Квантовые алгоритмы: возможности и ограничения

М. Вялый

(Санкт-Петербург, весна 2011)

Оглавление

Лекция 1. Стандартная модель

Квантовые системы устроены так же, как и классические,
но только совсем по-другому

К концу двадцатого века обнаружились неожиданные связи между информатикой и физикой. Оказалось, что эффективность решения многих задач обработки и передачи информации существенно зависит от законов физики.

Два основных вопроса, вытекающих из этих открытий: насколько велики возможности квантовых алгоритмов? возможно ли создание устройств, реализующих эти алгоритмы?

Я буду обсуждать в основном первый вопрос.[1)] Причем естественным кандидатом на сравнение будут вероятностные алгоритмы — классические процедуры, использующие случайность.

При анализе обычных, классических, алгоритмов мы обходимся без физических подробностей, а используем математические модели. Аналогично поступим и в случае квантовых систем: опишем математическую модель, в рамках которой можно строить и анализировать алгоритмы, оперирующие с квантовыми системами.

Практически все задачи информатики можно формулировать в рамках такой общей картины. Есть *носители информации* — некоторые системы, которые могут находиться в одном из нескольких состояний. *Исполнители* (люди или машины) могут пересылать носители информации друг другу, менять их содержимое, объединять в группы или выбрасывать часть информации.

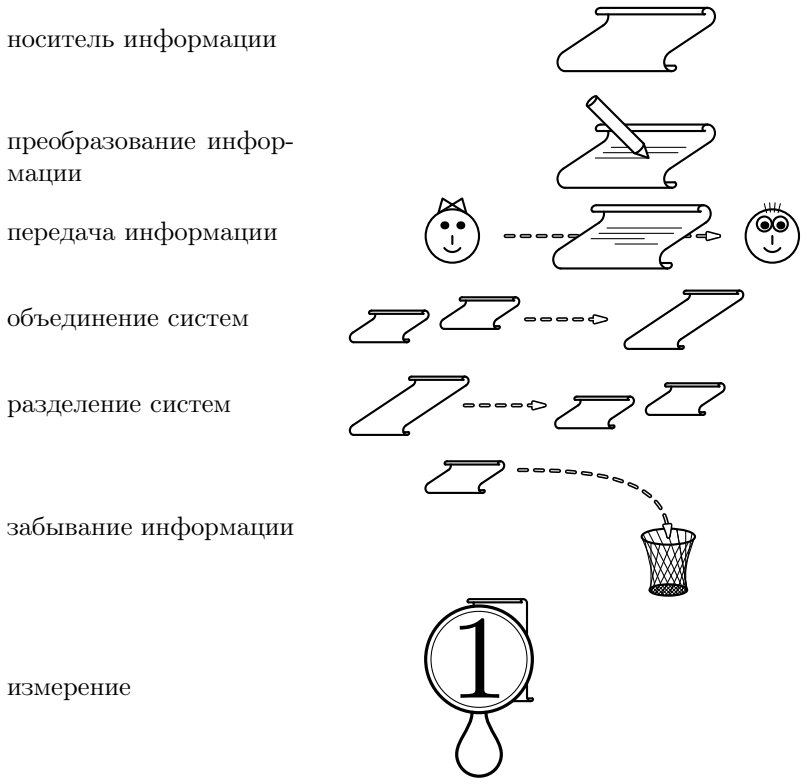


Рис. 1. Носители информации и фундаментальные примитивы действий с ними

[1)] Сразу отметим, что здесь не затрагиваются вопросы квантовой теории информации и квантовой криптографии. Рекомендуем читателю, интересующемуся этими вопросами, книги [11, 17].

Кроме того, в какой-то момент нам захочется узнать результат всех этих действий. Для этого нужно посмотреть на носитель информации и определить, в каком состоянии он находится. Такая *процедура измерения* существенна и в реальных компьютерах: у нас нет возможности посмотреть глазом на жесткий диск и определить, что там написано. Вместо этого мы должны запустить некоторую довольно сложную технику, которая выведет результат вычислений на экран или на печать.

Конечно, в случае обычных компьютеров выделение измерения как фундаментального примитива действий с информацией не слишком оправдано. Вполне возможно вести протокол вычислений, в котором будут записаны все промежуточные данные. В квантовой информатике это уже не так: различие между самой системой и результатами ее наблюдения становится принципиальным.

Далее я приведу *стандартную квантовую модель* носителей информации и действий с ними. Носитель информации я буду называть *квантовой системой*. Попутно будем сравнивать эту модель с классической.

1.1. Состояния классических и квантовых систем

Для простоты мы ограничимся случаем систем с конечным числом состояний.

В классическом случае такая система — это просто конечное множество. Первый нетривиальный случай: система с двумя состояниями (бит), т. е. множество из двух элементов. Эти элементы обычно обозначаются 0 и 1.

Пока мы описали детерминированную систему. Однако и в классическом случае возможно использование случайности. Простейший пример: подбросив «честную» монету, мы получим с равными вероятностями «орла» или «решку». Состояние монеты после подбрасывания естественно описать как линейную комбинацию

$$\frac{1}{2}\text{«орел»} + \frac{1}{2}\text{«решка»}. \tag{1}$$

Вообще говоря, вероятности состояний могут быть любыми неотрицательными числами, в сумме равными 1. Таким образом, пространство состояний случайного бита (два возможных исхода) имеет вид

$$\{(p_0, p_1) : p_0 \geqslant 0,\ p_1 \geqslant 0,\ p_0 + p_1 = 1\}. \tag{2}$$

Вместо множества из двух элементов мы получаем отрезок, а в общем случае — симплекс

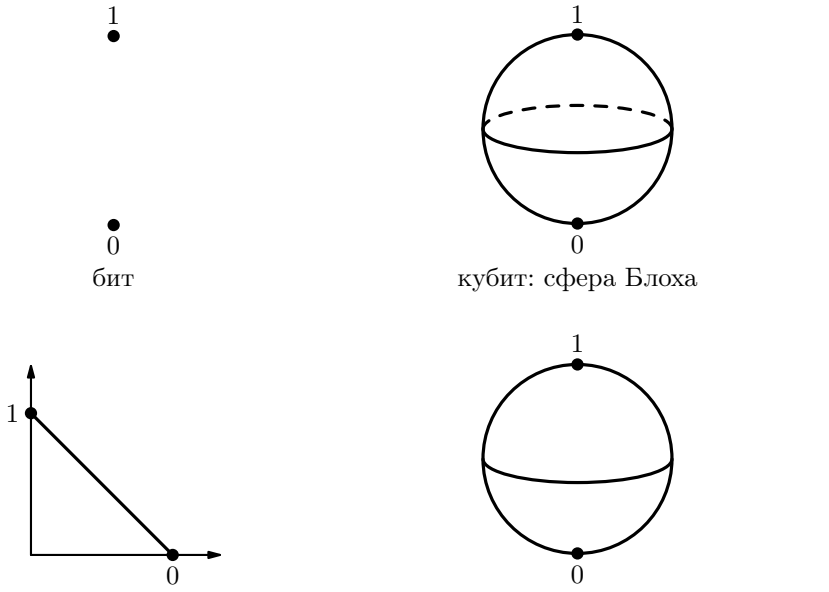
$$\{(p_0, \ldots, p_{n-1}) : p_i \geqslant 0,\ \sum_{i=0}^{n-1} p_i = 1\}. \tag{3}$$

Итак, состояние вероятностной системы — это вектор, размерность которого равна числу возможных исходов.

Случай квантовой системы с двумя состояниями (кубита) аналогичен, но (см. эпиграф) в деталях совершенно непохож на случай вероятностной системы.

Оказывается, что пространство состояний кубита — это 2-мерная сфера. Впрочем, и здесь, как и в случае детерминированной системы, это не вполне точно. Сфера описывает только «чистые» состояния. В самом общем случае, когда возможны рандомизированные смеси «чистых» квантовых состояний, получается шар.

Итак, конечная квантовая система имеет конечное количество исходов n, которые индексируются элементами некоторого конечного множества (в случае кубита это множество $\{0, 1\}$). Исход — это результат наблюдения над системой, что мы подробно обсудим позже.



бит кубит: сфера Блоха

случайный бит смешанное состояние кубита: шар Блоха

Рис. 2. Системы с двумя состояниями

Пространство чистых состояний квантовой системы с n исходами — это $(n-1)$-мерное комплексное проективное пространство. Точки проективного пространства задаются ненулевыми наборами n комплексных чисел, причем два ненулевых набора комплексных чисел задают одно и то же состояние, если они различаются на (комплексный) множитель:

$$(\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) \sim (\beta_0, \beta_1, \ldots, \beta_{n-1}) \quad \Leftrightarrow \quad \alpha_i = \gamma\beta_i.$$

Вопрос. Почему пространство состояний кубита (1-мерное комплексное проективное пространство) — 2-мерная вещественная сфера?

Среди всех наборов комплексных чисел, описывающих состояние квантовой системы, наиболее удобными, как станет ясно дальше, являются нормированные наборы:

$$(\alpha_0, \alpha_1, \ldots, \alpha_{n-1}), \quad \sum_{k=0}^{n-1} |\alpha_k|^2 = 1. \tag{4}$$

Числа из нормированного набора называются *амплитудами*. Нормированные наборы, отличающиеся на общий *фазовый множитель*[2)], равный 1 по модулю, т. е. на число вида[3)] $e^{i\varphi}$, $\varphi \in \mathbb{R}$, задают одно и то же состояние.

Немногие алгоритмы ограничиваются одним битом (или кубитом). Обычно мы вынуждены рассматривать более сложные системы. Обычный способ получить сложную систему: взять много битов.

В классическом случае состояния n битов — это двоичные слова длины n и их 2^n штук.

В вероятностном случае мы получаем «многомерное» распределение

$$(p_{i_0 i_1 \ldots i_{n-1}}), \quad p_{i_0 i_1 \ldots i_{n-1}} \geqslant 0, \quad \sum_{(i_0, i_1, \ldots, i_{n-1}) \in \{0,1\}^n} p_{i_0 i_1 \ldots i_{n-1}} = 1. \tag{5}$$

[2)] Физики называют аргумент комплексного числа фазой.

[3)] Напомним формулу Эйлера $\cos\varphi + i\sin\varphi = e^{i\varphi}$.

