинь, Б.А. Трахтенброт . – М.: Наука, 1970. – 399 с.

[15] *Бухараев Р.Г.* Управляемые генераторы случайных кодов / Р.Г. Бухараев, В.М. Захаров. – Казань, изд. Казанского университета, 1978. – 160 с.

[16] *Верещагин Н.* Коммуникационная сложность / Н. Верещагин – URL: https://www.lektorium.tv/course/22755 (дата обращения 14.12.2024).

[17] *Витушкин А.Г.* К тринадцатой проблеме Гильберта / А.Г. Витушкин // Доклады Академии наук, 1954. – Т. 95. – Вып. 4. – С. 243–250.

[18] *Витушкин А.Г.* Оценка сложности задачи табулирования / А.Г. Витушкин. – М.: Наука, 1959. – 228 с.

[19] *Габбасов Н.З.* Замечание об одной оценке, относящейся к теореме редукции Рабина / Н.З. Габбасов. – Рукопись деп. в ВИНИТИ 25 февраля 1988, № 1532–В88.

[20] *Галлагер Р.* Теория информации и надежная связь / Р. Галлагер. – М.: Мир, 1972. – 738 с.

[21] *Гашков С.Б.* Сложность приближенного вычисления действительных чисел, непрерывных функций и линейных функционалов: автореферат на соискание ученой степени доктора физико-математических наук / С.Б. Гашков. – МГУ, Москва, 1992. – 38 с.

[22] *Кузьмин В.А.* Реализация функций алгебры логики автоматами, нормальными алгоритмами и машинами Тьюринга / В.А. Кузьмин // Проблемы кибернетики, 1979. – Вып. 36. – С. 181–194.

[23] *Колмогоров А.Н.* О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного и сложения / А.Н. Колмогоров // Доклады Академии наук, 1957, – Т. 114. – Вып. 5. – С. 953–956.

[24] *Колмогоров А.Н.* Оценки минимального числа элементов ε-сетей в различных функциональных классах и их применение к вопросу о представимости функций нескольких переменных суперпозициями функций меньшего числа переменных / А.Н. Колмогоров // Доклады Академии наук, 1955. – Т. 101. – Вып. 2. – С. 192–194.

[25] *Колмогоров А.Н.* Различные подходы к оценке трудности приближенного задания и вычисления функций / А.Н. Колмогоров // В сб.: Proceedings of the International Congress of Mathrmaticians 1962, Stockholm, 1963. – P. 352–356.

[26] *Колмогоров А.Н.* ε-энтропия и ε-емкость множеств и в функциональных пространствах / А.Н. Колмогоров, В.М. Тихомиров // Успехи математических наук, 1959. – Т. 14. – № 2. – С. 3–86.

[27] *Кочкарев Б.С.* Об устойчивости вероятностных автоматов / Б.С. Кочкарев // Кибернетика, 1968. – Вып. 2. – С. 95–111.

[28] *Лупанов О.Б.* О возможности синтеза схем из произвольных элементов / О.Б. Лупанов // Труды Математического Института имени В.А.Стеклова АН СССР, 1958. – Т. 51. – С. 158–173.

[29] *Лупанов О.Б.* Асимптотические оценки сложности управляющих систем / О.Б. Лупанов – М.: изд. Московского университета, 1984. – 124 с.

[30] *Леу К.* Вычислимость на вероятностных машинах / К. Леу, Э.Ф. Мур, К.Э. Шеннон, Н. Шапиро // Автоматы, 1956. – С. 242–278.

[31] *Марченков С.С.* Об одном методе анализа суперпозиции непрерывных функций / С.С. Марченков // Проблемы кибернетики, 1980. – Вып. 37. – С. 5–17.

[32] *Нигматуллин Р.Г.* Сложность булевых функций / Р.Г. Нигматуллин — М.: Наука, 1991. – 238 с.

[33] *Никольский С.М.* Ряды Фурье функций с данным модулем непрерывности / С.М. Никольский // Доклады Академии наук, 1946. – Т. 52. – С. 191–194.

[34] *Офман Ю.П.* О приближенной реализации непрерывных функций на автоматах / Ю.П. Офман // Доклады Академии наук, 1963. – Т. 152. – Вып. 4. – С. 823–826.

[35] *Покровская И.А.* Некоторые оценки числа состояний вероятностных автоматов, представляющих регулярные языки / И.А. Покровская // Проблемы кибернетики, 1979. – Вып. 36. – С. 181–194.

[36] *Рабин М.* Вероятностные автоматы / М. Рабин // Кибернетический сборник, 1964. – Вып. 9. – С. 123–141.

[37] *Соловьев Н.А.* Тесты (теория, построение, применение) / Н.А. Соловьев. — Новосибирск, Наука, 1978. – 189 с.

[38] *Тиман А.Ф.* Теория приближений функций действительного переменного / А.Ф. Тиман — М.: Наука, 1960. – 624 с.