В квантовом случае мы получаем вектор в комплексном пространстве:

$$\sum_{(i_0,i_1,\ldots,i_{n-1})\in\{0,1\}^n} \alpha_{i_0 i_1 \ldots i_{n-1}} |i_0, i_1, \ldots, i_{n-1}\rangle, \quad \sum |\alpha_{i_0 i_1 \ldots i_{n-1}}|^2 = 1. \tag{6}$$

Здесь я использовал так называемые обозначения Дирака: $|\psi\rangle$ обозначает вектор из нашего пространства, а если этот вектор принадлежит вычислительному базису, то мы между $|$ и $\rangle$ пишем его индекс. Обозначения Дирака общеприняты в квантовой физике и квантовой информатике. Поэтому к ним лучше привыкнуть, даже если они кажутся вычурными.

Замечание 1. Довольно часто особенную силу квантовых вычислений видят в том, что пространство состояний системы из n кубитов имеет очень большую размерность 2^n. Сравнение формул (5) и (6) показывает неточность такого наблюдения: 300 кубитов описываются таким же количеством амплитуд, что и 300 случайных битов (вещественных параметров в два раза больше, конечно).

В общем случае объединение систем описывается операцией тензорного произведения. Простое определение тензорного произведения двух векторных пространств U и V можно дать, если в этих пространствах выделены базисы $U = (u_1, \ldots, u_n)$; $V = (v_1, \ldots, v_k)$. Тогда в тензорном произведении пространств $U \otimes V$ выделен базис

$$\begin{aligned} u_j \otimes v_\ell, &\quad 1 \leqslant j \leqslant n;\ 1 \leqslant \ell \leqslant k, \\ |j, \ell\rangle &\quad \text{(в обозначениях Дирака).} \end{aligned} \tag{7}$$

Обозначение $\otimes$ используется, как мы видим, и для тензорного произведения векторов. Тензорное произведение билинейно

$$(\lambda u' + \mu u'') \otimes v = \lambda(u' \otimes v) + \mu(u'' \otimes v); \quad u \otimes (\lambda v' + \mu v'') = \lambda(u \otimes v') + \mu(u \otimes v''). \tag{8}$$

Используя билинейность (8), можно выразить тензорное произведение любой пары векторов через базисные векторы (7).

Итак, если имеется система A в состоянии $|\psi\rangle$ и система B в состоянии $|\xi\rangle$, их объединение дает систему AB в состоянии $|\psi\rangle \otimes |\xi\rangle$.

Чему такая операция соответствует в случае вероятностных распределений? Объединение двух систем описывается совместным распределением, причем величины, относящиеся к двум подсистемам независимы.

Однако не все распределения на составных системах обладают таким свойством. Например, совместное распределение двух битов с вероятностями

$$p_{00} = \frac{1}{2}; \quad p_{11} = \frac{1}{2}; \quad p_{01} = p_{10} = 0. \tag{9}$$

Аналогичное состояние двух кубитов

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{10}$$

также не является тензорным произведением состояний отдельных кубитов.

Для неразложимых состояний составных квантовых систем используется специальный термин «сцепленность». Сцепленность играет большую роль в квантовой теории информации. Как, впрочем, и понятие независимости в теории вероятностей.

1.2. Преобразования чистых состояний

Правило из физики: наблюдение квантовой системы. Если посмотреть (сделать измерение) на квантовую систему в состоянии

$$\{(\alpha_0, \alpha_1, \ldots, \alpha_{n-1}) : \quad \alpha_k \in \mathbb{C}, \quad \sum_{k=0}^{n-1} |\alpha_k|^2 = 1\},$$

то исход k наблюдается с вероятностью $|\alpha_k|^2$.

Итак, с точки зрения «наблюдателя» чистое квантовое состояние уже выглядит как вероятностное распределение, и квадраты модулей амплитуд оказываются вероятностями. А какую роль играют фазовые множители?

Вопрос. Состояния, различающиеся на общий фазовый множитель $e^{i\varphi}$, $\varphi \in \mathbb{R}$, физически неразличимы: вероятности наблюдения исходов у них одинаковы.

Однако и умножение амплитуд на **разные** множители, равные по модулю 1, не изменяет вероятности исходов. Почему же мы не считаем все такие состояния одинаковыми? (В таком случае квантовое пространство состояний выродится в вероятностное.)

Общий ответ заключается в том, что состояния разные, если можно опытным путем обнаружить отличия между ними.

Разницу между состояниями, амплитуды которых различаются лишь фазовыми множителями (модуль равен 1), можно обнаружить двумя способами:

– Выполнить некоторое преобразование (одно и то же в обоих случаях) и потом произвести то же самое измерение.

– Выполнить измерение другим прибором.

Дело в том, что «приборы», которыми мы наблюдаем систему, могут быть разными и результат наблюдения (вероятности исходов) зависит от выбора прибора. Оказывается, выбор прибора — это выбор ортонормированного базиса в унитарном пространстве.[4)]

Слово «унитарный» означает, что пространство снабжено *эрмитовым скалярным произведением*[5)]

$$(\alpha_0, \ldots, \alpha_{n-1}) \cdot (\beta_0, \ldots, \beta_{n-1}) = \alpha_0^* \beta_0 + \cdots + \alpha_{n-1}^* \beta_{n-1}.$$

Здесь $z^* = x - iy$ обозначает число, комплексно сопряженное числу $z = x + iy$.

Ортонормированный базис состоит из попарно ортогональных векторов единичной длины:

$$u_k \cdot u_\ell = \delta_{k\ell} = \begin{cases} 1, & \text{если } k = \ell, \\ 0, & \text{иначе.} \end{cases}$$

Амплитуда состояния x относительно k-го вектора u_k в базисе равна скалярному произведению $x \cdot u_k$.

Мы почти всегда будем обсуждать ситуации, в которых «прибор» фиксирован. Поэтому у нас, как и в вероятностном случае, есть выделенный базис, который обычно называют *вычислительным базисом*.

[4)] Подробное изложение результатов измерения разными приборами можно найти в знаменитых «Фейнмановских лекциях по физике» [15, гл. 3–4] (и, конечно, во многих других учебниках).

[5)] По линейной алгебре существует много замечательных учебников. Например, учебник Кострикина и Манина [9] (в котором, в частности, объясняется связь между линейной алгеброй и квантовой механикой). В общем курсе алгебры Винберга [3] линейной алгебре уделено достаточно много внимания.

Один из основных постулатов стандартной модели квантовой механики: преобразования должны быть линейными.

Правило из физики: преобразования чистых состояний. Преобразования чистых состояний описываются произвольными унитарными операторами:

$$\psi \mapsto U\psi, \quad U^\dagger U = I. \tag{11}$$

Унитарный оператор сохраняет длину вектора

$$\langle\psi|U^\dagger U|\psi\rangle = \langle\psi|\psi\rangle$$

и, более общим образом, скалярное произведение между векторами

$$\langle\xi|U^\dagger U|\psi\rangle = \langle\xi|\psi\rangle$$

Тут нужно подробнее объяснить обозначения Дирака.[6)] Векторы мы обозначаем $|\psi\rangle$, это так называемые *кет-векторы*. Обозначение $\langle\psi|$ — *бра-вектор* — относится к линейным функционалам на нашем исходном векторном пространстве. Как известно, из линейной алгебры, наличие скалярного произведения позволяет определить изоморфизм векторного пространства и пространства линейных функционалов на этом пространстве

$$|\psi\rangle \mapsto \langle\psi|,$$

где значение функционала $\langle\psi|$ на векторе $|\xi\rangle$ равно скалярному произведению, которое обозначается $\langle\psi|\xi\rangle$.

Запись $\langle\xi|A|\eta\rangle$ (A — линейный оператор) можно толковать двояко: либо как скалярное произведение вектора $\langle\xi|$ на вектор $A|\eta\rangle$, либо как — $\langle\xi|A$ на $|\eta\rangle$. Вторая интерпретация задает линейный функционал $\langle\psi| = \langle\xi|A$. Соответствующий кет-вектор $|\psi\rangle$ получается из $|\xi\rangle$ применением линейного оператора $A^\dagger$, который называется *эрмитово сопряженным* к A. Из определения сразу следует, что

$$\langle A^\dagger\xi|\eta\rangle = \langle\xi|A|\eta\rangle.$$

Операторы можно задавать матрицами в ортонормированном базисе:

$$A = \sum_{j,k} a_{jk}|j\rangle\langle k|, \quad \text{где } a_{jk} = \langle j|A|k\rangle \text{ — матричный элемент.}$$

Упражнение 1. Проверьте, что матрица оператора $A^\dagger$ получается транспонированием и комплексным сопряжением: $(A^\dagger)_{jk} = (A_{kj})^*$.

Известная теорема линейной алгебры утверждает, что в некотором ортонормированном базисе унитарный оператор диагонален и на диагонали стоят его собственные числа. То есть матрица оператора в этом базисе имеет вид

$$U_{jk} = \lambda_j\delta_{jk}. \tag{12}$$

Из условия унитарности следует, что $\lambda_j^*\lambda_j = 1$, т. е. все собственные числа унитарного оператора равны по модулю 1.

В вычислительном базисе унитарный оператор записывается матрицей, столбцы (и строки) которой образуют ортонормированный базис.

Эрмитовы операторы. По определению оператор A эрмитов, если он совпадает с эрмитово сопряженным: $A^\dagger = A$. Для эрмитовых операторов также

[6)] Обозначения Дирака используются в квантовой физике и в подавляющем большинстве работ по квантовой информатике. Более подробное описание этих обозначений можно найти, например, в [7, 11] (последнюю книгу можно также использовать в качестве совмещенного учебника по линейной алгебре и квантовой механике).

справедлива теорема о существовании ортонормированного базиса из собственных векторов. Поскольку на диагонали матрицы эрмитова оператора должны стоят вещественные числа ($a_{kk} = a_{kk}^*$), то все собственные числа эрмитова оператора вещественны.

Эрмитовы операторы в квантовой механике играют роль скалярных величин в обычной классической физике. Правило здесь такое: если оператор A (наблюдаемая) имеет собственные числа λ_k и собственные векторы $|\psi_k\rangle$, то при измерении состояния

$$|\psi\rangle = \sum_k c_k |\psi_k\rangle$$

наблюдается значение λ_k с вероятностью $|c_k|^2$. Среднее значение наблюдаемой равно

$$E(|\psi\rangle, A) = \sum_k |c_k|^2 \lambda_k = \sum_k \langle \psi_k | c_k^* c_k \lambda_k | \psi_k \rangle = \langle \psi | A | \psi \rangle.$$

Примером наблюдаемой является *событие*. Это аналог вероятностного понятия события. В квантовой механике событие L — подпространство унитарного пространства. С событием L связана наблюдаемая: ортогональный проектор на это подпространство Π_L. Собственные числа проектора равны 1 (событие происходит) и 0 (событие не происходит).

Вероятность события в состоянии $|\psi\rangle$

$$\mathbf{Pr}(|\psi\rangle, L) = \langle \psi | \Pi_L | \psi \rangle = \langle \psi | \Pi_L^\dagger \Pi_L | \psi \rangle,$$

так как $\Pi_L^2 = \Pi_L$.

Таким образом, вероятность события равна квадрату длину проекции вектора состояния на подпространство, отвечающее этому событию.

Преобразования составной системы. По общему правилу они унитарны. Мы часто будем применять преобразование только к части системы, поэтому сразу обсудим, какие операторы соответствуют этому случаю.

Если мы применяем оператор U к первой части (первому регистру) составной системы AB, то на составную систему действует оператор $U \otimes I$.

Как устроено тензорное произведение операторов? Оно на разложимых векторах действует покомпонентно:

$$Z = X \otimes Y \;\Leftrightarrow\; Z(u \otimes v) = (Xu) \otimes (Yv), \tag{13}$$

а на остальные продолжается по линейности.

Обратите внимание на одно тонкое место: почему такое определение корректно? Ведь для одного и того же вектора можно записать разные разложения в сумму тензорных произведений векторов. Корректность вытекает из более общего утверждения, которое полезно почти в любом рассуждении о тензорных произведениях.

Задача 2. Пусть $\alpha\colon U \times V \to W$ — билинейное отображение. Тогда существует единственное линейное отображение $\beta\colon U \otimes V \to W$, для которого равенство $\beta(u \otimes v) = \alpha(u, v)$ выполняется для любых векторов $u \in U$, $v \in V$.

В определении (13) $\alpha(u, v) = (Xu) \otimes (Yv)$ и тогда β является искомым тензорным произведением.

Задача 3. Докажите, что тензорное произведение унитарных операторов унитарно.

В решении этой задачи удобно использовать мультипликативность скалярного произведения в тензорном произведении унитарных пространств.

Упражнение 4. Проверьте мультипликативность скалярного произведения на разложимых векторах в тензорном произведении унитарных пространств

$$\langle \psi', \xi' | \psi'', \xi'' \rangle = \langle \psi' | \psi'' \rangle \cdot \langle \xi' | \xi'' \rangle.$$

1.3. Стандартная идеализация квантового компьютера

При изучении алгоритмов мы почти всё время будем иметь дело со следующей схемой использования квантового ресурса:

- предварительные манипуляции с классическими системами;
- приготовление некоторого чистого состояния (обычно это одно из состояний вычислительного базиса);
- унитарные преобразования;
- измерение в вычислительном базисе;
- обработка результатов измерения классическими средствами.

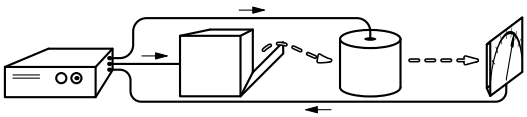


Рис. 3. Принципиальная схема квантового компьютера

Эта последовательность шагов может циклически повторяться. Таким образом, квантовый ресурс порождает вероятностное распределение.

Сила квантовых алгоритмов состоит в том, что в некоторых случаях получение похожего распределения классическими средствами требует большой затраты классического ресурса (по крайней мере, в известных алгоритмах).

Лекция 2. Квантовые запросы к «черному ящику»

Сейчас мы рассмотрим класс алгоритмов, которые я называю «алгоритмами черного ящика». Еще они называются оракульными (неудачное название) или решающими деревьями. Стандартный английский термин — query algorithms.

В общем виде ситуация такова: есть некоторое устройство (черный ящик), способное отвечать на наши вопросы по поводу данных задачи. Алгоритмы решения задачи сравниваются по числу запросов к устройству. Чем меньше запросов, тем лучше алгоритм. На сложность обработки запросов внимания не обращаем.

Рис. 1. Запрос и ответ в модели «черного ящика»

Мы будем рассматривать запросы на вычисление значения булевозначной функции:

$$x \mapsto f(x), \quad x \in X,\ f(x) \in \{0,1\}.$$

2.1. Квантовый запрос

Переведем задачу на квантовый язык. Решающий задачу алгоритм оперирует с какой-то квантовой памятью (квантовой системой). Бит запроса — это часть его памяти. Отдавая этот бит «черному ящику», алгоритм получает в ответ значение функции. Таким образом, «черный ящик» осуществляет некоторое преобразование этого бита.

И тут мы сталкиваемся с трудностью. Как квантовое устройство выдает результат вычисления булевой функции? Некоторые булевы функции необратимы, поэтому нельзя рассчитывать только на применение унитарного оператора (обратимого).

На самом деле, эта проблема возникает и в обычном, классическом случае. Фундаментальные законы классической механики обратимы по времени. Поэтому даже в классическом случае вычисление на микроуровне должно быть обратимым. Например, для вычисления булевозначной функции оно может выглядеть так:

$$\hat{f}\colon (x,y) \mapsto (x, y \oplus f(x)). \tag{1}$$

Заметим, что пару $(x,0)$ такое вычисление переводит в $(x,f(x))$.

Условно механизм вычисления необратимой функции можно описать так: к аргументу присоединяется бит результата (в состоянии 0), значение функции прибавляется по модулю 2 к этому биту, после чего аргумент «выбрасывается», т. е. мы навсегда теряем к нему доступ. См. рис. 2.

Преобразование $\hat{f}$ можно продолжить до унитарного оператора в пространстве $\mathbb{C}^{|X|} \otimes \mathbb{C}^2$. Матрица этого оператора в вычислительном базисе перестановочная.

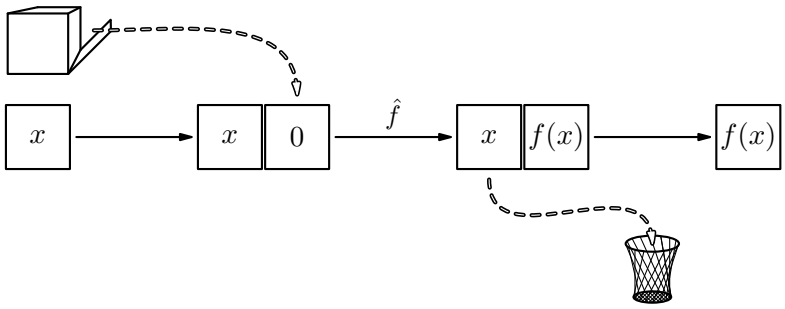


Рис. 2. Необратимое вычисление: взгляд на микроуровне

Например, для функции $f = x$ на одном бите она имеет вид

$$\hat{f} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

в остальных случаях записывается аналогично. При записи матриц в вычислительном базисе мы упорядочиваем битовые строки в лексикографическом порядке.

Будем считать оператор

$$\hat{f}\colon |x, b\rangle \mapsto |x, b \oplus f(x)\rangle \tag{2}$$

квантовым запросом общего вида (см. рис. 3).

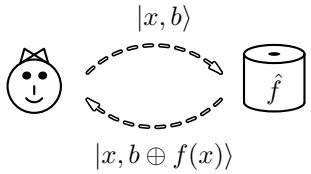


Рис. 3. Квантовый запрос значения функции: передаются и регистр аргумента, и регистр результата

2.2. Моделирование классических действий квантовыми

Наша общая схема использования квантового ресурса включает в себя возможность выполнения классических действий между обращением к квантовому устройству. Это не всегда удобно. Поэтому разберем возможность моделирования классических действий в нашей задаче.

Имитация детерминированных запросов сводится к устранению необратимости. Действительно, квантовый алгоритм может реализовать любую перестановку векторов вычислительного базиса, которой отвечает обратимое классическое преобразование.

Чтобы избавиться от необратимости, нужно обобщить приведенную выше схему. Для отображения $f\colon A \to A$ обратимая имитация состоит в использовании «свежей» копии памяти A в фиксированном состоянии a_0. При этом отображение (1) заменяется на

$$\tilde{f}\colon (a, a_0) \mapsto (a, f(a)). \tag{3}$$

Легко проверить, что частичное отображение (3) продолжается до перестановки на $A \times A$.

Частным случаем является операция «обратимого копирования». В общем случае копирование — необратимое действие. Но существует обратимое преобразование $c\colon A \times A \to A \times A$, для которого

$$c\colon (a, a_0) \mapsto (a, a) \tag{4}$$

при некотором фиксированном $a_0 \in A$. Заметим, что для корректности обратимого копирования необходимо подготовить регистр для новой копии в состоянии a_0. В других состояниях регистра копии копирование не случится.

Теперь рассмотрим случай вероятностных запросов. Запрос по распределению p_k имитируем приготовлением вектора

$$|p\rangle = \sum_k \sqrt{p_k}|k\rangle$$

в регистре аргумента и применением квантового запроса к $|p, 0\rangle$.

Получим в результате состояние

$$\sum_k \sqrt{p_k}|k, f(k)\rangle.$$

Если в этот момент провести *измерение*, получится то же самое вероятностное распределение, что и при вероятностном запросе.

Поскольку данный запрос мог быть не единственным, нужно позаботиться о том, чтобы дальнейшие действия приводили бы к точно тем же результатам, что и в случае вероятностных запросов.

Правило здесь простое: нужно использовать каждый раз свежую копию регистра аргумента, выполняя обратимое копирование (4).

При соблюдении этого правила распределение исходов после финального измерения в точности совпадает с распределением, порождаемым вероятностным алгоритмом.

Продемонстрируем это на примере двух запросов. Первому запросу в вероятностном алгоритме отвечает распределение $p^{(1)}$, а второму — распределения $p^{(2,k)}$ (второе распределение зависит, вообще говоря, от результатов первого).