[39] *Тихомиров В.М.* Некоторые вопросы теории приближений / В.М. Тихомиров – М.: изд. Московского университета, 1976. – 304 с.

[40] *Фрейвалд Р.В.* Ускорение распознавания некоторых множеств применением датчика случайных чисел / Р.В. Фрейвалд // Проблемы кибернетики, 1979. – Вып. 36. – С. 209–224.

[41] *Фрейвалд Р.В.* О некоторых преимуществах вероятностных машин по сравнению с детерминированными / Р.В. Фрейвалд, Э. Икауниекс // Изв. вузов. Математика. 1977. – № 2. – С. 118–123.

[42] *Фрейвалд Р.В.* Об увеличении числа состояний при детерминизации конечных вероятностных автоматов / Р.В. Фрейвалд // Автоматика и вычислительная техника, 1982. – Вып. 3. – С. 39–42.

[43] *Фрейвалд Р.В.* Сложность вычислений на вероятностных и детерминированных машинах односторонних машинах Тьюринга / Р.В. Фрейвалд // Кибернетика и вычислительная техника, 1986. – – Вып. 2. – С. 147–179.

[44] *Храпченко В.М.* Об одном методе получения нижних оценок сложности П-схем / В.М. Храпченко // Математические заметки, 1971. – Т. 10. – Вып. 1. – С. 83–92.

[45] *Чегис И.А.* Логические способы контроля работы электрических схем / И.А. Чегис, С.В. Яблонский // Труды Математического Института имени В.А. Стеклова АН СССР, Москва, 1958. – Т. 51. – С. 270–360.

[46] *Яблонский С.В.* Об алгоритмических трудностях синтеза минимальных контактных схем / С.В. Яблонский // Проблемы кибернетики, 1959. – Вып. 2. – С. 75–121.

[47] *Яблонский С.В.* Введение в дискретную математику / С.В. Яблонский — М: Наука, 1986. – 325 с.

[48] *Ablayev F.* Why sometimes probabilistic algorithms can be more effective / F. Ablayev, R. Freivalds // in Proc. of the MFCS'86, Lecture Notes in Computer Science, 1986. – V. 233. – P. 1–14.

[49] *Ablayev F.* Possibilities of probabilistic one-way counting machines / F. Ablayev // in Proc. of the FCT'87, Lecture Notes in Computer Science, 1987. – V. 278. – 1–4.

[50] *Ablayev F.* The complexity properties of probabilistic automata with isolated cut point / F. Ablayev // Theoretical Computer Science, 1988. – No. 57. – P. 87–95.

[51] *Ablayev F.* On Comparing Probabilistic and Deterministic Automata Complexity of Languages / F. Ablayev // in Proc. of the MFCS'89, Lecture Notes in Computer Science, 1989. – V. 379. – P. 599–605.

[52] *Ablayev F.* Lower bounds for probabilistic space complexity: automata approach / F. Ablayev // University of Rochrster (USA), Technical Report 423, 1992. – P. 1–13.

[53] *Ablayev F.* Lower bounds for one-way probabilistic communication complexity / F. Ablayev // in Proc. of the ICALP'93, Lect. Lecture Notes in Computer Science, 1993. – V. 700. – P. 241–252.

[54] *Aho A.* On notion of information transfer in VLSI circuits / A. Aho, J. Ulman, M. Yanakakis // in Proc. of the 15th Annual ACM Symposium on the Theory of Computing, 1983. – P. 133–139.

[55] *Chor B.* Unibased bits from sources of weak randomness and probabilistic communication complexity / B. Chor, O. Goldreich // Siam J. Comput. 1988. – V. 17. – No. 2. – P. 230–260.

[56] *Gill J.* Computational complexity of probabilistic Turing machines / J. Gill // SIAM J. Comput., 1977. – No. 6. – P. 675–695.

[57] *Phan Dinh Dieu* On a Necessary Condition for Stochastic Languages / Dinh Dieu Phan // Elektronische Informationsverarbeitung und Kybernetik, 1972. – V. 8. – P. 575–588.

[58] *Duris P.* Lower bounds on Communication Complexity / P. Duris, Z. Galil, G. Schnitger // in Proc. of the 16th Annual ACM Symposium on the Theory of Computing, 1984. – P. 81–89.

[59] *Gamal A.* Communication complexity / A. Gamal, A. Orlitsky // in Complexity in Information Theory Editor Yaser S. Abu-Mostafa, Springer-Verlag, 1988. – P. 16–61.

[60] *Freivalds R.* Fast Probabilistic Algorithms / R. Freivalds // in Proc. of the Conference Mathematical Foundation of Computer Science 1979, 1979. – V. 74. – P. 57–69.

[61] *Freivalds R.* Probabilistic two-way machines / R. Freivalds // in Proc. of MFCS'81, Lecture Notes in Computer Science, 1981. – V. 118. – P. 33–45.