Действуя по указанному выше правилу, получим квантовым алгоритмом такое состояние перед измерением:

$$\sum_{k_1} \sqrt{p_k^{(1)}}|k_1\rangle \otimes \sum_{k_2} \sqrt{p_{k_2}^{(2,k_1)}}|k_2, F(k_1, k_2)\rangle = \sum_{k_1,k_2} \sqrt{p_k^{(1)}}\sqrt{p_{k_2}^{(2,k_1)}}|k_1, k_2, F(k_1, k_2)\rangle.$$

Вероятность наблюдения исхода $(k_1, k_2, F(k_1, k_2))$ равна

$$p_{k_1}^{(1)} p_{k_2}^{(2,k_1)},$$

как и случае вероятностного алгоритма.

Для нескольких запросов вычисления становятся более громоздкими, но они совершенно аналогичны.

2.3. Фазовый запрос

Есть важный частный случай применения запроса (2), который мы будем далее использовать. А именно, можно приготовить кубит результата в «магическом» состоянии

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

и применить оператор $\hat{f}$.

Оказывается, такая последовательность действий эквивалентна действию унитарного оператора

$$O_f\colon |x\rangle \mapsto (-1)^{f(x)}|x\rangle \tag{5}$$

на регистре аргумента (регистр результата не меняется). Будем называть такой оператор *фазовым запросом*.

Действительно, рассмотрим применение $\hat{f}$ к $|x\rangle \otimes |\psi\rangle$:

$$|x\rangle \otimes \frac{1}{\sqrt{2}}\Big(|0\rangle - |1\rangle\Big) \overset{\hat{f}}{\longmapsto} \frac{1}{\sqrt{2}}\Big(|x\rangle \otimes |f(x)\rangle - |x\rangle \otimes |1 \oplus f(x)\rangle\Big) =$$

$$= \left\{ \begin{array}{ll} |x\rangle \otimes \dfrac{1}{\sqrt{2}}\Big(|0\rangle - |1\rangle\Big), & \text{если } f(x) = 0, \\ -|x\rangle \otimes \dfrac{1}{\sqrt{2}}\Big(|0\rangle - |1\rangle\Big), & \text{если } f(x) = 1, \end{array} \right\} =$$

$$= (-1)^{f(x)}|x\rangle \otimes \frac{1}{\sqrt{2}}\Big(|0\rangle - |1\rangle\Big).$$

Итак, кубит результата вообще не меняется. Поэтому можно рассматривать лишь действие оператора (5) на регистр аргумента и вообще не писать регистр результата в формулах.

Чтобы понять, откуда берется «магическое» состояние, полезно выполнить следующее упражнение.

Упражнение 1. Проверьте, что собственные числа оператора $\hat{f}$ (см. (2)) равны ± 1. Найдите соответствующие им собственные пространства.

2.4. Задача Дойча

Булевых функций от одной переменной ровно 4:

$$0,\ 1,\ x,\ \neg x.$$

Первые две из них — константы, вторые две — нет.

Задача Дойча состоит в том, чтобы определить, является ли функция, вычисляемая «черным ящиком», константой.

Более-менее очевидно, что одного запроса для решения задачи Дойча в классическом случае недостаточно: по значению в одной точке нельзя понять, является ли функция константой.

Сейчас мы увидим, что в квантовом случае достаточно одного фазового запроса вида (5).

Применим к состоянию $|0\rangle$ унитарный оператор HO_fH, где

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{(преобразование Адамара)},$$

и произведем измерение в вычислительном базисе. Ответ 0 будет означать, что функция — константа. Действительно,

$$|0\rangle \overset{H}{\longmapsto} \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \overset{O_f}{\longmapsto} \frac{1}{\sqrt{2}}(-1)^{f(0)}|0\rangle + \frac{1}{\sqrt{2}}(-1)^{f(1)}|1\rangle \overset{H}{\longmapsto}$$

$$\frac{1}{2}(-1)^{f(0)}(|0\rangle + |1\rangle) + \frac{1}{2}(-1)^{f(1)}(|0\rangle - |1\rangle) =$$

$$= \frac{1}{2}\big((-1)^{f(0)} + (-1)^{f(1)}\big)|0\rangle + \frac{1}{2}\big((-1)^{f(0)} - (-1)^{f(1)}\big)|1\rangle =$$

$$= \left\{ \begin{array}{ll} \pm|0\rangle, & \text{если } f \text{ константа}, \\ \pm|1\rangle, & \text{в противном случае}. \end{array} \right. .$$

Задача Дойча — простейший пример дополнительных возможностей квантовых систем в информатике. Преобразование Адамара переводит кубит в состояние

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle.$$

Если мы будем производить измерение в этом состоянии, то получим равномерное распределение. Такое же распределение можно получить подбрасыванием «честной» монеты. Но суть применения преобразования Адамара не в построении какого-то вероятностного распределения, а в «подкрутке» фаз. Амплитуды могут быть отрицательными, в отличие от вероятностей, и поэтому их сумма может оказаться равной 0.

2.5. Задача Дойча – Джоза

Теперь усложним задачу.

Дано: «черный ящик», который вычисляет функцию $f\colon \{0,1\}^n \to \{0,1\}$. Заранее известно, что выполняется одно из двух:

(к) функция f — константа;

(б) функция f сбалансирована, т. е. число нулей и единиц у нее одинаково.

Выяснить: какой из случаев имеет место.

Это пример задачи с априорными ограничениями на входные данные (promise problem).

По-прежнему используем фазовый запрос (5). Только теперь x — строка из нулей и единиц длины n.

Решение задачи Дойча – Джоза почти такое же, как и раньше. Теперь нужно применить оператор Адамара к каждому кубиту, т. е. использовать $H^{\otimes n}$.

Итак, применим оператор $J = H^{\otimes n} O_f H^{\otimes n}$ к исходному состоянию $|0^n\rangle$ и выполняем измерение в классическом базисе. Если результат измерения — $|0^n\rangle$, то f — константа, т. е. имеет место случай (к). В противном случае имеет место случай (б).

Оператор O_f применяется один раз, так что задача Дойча – Джоза решается за один запрос, как и в случае одного бита.

Проверим это утверждение, вычислив вероятность наблюдения исхода 0^n в описанной процедуре.

Прежде всего запишем оператор H в виде, пригодном для суммирования:

$$H|\alpha\rangle = \frac{1}{\sqrt{2}} \sum_{\beta\in\{0,1\}} (-1)^{\alpha\cdot\beta}|\beta\rangle.$$

Здесь в показателе написано булево произведение, которое, впрочем, не отличается от обычного.

Оператор $H^{\otimes n}$ действует на каждом кубите оператором Адамара. Вычислим действие на базисных векторах вычислительного базиса

$$H^{\otimes n}|\alpha\rangle = H^{\otimes n}|\alpha_1,\dots,\alpha_n\rangle = \bigotimes_{k=1}^{n} H|\alpha_k\rangle = \frac{1}{2^{n/2}} \bigotimes_{k=1}^{n} \sum_{\beta_k\in\{0,1\}} (-1)^{\alpha_k\cdot\beta_k}|\beta_k\rangle =$$

$$= \frac{1}{2^{n/2}} \sum_{(\beta_1,\dots,\beta_k)\in\{0,1\}^n} (-1)^{\alpha_1\beta_1\oplus\alpha_2\beta_2\oplus\dots\oplus\alpha_n\beta_n}|\beta_1\beta_2\dots\beta_n\rangle =$$

$$= \frac{1}{2^{n/2}} \sum_{\beta\in\{0,1\}^n} (-1)^{(\alpha,\beta)}|\beta\rangle.$$

Здесь в показателе мы используем обозначение (α, β) для скалярного произведения по модулю 2.

Теперь вычислим действие J на векторе $|0^n\rangle$:

$$|0^n\rangle \xmapsto{H^{\otimes n}} \frac{1}{2^{n/2}} \sum_{\alpha\in\{0,1\}^n} |\alpha\rangle \xmapsto{O_f} \frac{1}{2^{n/2}} \sum_{\alpha\in\{0,1\}^n} (-1)^{f(\alpha)}|\alpha\rangle \xmapsto{H^{\otimes n}}$$
$$\frac{1}{2^n} \sum_{\alpha\in\{0,1\}^n} \sum_{\beta\in\{0,1\}^n} (-1)^{f(\alpha)+(\alpha\cdot\beta)}|\beta\rangle =$$
$$= \frac{1}{2^n} \sum_{\beta\in\{0,1\}^n} \left(\sum_{\alpha\in\{0,1\}^n} (-1)^{f(\alpha)+(\alpha,\beta)} \right) |\beta\rangle. \quad (6)$$

Так как $(\alpha, 0^n) = 0$, то в силу (6) вероятность наблюдения исхода 0^n равна

$$\left(\frac{1}{2^n} \sum_{\alpha\in\{0,1\}^n} (-1)^{f(\alpha)} \right)^2 = \begin{cases} 1, & \text{в случае (к)}, \\ 0, & \text{в случае (б)}. \end{cases}$$

Давайте сравним квантовое решение задачи Дойча – Джоза с классическим.

Детерминированный алгоритм должен сделать не менее 2^{n-1} запросов, чтобы различить случаи (б) и (к). В самом деле, если нам известны значения функции в менее чем половине точек и все эти значения одинаковы, мы не можем различить случай константы и сбалансированной функции.

Однако вероятностный алгоритм справляется с задачей Дойча – Джоза гораздо лучше. Давайте выберем две случайные точки и запросим значения функции в них. Вероятность получения одинакового результата в случае (к) равна 1. В случае (б) вероятность получения одинакового результата равна $1/2$.

Выберем k различных точек, запросим значения функции в них и дадим ответ (к) в том случае, если все полученные значения одинаковы.

В случае (к) все значения будут одинаковы с вероятностью 1, и наш алгоритм обязательно даст правильный ответ. А в случае (б) вероятность того, что все значения одинаковы будет всего $1/2^k$. Поэтому вероятность ошибки нашего алгоритма $1/2^k$.

Таким образом, за k запросов описанный вероятностный алгоритм решает задачу Дойча – Джоза с вероятностью ошибки $1/2^k$.

Далее мы будем почти всё время иметь дело с алгоритмами, которые могут ошибаться. Для таких алгоритмов возникает еще один естественный параметр — вероятность ошибки. Обычно алгоритмы строят в предположении, что вероятность ошибки ограничена константой. Уменьшить эту константу до сколь угодно малой величины как правило можно, повторив алгоритм несколько раз. Когда это потребуется, мы уточним это неформальное соображение.

Итак, мы видим, что квантовый алгоритм решения задачи Дойча – Джоза лучше классического вероятностного лишь в константу раз (при фиксированной вероятности ошибки). Поэтому, несмотря на эффектность этого алгоритма, реальной пользы от него немного: в вопросах сложности различие на мультипликативный множитель считается несущественным.

2.6. Алгоритм Гровера: поиск иголки в стоге сена

Приведем теперь пример задачи, в которой квантовый алгоритм лучше классического более чем в константу раз.

Для этого сформулируем такую задачу поиска.

Дано: «черный ящик», который вычисляет функцию $f\colon M \to \{0,1\}$, где M — некоторое конечное множество. Заранее известно, что функция равна 1 ровно в одной точке y множества M.

Найти: точку y.

Вначале докажем, что любой вероятностный алгоритм, решающий эту задачу с фиксированной вероятностью ошибки ε, должен сделать линейное от размера множества M количество запросов.

Обозначим количество элементов в множестве M через m.

Предполагаем, что алгоритм делает ровно k запросов и смотрит на ответы после выполнения всех запросов, т. е. что алгоритм *неадаптивный*.

Это не ограничивает общности, поскольку до момента, когда найдена точка y, все ответы одинаковы и любой алгоритм ведет себя так же, как некоторый неадаптивный.

Мы оцениваем работу алгоритма для всех возможных входов. Поэтому нас интересует «худший случай». Представьте, что ответы готовит противник, цель которого — заставить вас дать неверный ответ. Противник знает ваш алгоритм и выбирает вход задачи, основываясь на этом знании и на уже сделанных запросах.

Пусть p_S вероятность того, что алгоритм запросил точки из множества $S \subset M$, $|S| = k$.

Успех алгоритма означает, что $y \in S$. Вероятность успеха:

$$p(y) = \sum_{S \ni y} p_S.$$

Существует такое y, что $p(y) \leqslant k/m$. Действительно,

$$\frac{1}{m}\sum_y p(y) = \frac{1}{m}\sum_y \sum_{S\ni y} p_S = \frac{k\sum p_S}{m} = \frac{k}{m}.$$

Вероятность успеха в худшем случае

$$p^* \leqslant \frac{k}{m}.$$

Если $p^* > 1-\varepsilon$, то

$$1-\varepsilon < p^* \leqslant \frac{k}{m}.$$

Итак, $k = \Omega(m)$, где константа в Ω зависит от выбора вероятности ошибки ε.

Теперь построим квантовый алгоритм, который решает задачу поиска за гораздо меньшее количество запросов.

И опять используем фазовый запрос (5)

$$O_y\colon |x\rangle \mapsto (-1)^{\delta(y,x)}|x\rangle.$$

Другими словами, «черный ящик» дает ответ на запрос о значении характеристической функции множества $\{y\}$.

Геометрически оператор O_y задает отражение в пространстве $\mathbb{C}(M)$ относительно гиперплоскости, ортогональной вектору $|y\rangle$.

Нам еще потребуется оператор

$$R_\psi = 2|\psi\rangle\langle\psi| - I, \quad |\psi\rangle = \frac{1}{\sqrt{m}}\sum_x |x\rangle. \tag{7}$$

С точностью до центральной симметрии (обращения знака) R_ψ подобен O_y и задает отражение относительно гиперплоскости, перпендикулярной вектору $|\psi\rangle$. Из формулы (7) легко проверить, что $R_\psi|\psi\rangle = |\psi\rangle$, а для любого $|\xi\rangle$, ортогонального ψ, получается $R_\psi|\xi\rangle = -\xi$.

Алгоритм Гровера состоит в последовательном применении оператора $G = R_\psi O_y$. Начальным состоянием выбирается $|\psi\rangle$.

Алгоритм Гровера. Выбор преобразования на шаге 2 произволен.

1°. Приготавливаем состояние $|0\rangle$.

2°. Преобразуем состояние $|0\rangle$ в $|\psi\rangle$.

3°. Применяем $\lfloor(\pi/4)\sqrt{m}\rfloor$ раз оператор G.

4°. Измеряем полученное состояние в классическом базисе.

5°. Ответ: результат измерения.

Непосредственно из описания алгоритма видно, что он выполняет $O(\sqrt{m})$ запросов. Для оценки вероятности ошибки нужно разобраться в работе этого алгоритма.

Заметим, что у оператора G итерации Гровера есть инвариантное подпространство, натянутое на векторы $|y\rangle$ и $|\psi\rangle$. Действительно, это подпространство инвариантно и для R_ψ, и для O_y.

В этом двумерном пространстве итерации Гровера применяются только к векторам с вещественными амплитудами. Поэтому достаточно посмотреть на итерацию Гровера в двумерном евклидовом пространстве, натянутом на векторы $|y\rangle$ и $|\psi\rangle$. В этом пространстве каждый из операторов R_ψ, O_y — отражение. А композиция отражений, как известно, это поворот на удвоенный угол между прямыми.

Нас интересуют оценки при $m \to \infty$. В этом случае угол между прямыми

$$\sin\vartheta = \langle\psi|y\rangle = \frac{1}{\sqrt{m}}\,, \quad \vartheta = \frac{1}{\sqrt{m}} + o(m^{-1}). \tag{8}$$

Оценки (8) показывают, что вектор состояния за одну итерацию Гровера поворачивается на угол, приблизительно равный $2/\sqrt{m}$. Поэтому после $k = \lfloor(\pi/4)\sqrt{m}\rfloor$ итераций угол между вектором состояния и вектором $|y\rangle$ станет очень маленьким (поначалу он почти прямой). Величина этого угла станет порядка $1/\sqrt{m}$. Но вероятность ошибки (т. е. наблюдения не исхода y после измерения) — это квадрат синуса угла между $|y\rangle$ и вектором состояния перед измерением. Поэтому она не превосходит $O(1/m)$.

Итак, доказана теорема.

Теорема 1. *Алгоритм Гровера решает задачу поиска среди m объектов с вероятностью ошибки $O(1/m)$ за $O(\sqrt{m})$ запросов.*

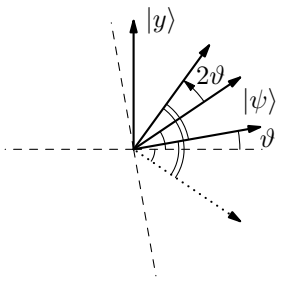


Рис. 4. Итерация Гровера

Лекция 3. Сложность булевых функций в модели запросов

Для булевых функций есть много мер сложности. Мы сейчас рассмотрим те, которые имеют отношение к вычислению значения функции с помощью запросов к «черному ящику». Будем рассматривать три вида запросов.

Пусть мы хотим вычислять значения булевой функции $x \mapsto f(x), \quad x \in \{0,1\}^n,\ f(x) \in \{0,1\}$.

Классический запрос. Посылаем «черному ящику» число k, $1 \leqslant k \leqslant n$. Получаем в ответ значение переменной x_k.

Вероятностный запрос. Посылаем число k, выбранное по некоторому вероятностному распределению (p_k), $\sum_{k=1}^n p_k = 1$, $p_k \geqslant 0$. Получаем в ответ значение переменной x_k.

Квантовый запрос. Применяем оператор

$$O_x \colon |k, b\rangle \mapsto |k, b \oplus x_k\rangle \tag{1}$$

к регистру номера переменной и регистру результата. В частности, как объяснялось в предыдущий раз, применением оператора (1) можно реализовать фазовый запрос

$$O_x \colon |k\rangle \mapsto (-1)^{x_k}|k\rangle. \tag{2}$$

Сформулируем стандартную задачу вычисления булевой функции запросами (на самом деле это три задачи — своя для каждого вида запроса).

Дано: «Черный ящик», который выдает значения переменных из некоторого набора $(x_1, \ldots, x_n)$.

Найти: значение $f(x)$.

Сложность алгоритма: количество запросов.

Вероятность ошибки: меньше $1/3$.

Уточним, что квантовый алгоритм вычисления функции в конце работы выполняет измерение своей квантовой памяти в некотором фиксированном базисе и выдает ответ в соответствии с результатом измерения. Таким образом, во всём множестве исходов W выделяется множество W_1 тех исходов, наблюдение которых влечет ответ 1, наблюдение остальных влечет ответ 0.

Определение. $\mathrm{D}(f)$ — минимальная сложность алгоритма вычисления функции f по классическим запросам.

$\mathrm{R}_{1/3}(f)$ — минимальная сложность алгоритма вычисления функции f по вероятностным запросам.

$\mathrm{Q}_{1/3}(f)$ — минимальная сложность алгоритма вычисления функции f по квантовым запросам.

Из объяснений в предыдущей лекции о моделировании классических запросов квантовыми следует, что

$$\mathrm{Q}_{1/3}(f) \leqslant \mathrm{R}_{1/3}(f) \leqslant \mathrm{D}(f)$$

для любой функции f.

Разрыв между этими мерами сложности характеризует «силу» моделей. Оказывается, что сила квантовых алгоритмов в модели запросов не так уж велика.

Теорема 1 (полиномиальная эквивалентность запросов, [30]). *Для любой* всюду определенной *булевой функции* f

$$\mathrm{D}(f) = O\big(\mathrm{Q}_{1/3}(f)^6\big).$$

Таким образом, квантовые алгоритмы могут дать не более чем полиномиальное ускорение для вычисления всюду определенной функции.

Для частичных функций это неверно, как показывает пример задачи Дойча – Джоза.

Каков наилучший разрыв между введенными мерами сложности? Этот вопрос остается открытым. Рекордным является квадратичное ускорение и рекорд достигается для очень простой функции — дизъюнкции.

Теорема 2. *Для функции* $OR = x_1 \vee x_2 \vee \cdots \vee x_n$ *выполняется*

$$\mathrm{Q}_{1/3}(OR) = \Theta(\sqrt{n}); \quad \mathrm{R}_{1/3}(OR) = \Omega(n); \quad \mathrm{D}(OR) = n.$$

Последнее утверждение этой теоремы очевидно. Дизъюнкция от нулевого набора равна 0, а от остальных — 1. Если сделать меньше n запросов противник даст нулевые ответы и затем обмануть алгоритм.