[62] *Freivalds R.* Complexity of Probabilistic Versus Deterministic Automata / R. Freivalds // Baltic Computer Science Selected Papers, Lecture Notes in Computer Science, 19911. – V. 502. – P. 565–613.

[63] *Goldman M.* Majority gates vs. general weighted threshold gates / M. Goldman, J. Hastad, A. Razborov. // in Proc. of 7-th Anual conf. Structure in Complexity Theory, 1992. – P. 2–13.

[64] *Halstenberg B.* On Different Modes of Communication / B. Halstenberg, R. Reischuk // in Proc. of the 20th Annual ACM Symposium on the Theory of Computing, 1988. – P. 162–172.

[65] *Hilbert D.* Mathematische Probleme / D. Hilbert // Nachr. Akad. Wyss. Gottingen 19002. – P. 253–297.

[66] *Hromkovich J.* Communication complexity and parallel computing / J. Hromkovich. – Springer-Verlag, Berlin, Heidelberg, New York, 1997. – 336 p.

[67] *Ja'Ja' J.* Information transfer under different sets of protocols / J. Ja'Ja', V. K. Prasana Kumar, J. Simon // SIAM J. Comput. 1984. – V. 13. – P. 840–849.

[68] *Karchmer M.* Monotone circuits for connectivity require superlogarithmic depth / M. Karchmer, A. Wigderson // in Proc. of the 20th ACM STOC'88, 1988. – P. 539–550.

[69] *Karchmer M.* Characterizing non-deterministic circuits size / M. Karchmer, A. Wigderson // in Proc. of the ACM STOC'93, 1993. – P. 532–540.

[70] *Kushilevitz E.* Communication Complexity / E. Kushilevitz, N. Nisan. – Cambridge University Press, 1997. – 189 p.

[71] *Wah Lam T.* Results on Communication Complexity Classes / T. Wah Lam, W. Ruzzo // Journal of Computer and System Sciences, 1992. – V. 44. – P. 324–342.

[72] *Lengauer T.* VLSI Theory / T. Lengauer // in Handbook of Theoretical Computer Science, Edited by J. van Leeuwen, Elsevier Science Publishers B. – V. 1990. – P. 837–868.

[73] *Lorentz G.* Metric Entropy, Widths and Superpositions Functions / G. Lorentz// Amer. Math. Monthly, 1962. – V. 69. – P. 469-485.

[74] *Lovasz L.* Communication Complexity: A Survey / L. Lovasz // Paths, Flows and VLSI Layout, 1990. – P. 235–266.

[75] *Lipton R.* Lower bounds for VLSI / R. Lipton, R. Sedgewick// in Proc. of the 13-th STOC, 1981. – P. 300–307.

[76] *Mehlhorn K.* Las Vegas is better than determinism in VLSI and distributed computing / K. Mehlhorn, E. Schmidt // in Proc. of the 14th ASM STOC, 1982. – P. 330–337.

[77] *van Leeuwen J. (ed.)* Handbook of Theoretical Computer Science / J. van Leeuwen (ed.). – Elsevier Science Publishers B. V. 1990. – 2298 p.

[78] *Papadimitrio C.H.* Communication Complexity / C.H. Papadimitrio, M. Sipser // in Proc. of the 14th ACM STOC 1983. – P. 196–200.

[79] *Paturi H.* Probabilistic Communication Complexity / H. Paturi, J. Simon // Journal of Computer and System Sciences, 1986. – No. 33. – P. 106–123.

[80] *Paz A.* Introduction to probabilistic automata / A. Paz. – N.Y., Wesly, 1971. – 254 p.

[81] *Razborov A.* The gap between the chromatic number of a gap and the rank of its adjacency matrix is superlinear / A. Razborov // Discr. Mathematics, 1992. – No. 108. – P. 393–396.

[82] *Razborov A.* On the distributional complexity of disjointness / A. Razborov // Theor. Comput. Sci., 1992. – No. 106. – P. 385–390.

[83] *Hopcroft J.* Introduction to automata theory, languages, and computation / J. Hopcroft, J. Ulman — N.Y. Wesley, 1979. – 535 p.

[84] *Rabin M.* Probabilistic Algorithms for Testing Primality / M. Rabin // J. Number Theory, 1980. – No. 12. – P. 128–138.

[85] *Stearns R.E.* Hierarchies of memory limited computation / R.E. Stearns, J. Hartmanis, P. M. Lewis // 1965 IEEE Conference Record on Switching Circuit Theory and Logical Design, 19652. – P. 179-190.

[86] *Thompson C.* Area-time complexity for VLSI / C. Thompson // in Proc. of the 11th ACM STOC'79, 1979. – P. 81–88.

[87] *Yao A.* Some Complexity Questions Related to Distributive Computing / A. Yao // in Proc. of the 11th ACM Symposium on the Theory of Computing, 1979. – P. 209–213.

[88] *Yao A.* Lower Bounds by Probabilistic Arguments / A. Yao // in Proc. of the 24th IEEE Symposium on Foundations of Computer Science, 1983. – P. 420–428.