3.1. Квантовое вычисление дизъюнкции

Алгоритм со сложностью $O(\sqrt{n})$ получается модификацией алгоритма Гровера.[1)]

Опять применяем фазовый запрос

$$O_x\colon |k\rangle \mapsto (-1)^{x_k}|k\rangle.$$

Будем применять итерацию Гровера $G = R_\psi O_x$, начиная с вектора $|\psi\rangle$. Здесь, как и раньше,

$$R_\psi = 2|\psi\rangle\langle\psi| - I.$$

Если все переменные равны 0, то $G = R_\psi$ и прямая, содержащая вектор состояния, попросту не меняется. Предположим, что h переменных равны 1, а остальные равны 0. В этом случае у G по-прежнему есть инвариантное двумерное подпространство, натянутое на вектора $|\psi\rangle$ и

$$|\xi\rangle = \frac{1}{\sqrt{h}} \sum_{k:x_k=1} |k\rangle.$$

Действительно, $|\xi\rangle$ является ортогональной проекцией $|\psi\rangle$ на собственное подпространство O_x, отвечающее собственному числу -1:

$$|\psi\rangle = \frac{1}{\sqrt{n}} \sum_{k:x_k=1} |k\rangle + \frac{1}{\sqrt{n}} \sum_{k:x_k=0} |k\rangle =$$
$$= \sqrt{\frac{h}{n}}|\xi\rangle + \sqrt{\frac{n-h}{n}} \frac{1}{\sqrt{n-h}} \sum_{k:x_k=0} |k\rangle = \sqrt{\frac{h}{n}}|\xi\rangle + \sqrt{\frac{n-h}{n}}|\eta\rangle$$

и $O_x|\xi\rangle = -|\xi\rangle$, $O_x|\eta\rangle = |\eta\rangle$.

Угол поворота находится аналогично (2.8):

$$\sin\vartheta = \langle\psi|\xi\rangle = h\frac{1}{\sqrt{n}}\frac{1}{\sqrt{h}} = \sqrt{\frac{h}{n}}\,.$$

Видно, что возникает трудность: угол поворота зависит от общего числа единиц, но оно нам неизвестно. Поэтому неясно, сколько итераций Гровера требуется сделать.

[1)] Изложенный ниже алгоритм основан на статье [34].

Идея преодоления этой трудности проста: нужно «угадать» h, т. е. выбрать его случайно.

Алгоритм $Q_\vee$. Вычисление дизъюнкции по квантовым запросам с односторонней ошибкой.

1°. Повторить следующие действия 5 раз:

   а) выбрать номер переменной k по равномерному распределению на $[1; n]$;

   б) запросить x_k;

   в) если $x_k = 1$, то завершить алгоритм с результатом 1.

2°. Положить $m = \sqrt{n}$.

3°. Выбрать t по равномерному распределению на $[0; m-1]$;

4°. Приготовить состояние $|\psi\rangle$.

5°. Выполнить t итераций Гровера.

6°. Измерить полученное состояние, результат обозначим k.

7°. Закончить работу с результатом x_k.

Из описания алгоритма видно, что количество запросов в нём $O(\sqrt{n})$. Кроме того, этот алгоритм никогда не ошибается, если дизъюнкция равна 0: в этом случае результат также будет равен 0. Но возможен случай, когда дизъюнкция равна 1, а алгоритм дает ответ 0. Вероятность этого события мы и будем теперь оценивать.

Сразу заметим, что нас устроит, если вероятность ошибки в этом алгоритме будет любой константной δ, меньшей 1. Действительно, в таком случае алгоритм с вероятностью ошибки меньше ε построить легко: нужно повторить $O(1)$ раз алгоритм $Q_\vee$. После k повторений вероятность ошибки станет меньше δ^k.

При оценке вероятности ошибки рассмотрим два случая.

I. Пусть $h \geqslant n/5$. Тогда вторая стадия алгоритма потребуется с вероятностью, не превосходящей

$$\left(1 - \frac{1}{5}\right)^5 \approx e^{-1}$$

и вероятность ошибки не больше этой величины.

II. Далее предполагаем, что $0 < h < n/5$.

Будем называть «успехом» такое измерение, которое дает номер переменной, равной 1.

Лемма 1. *Вероятность успеха после t итераций Гровера, начиная с состояния $|\psi\rangle$, равна $\sin^2((2t+1)\vartheta)$.*

Доказательство. Глядя на рис. 4, замечаем, что угол между горизонтальной прямой (перпендикулярной $|\xi\rangle$) и начальным состоянием $|\psi\rangle$ равен ϑ.

Каждая итерация поворачивает вектор на угол 2ϑ. Значит, в координаты вектора после t итераций равны $(\cos((2t+1)\vartheta), \sin((2t+1)\vartheta))$. Мы считаем первым вектор $|\eta\rangle$ (на рис. 4 он горизонтален, но не отмечен), а вторым — $|\xi\rangle$.

Вероятность успеха по общему квантовому определению — квадрат модуля амплитуды, т. е. проекции на вектор $|\xi\rangle$. □

Теперь оценим вероятность успеха при случайном выборе t по равномерному распределению на отрезке $[0; m-1]$.

Лемма 2. *Вероятность успеха в указанном выше случае равна*

$$P_m = \frac{1}{2} - \frac{\sin(4m\vartheta)}{4m\sin(2\vartheta)} \ .$$

Доказательство. По формуле полной вероятности

$$P_m = \frac{1}{m}\sum_{j=0}^{m-1}\sin^2((2j+1)\vartheta) = \frac{1}{2m}\sum_{j=0}^{m-1}\Big(1-\cos((2j+1)2\vartheta)\Big).$$

Теперь свернем сумму косинусов, используя формулу

$$\sum_{j=0}^{m-1}\cos((2j+1)\varphi) = \frac{\sin(2m\varphi)}{2\sin\varphi} \ ,$$

и получим искомое выражение. □

Из леммы 2 в случае $m > 1/\sin(2\vartheta)$ следует, что вероятность успеха не меньше $1/4$.

В предположении $h < n/5$ справедлива оценка $\sin(2\vartheta) > \sin\vartheta = \sqrt{h/n}$. Поэтому

$$\sqrt{n} \geqslant \sqrt{\frac{n}{h}} = \frac{1}{\sin\vartheta} > \frac{1}{\sin(2\vartheta)} \ .$$

Поэтому вероятность успеха на шаге 6 алгоритма $Q_\vee$ не меньше $1/4$, а вероятность ошибки не больше $3/4$.

Итак, вероятность ошибки в алгоритме $Q_\vee$ не выше $3/4$. Как уже объяснялось, это означает, что вероятность ошибки можно сделать сколь угодно малой несколькими повторениями алгоритма.

Теорема 3. *Существует алгоритм, который вычисляет дизъюнкцию n переменных с вероятностью ошибки ε за $O(\sqrt{n}\log\varepsilon^{-1})$ квантовых запросов.*

3.2. Вероятностное вычисление дизъюнкции: нижняя оценка

Теперь докажем линейную нижнюю оценку сложности вероятностного вычисления дизъюнкции.

Для этого введем вспомогательную меру сложности функции.

Определение. *Чувствительность* $\mathrm{s}_x(f)$ функции f в точке x равна количеству переменных x_k, для которых $f(x) \neq f(x \oplus e_k)$. Здесь $e_k = (\underbrace{0,\ldots,0}_{k-1},1,0,\ldots,0)$.

Чувствительность $\mathrm{s}(f)$ функции f равна $\max_x \mathrm{s}_x(f)$.

Другими словами, чувствительность функции в точке указывает, сколько у этой точки соседей на булевом кубе, в которых значение функции другое.

Теорема 4. $\mathrm{s}(f) \leqslant 3\mathrm{R}_{1/3}(f)$.

Доказательство. Рассуждаем аналогично нижней оценке для числа вероятностных запросов в задаче поиска из прошлой лекции.

Пусть в x достигается максимум $\mathrm{s}_x(f) = s$, а $x_1,\ldots,x_s$ — соответствующие переменные. Если алгоритм делает $k \leqslant s/3$ запросов, то вероятность того, что одна из этих переменных пропущена, не меньше $1/3$.

Но в этом случае противник может *гарантировать* ошибку, так как $f(x) \neq$ $\neq f(x \oplus e_j)$. Поэтому вероятность ошибки при таком количестве запросов не меньше $1/3$. □

Следствие. $R_{1/3}(OR) \geqslant n/3$.

Таким образом, мы видим квадратичный разрыв между квантовой и вероятностной сложностями вычисления дизъюнкции запросами. Этот разрыв оптимален. Мы докажем это в следующем разделе. Это рассуждение будет хорошей разминкой перед доказательством теоремы о полиномиальной эквивалентности квантовых и классических запросов.

3.3. Степень многочлена, приближающего булеву функцию

Нам потребуется еще одна мера сложности для булевых функций.

Определение. Многочлен от n переменных $p\colon \mathbb{R}^n \to \mathbb{R}$ *приближает* булеву функцию $f\colon \{0,1\}^n \to \{0,1\}$, если для любого $x \in \{0,1\}^n$ выполняется

$$|p(x) - f(x)| < \frac{1}{3}\,.$$

Степенью (приближения) $\widetilde{\deg}(f)$ булевой функции f называется наименьшая степень многочлена, приближающего f.

Поскольку нас интересуют значения многочлена в точках единичного куба, то можно ограничиться только мультилинейными многочленами (степень по каждой переменной равна 1). Действительно $x^k = x$ на $\{0,1\}$.

Задача 1. Докажите, что для любой булевой функции f от n переменных существует единственный мультилинейный многочлен p степени не выше n, который точно представляет f: равенство $f(x) = p(x)$ выполняется для всех $x \in \{0,1\}^n$.

Многочленов, которые приближают f, конечно же, много.

Степень приближения булевой функции интересует нас потому, что она дает нижнюю оценку на количество квантовых запросов для вычисления функции. С другой стороны, как мы покажем далее, некоторой степенью $\widetilde{\deg}(f)$ можно оценить сверху число детерминированных запросов для вычисления функции, откуда и будет следовать теорема о полиномиальной эквивалентности.

Теорема 5. *Для любой булевой функции* $\widetilde{\deg}(f) \leqslant 2\mathrm{Q}_{1/3}(f)$.

Доказательство этой теоремы легко следует из следующей леммы.

Лемма 3. *Если алгоритм вычисления f делает d квантовых запросов, то состояние его памяти перед финальным измерением имеет вид*

$$\sum_{w \in W} \alpha_w(x)|w\rangle,$$

где w — состояния памяти алгоритма, а $\alpha_w(x)$ — (комплексные) многочлены от x степени не выше d.

Доказательство. Индукция по числу запросов.

Для $k = 0$ амплитуды не зависят от x, степень 0.

Теперь рассмотрим состояние памяти алгоритма после k запросов:

$$|\psi_k\rangle = U_k O_x U_{k-1} O_x \dots U_1 O_x |\psi_0\rangle.$$

Предположение индукции: амплитуды являются многочленами степени не выше k от значений переменных.

Что происходит при применении оператора запроса? Выделим часть вектора состояния, отвечающего базисным векторам $|w,k,0\rangle$ и $|w,k,1\rangle$ (предполагаем, что запрос действует на два последних регистра):

$$\dots + \alpha(x)|w,k,0\rangle + \beta(x)|w,k,1\rangle + \dots$$

Если $x_k = 0$, то амплитуды не изменятся, а если $x_k = 1$, то переставятся. Это можно записать одной формулой

$$\dots + (\beta(x)x_k + \alpha(x)(1-x_k))|w,k,0\rangle + (\alpha(x)x_k + \beta(x)(1-x_k))|w,k,1\rangle + \dots$$

Из этой формулы видно, что степени многочленов, представляющих амплитуды, увеличиваются не более, чем на 1.

После запроса применяется унитарный оператор U_{k+1}. Его коэффициенты определяются алгоритмом и не зависят от значений переменных. Амплитуды после преобразования выражаются как линейные комбинации амплитуд перед преобразованием. Поэтому степени многочленов, представляющих амплитуды, не увеличиваются. □

Доказательство теоремы 5. Исход w наблюдается при измерении с вероятностью $p_w(x) = |\alpha_w(x)|^2$.

Пусть W_1 — множество тех состояний памяти, в которых алгоритм выдает ответ «1». Тогда вероятность ответа 1 равна

$$p_1(x) = \sum_w p_w(x).$$

Это многочлен от x степени $\leqslant 2d$ такой, что $1 \geqslant p_1(x) > 2/3$ при $f(x) = 1$ и $1/3 > p_1(x) \geqslant 0$ при $f(x) = 0$. Т. е. $p_1(x)$ приближает $f(x)$.

Таким образом, $\deg(f) \leqslant \deg p_1(x) \leqslant 2d$. □

Замечание 1. Теорема 5 лежит в основе так называемого полиномиального метода получения нижних оценок квантовой сложности (см. [30, 37]). Степень приближения оказывается удобным приближением снизу к квантовой сложности.

Помимо этого известен другой подход к получению нижних оценок квантовой сложности: так называемый метод квантового противника, основанный на задачах неотрицательно определенного программирования [25]. Точнее говоря, было предложено несколько способов оценок в таком духе. Потом оказалось, что все эти оценки эквивалентны [62]. А в 2010 году Рейнхарт [56] доказал для одной из версий метода квантового противника, что она дает оптимальные (с точностью до мультипликативной константы) оценки квантовой сложности.

Используя метод квантового противника, Амбайнис ([26], 2006) построил семейство функций, для которых $\deg(f_k) = 2^k$, а $Q_{1/3}(f) = \Omega(2.5^k)$. Это означает, что полиномиальный метод неоптимален.

3.4. Нижняя оценка на $\mathrm{Q}_{1/3}(OR)$

Для доказательства нижней оценки на число квантовых запросов для вычисления дизъюнкции нам потребуется факт из анализа. Мы сформулируем его в нужном нам виде, имея также в виду дальнейшие приложения.

Теорема 6. *Пусть $f(x)$ — многочлен такой, что $|f(0)| < 1/3$, $|f(1)-1| < 1/3$, а $-1/3 < f(k) < 4/3$ при $2 \leqslant k \leqslant n$.*

Тогда $\deg f \geqslant \sqrt{n/6}$.

Этот факт легко следует из неравенства Маркова.

Теорема 7 (Марков, [10] 1889). *Если для многочлена от одной переменной выполняется* $|f(x)| \leqslant 1$ *на отрезке* $[-1;1]$, *то* $|f'(x)| \leqslant (\deg f)^2$ *на отрезке* $[-1;1]$.

Доказательство неравенства Маркова может быть найдено в книжках (например, в задачнике Полиа и Сегё [13]). Но его полезно вывести самостоятельно. Указание: используйте интерполяцию в корнях многочленов Чебышева и переход к тригонометрическим полиномам.

Для разминки рекомендуется следующая задача.

Задача 2. Выведите нижнюю оценку на степень многочлена (теорема 6) из неравенства Маркова.

Мы сейчас применим теорему 6 для нижней оценки квантовой сложности дизъюнкции.

Теорема 8. $\widetilde{\deg}(OR) \geqslant \sqrt{n/6}$.

Доказательство. Пусть многочлен $p(x_1, \ldots, x_n)$ приближает $OR(x_1, \ldots, x_n)$.

Используем симметризацию этого многочлена

$$p^{sym}(x_1, \ldots, x_n) = \frac{1}{n!} \sum_{\sigma \in S_n} p(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)}). \tag{3}$$

Это симметрический мультилинейный многочлен от переменных $x_1, \ldots, x_n$. Поэтому он представляется в виде линейной комбинации элементарных симметрических функций:

$$p^{sym}(x_1, \ldots, x_n) = a_0 + a_1\sigma_1(x) + \cdots + a_d\sigma_d(x),$$
$$\sigma_j(x_1, \ldots, x_n) = \sum_{S \subset \{1,\ldots,n\},\ |S|=j} \prod_{k \in S} x_k.$$

Построим симметризации многочлен от одной переменной

$$\tilde{p}(t) = (\underbrace{1, 1, \ldots, 1}_{t \text{ единиц}}, 0, \ldots, 0).$$

Это многочлен степени d, так как значения элементарной симметрической функции σ_j на таких наборах задаются многочленом степени j:

$$\sigma_j(\underbrace{1, 1, \ldots, 1}_{t \text{ единиц}}, 0, \ldots, 0) = \binom{t}{j}$$

Поскольку линейные операции не увеличивают степени многочлена, то

$$d = \deg p \geqslant \deg p^{sym} \geqslant \deg \tilde{p}.$$

Заметим, что $|\tilde{p}(0)| = |p(0^n)| < 1/3$, так как вектор 0^n сохраняется при перестановках переменных, а значение дизъюнкции в нуле равно 0.

В остальных точках значение дизъюнкции равно 1. Поэтому $|\tilde{p}(k) - 1| < 1/3$ при $1 \leqslant k \leqslant n$. Действительно, значение $\tilde{p}(k)$ равно среднему от значений $p(x)$ в точках k-го слоя булева куба (каждая точка входит $k!(n-k)!$ раз в (3), а всего точек $\binom{n}{k}$).

Поэтому применима теорема 6, откуда $\deg p \geqslant \deg \tilde{p} \geqslant \sqrt{n/6}$. □

Из теорем 5 и 8 получаем

$$\frac{1}{\sqrt{24}}\sqrt{n} \leqslant Q_{1/3}(OR) \leqslant \sqrt{n},$$
$$\frac{1}{3}\sqrt{n} \leqslant \frac{R_{1/3}(OR)}{Q_{1/3}(OR)} \leqslant \sqrt{24}\sqrt{n}.$$

Таким образом, для дизъюнкции разрыв между классической и квантовой сложностью в точности квадратичный.

Лекция 4. Полиномиальная эквивалентность классических и квантовых запросов. Коммуникационная сложность

4.1. Завершение доказательства теоремы о полиномиальной эквивалентности

Теорема о полиномиальной эквивалентности 3.1 утверждает, что $\mathrm{D}(f) = O\big(\mathrm{Q}_{1/3}(f)^6\big)$. На прошлой лекции была доказана нижняя оценка квантовой сложности через степень приближения $\widetilde{\deg}(f) \leqslant 2\mathrm{Q}_{1/3}(f)$ (теорема 3.5). Поэтому для завершения доказательства теоремы о полиномиальной эквивалентности достаточно доказать, что

$$\mathrm{D}(f) = O\big(\widetilde{\deg}(f)^6\big). \tag{1}$$

Оценка (1) получается комбинацией нескольких оценок на различные меры сложности булевых функций. Ничего «квантового» в этих оценках уже нет. Но сами по себе оценки весьма интересны, а связи между возникающими мерами сложности не вполне ясны, поэтому завершение доказательства заслуживает подробного рассказа.[1)]

Начнем с определения *сертификатной сложности*. Будем рассматривать частичные присваивания значений переменных

$$C\colon S \to \{0,1\}, \quad S \subseteq \{1,2,\ldots,n\}. \tag{2}$$

Иногда знание лишь части значений переменных позволяет определить значение функции. Разумеется, такой эффект имеет прямое отношение к алгоритмам вычисления функции по классическим запросам (решающим деревьям). Введем соответствующую меру сложности булевой функции.

Определение. *b-сертификатом* функции f называется такое частичное присваивание C, что $f(x) = b$ для всех x, согласованных с C, т. е. $x_i = C(i)$ при $i \in S$ в обозначениях (2).

Обозначим через $C_x(f)$ длину наименьшего $f(x)$-сертификата, согласованного с x.

Сертификатной сложностью функции f называется

$$C(f) = \max_x C_x(f).$$

Определим аналогично 1-*сертификатную* и 0-*сертификатную* сложности:

$$C^{(1)}(f) = \max_{x:f(x)=1} C_x(f), \quad C^{(0)}(f) = \max_{x:f(x)=0} C_x(f).$$

Заметим, что 1-сертификатная и 0-сертификатная сложности могут сильно различаться.

Пример. Для дизъюнкции $C^1(OR) = 1$, так как 1-сертификатом, согласованным с любым ненулевым набором будет любая переменная, принимающая значение 1.

С другой стороны, $C^{(0)}(OR) = C(OR) = n$, так как единственный 0-сертификат, согласованный с нулевым набором содержит все переменные.

[1)] Последующее изложение мер сложности функций и связей между ними основано на обзоре [37].

Чтобы понять, как именно соотносятся сертификатная сложность и сложность разрешающего дерева $\mathrm{D}(f)$, рассмотрим следующий алгоритм вычисления f.

Жадный алгоритм вычисления $f(x)$. (t — параметр)

1°. Строим частичное присваивание, начиная с пустого.

2°. Повторить не более t раз:

   а) выбрать совместимый с текущим присваиванием 1-сертификат C;

   б) если такого нет, закончить работу с ответом 0;

   в) запросить значения переменных из C (какие еще неизвестны);

   г) если полученные значения согласованы с C, закончить работу с результатом 1.

3°. Выбрать любой y, согласованный с текущими значениями переменных и выдать ответ $f(y)$.

К этому алгоритму необходимы пояснения. Шаги 2а и 3 описаны недетерминировано. Уточняются они естественным образом. На шаге 3 выбираем наименьший в лексикографическом порядке y, согласованный с известными к этому моменту значениями переменных (далее увидим, что этот выбор ни на что не влияет). На шаге 2а разумно выбирать сертификат покороче. Достаточно такого уточнения: найдем наименьший в лексикографическом порядке y, для которого $f(y) = 1$ и значения y согласованы с известными к этому моменту значениями x, и выберем для него самый короткий 1-сертификат.

После такого уточнения становится очевидным

Утверждение 1. *Жадный алгоритм делает не более $C^{(1)}(f)t$ запросов.*

Теперь выясним, при каких t жадный алгоритм работает корректно. Конечно, точное значение t зависит от выбора функции. Мы хотим найти удобную верхнюю оценку. Оказывается, такую оценку можно сформулировать через меру сложности, обобщающую чувствительность булевой функции. Напомним, что чувствительность в точке — это количество соседей точки на булевом кубе, значения функции в которых отличаются от значения функции в точке. Обобщение состоит в том, что переключаются не значения отдельных переменных, а целых блоков переменных.

Для удобства обозначений мы далее отождествляем множество $S \subseteq [1, \ldots, n]$ и его характеристический вектор $\chi(S)$ ($\chi_k(S) = 1 \Leftrightarrow k \in S$).

Определение. *Блочная чувствительность* $\mathrm{bs}_x(f)$ функции f в точке x равна такому максимальному b, что существует набор из b попарно непересекающихся подмножеств $B_1, \ldots, B_b$, для которых $f(x) \neq f(x \oplus B_i)$.

Блочная чувствительность $\mathrm{bs}(f)$ функции f равна $\max_x \mathrm{bs}_x(f)$.

Утверждение 2. *Жадный алгоритм работает корректно при $t \geqslant \mathrm{bs}(f)$.*

Доказательство. Прежде всего заметим, что ошибка в алгоритме возможна лишь на последнем шаге. Действительно, завершение работы на шаге 2 возможно лишь в двух случаях: если нет ни одного 1-сертификата, совместимого с текущими значениями переменных (и в этом случае значение функции, конечно же, равно 0); либо текущие значения переменных согласованы с 1-сертификатом (в этом случае значение функции равно 1 по определению 1-сертификата).

Теперь предположим, что алгоритм может выдать неправильный ответ на шаге 3.

Пусть алгоритм запрашивал сертификаты $C_1, \ldots, C_t$. Раз неправильный ответ возможен, найдутся такие y, y', согласованные со всеми известными значениями x, что $f(y) = 0$, $f(y') = 1$.

Обозначаем через B_j множество переменных, по которым различаются C_j и y, и полагаем $B_{t+1} = y \oplus y'$.

Докажем, что множества B_j непусты и не пересекаются.

Действительно, 1-сертификат не может быть полностью согласован с y, так как $f(y) = 0$. Значит, $B_j \neq \varnothing$.

Если $r \in B_j$, то $x_r = y_r \neq C_j(r)$. При $k > j$ сертификат C_k согласован со всеми известными к этому моменту значениями переменных. Поэтому либо r не входит в область определения сертификата C_k, либо $x_r = y_r = C_k(r)$. В обоих случаях $r \notin B_k$. Значит, $B_j \cap B_k = \varnothing$.

Осталось заметить, что $f(y \oplus B_j) = 1$ для $j = 1, \ldots, t+1$. Поэтому $\mathrm{bs}_y(f) \geqslant \geqslant t + 1$ и утверждение доказано. □

Из утверждения 2 следует оценка

$$\mathrm{D}(f) \leqslant \mathrm{bs}(f) C^{(1)}(f). \tag{3}$$

Чтобы получить оценку $\mathrm{D}(f)$ через степень приближения, нужно оценить каждый из сомножителей в (3) через степень приближения. Оказывается, сертификатную сложность можно оценить через чувствительность.

Лемма 1. *Для любой функции f выполнено*

$$C^{(1)}(f) \leqslant C(f) \leqslant \mathrm{s}(f)\mathrm{bs}(f) \leqslant \mathrm{bs}(f)^2.$$

Крайние неравенства в этой лемме очевидны из определений, доказывать нужно среднее.

Доказательство. Рассмотрим для некоторого x максимальный набор чувствительных блоков $B_1, \ldots, B_b$, причем каждый блок выберем минимальным по включению.

Присваивание $X\colon \cup_i B_i \xmapsto{x} \{0,1\}$ является $f(x)$-сертификатом. Действительно, если y согласован с X и $f(y) \neq f(x)$, то $\{B_j\} \cup \{y \oplus x\}$ является системой чувствительных блоков для x, что противоречит максимальности системы блоков $\{B_j\}$.

С другой стороны, размер каждого блока B_j не больше $\mathrm{s}_{x \oplus B_j}(f) \leqslant \mathrm{s}(f)$, так как

$$f(x \oplus B_j \oplus e_k) = f(x \oplus B'_j) = f(x) \neq f(x \oplus B_j).$$

Второе равенство выполняется в силу минимальности блока B_j, так как при $f(x \oplus B'_j) \neq f(x)$ блок B_j можно заменить на меньший B'_j.

Таким образом, длина $f(x)$-сертификата не превосходит $\mathrm{s}_x(f)\mathrm{bs}_x(f)$, т. е. $C(f) \leqslant \mathrm{s}(f)\mathrm{bs}(f)$. □

Лемма 2 (Нисан – Сегеди [53], 1994). *Для любой функции f выполнено* $\mathrm{bs}(f) \leqslant \leqslant 6\widetilde{\deg}(f)^2$.

Оценка леммы напоминает нижнюю оценку для квантовой сложности вычисления дизъюнкции. Это неслучайно: доказательство основано на той же нижней оценке степени многочлена (теорема 3.6).

Доказательство. Рассмотрим такой набор значений переменных a, на котором достигается максимум блочной чувствительности. Обозначим через $B_1, \ldots, B_b$ соответствующие блоки. Считаем без ограничения общности, что $f(a) = 0$.

Предположим, что мультилинейный многочлен $p(x_1, \ldots, x_n)$ степени d приближает $f(x)$. Построим новый многочлен $q(y_1, \ldots, y_b)$, заменяя переменные x_j в многочлене p по правилам

– если $a_j = 0$ и $j \in B_k$, то $x_j := y_k$;

– если $a_j = 1$ и $j \in B_k$, то $x_j := 1 - y_k$;

– в противном случае $x_j := a_j$.

Полученный многочлен $q(y_1, \ldots, y_b)$ удовлетворяет следующим свойствам:

1. $\deg q(y) \leqslant d$ (замена переменной на многочлен степени $\leqslant 1$ не увеличивает степень многочлена);
2. многочлен $q(y)$ — мультилинейный (поскольку таким является $p(x)$);
3. $-1/3 < q(y) < 4/3$ при $y \in \{0,1\}^n$ (для таких y значение $q(y)$ совпадает со значением p в некоторой точке булева куба);
4. $|q(0) = p(a)| < 1/3$;
5. $q(e_j) = p(a \oplus B_j)$, $f(a \oplus B_j) = 1$, поэтому $|q(e_j) - 1| < 1/3$.

Далее, как и в доказательстве теоремы 3.8 рассмотрим симметризацию

$$q^{sym}(y_1, \ldots, y_b) = \frac{1}{b!} \sum_{\sigma \in S_b} q(y_{\sigma(1)}, y_{\sigma(2)}, \ldots, y_{\sigma(n)})$$

и многочлен от одной переменной

$$\tilde{q}(t) = q^{sym}(\underbrace{1, 1, \ldots, 1}_{t \text{ единиц}}, 0, \ldots, 0).$$

Для многочлена $\tilde{q}$ выполнены условия теоремы 3.6. Действительно, $|\tilde{q}(0) = = q(0^b)| < 1/3$. Усреднение по наборам y веса 1 (с одной единицей) даст величину, которая лежит на интервале $(2/3; 4/3)$, так как значения $q(y)$ в этих точках равны 1. При $2 \leqslant k \leqslant b$ значения $\tilde{q}(k)$ попадают в интервал $(-1/3; 4/3)$, поскольку в этот интервал попадают все значения $q(y)$ на булевом кубе.

Поэтому по теореме 3.6 о нижней оценке степени многочлена $\deg \tilde{q} \geqslant \sqrt{b/6}$, откуда и следует оценка леммы. □

Итак, что же мы получили? Из утверждения 2 мы получили оценку

$$\mathrm{D}(f) \leqslant \mathrm{bs}(f) C^{(1)}(f).$$

Лемма 1 уточняет, что

$$\mathrm{D}(f) \leqslant \mathrm{bs}(f)^3,$$

а лемма 2 —

$$\mathrm{D}(f) \leqslant 216 \widetilde{\deg}(f)^6.$$

Наконец, используя оценку $\widetilde{\deg}(f) \leqslant 2\mathrm{Q}_{1/3}(f)$ теоремы 3.5, получаем

$$\mathrm{D}(f) \leqslant 13824 \mathrm{Q}_{1/3}(f)^6.$$

Это завершает доказательство теоремы о полиномиальной эквивалентности.

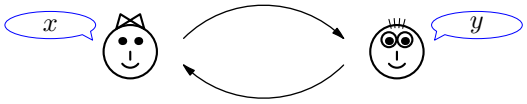


Рис. 1. Основная модель коммуникации: детерминированное вычисление

4.2. Коммуникационная сложность

Перейдем теперь к рассмотрению вопросов, относящихся к сложности распределенных вычислений. Здесь нас интересует оценка количества информации, которым должны обменяться участники вычисления, чтобы решить поставленную перед ними задачу. Рассмотренные выше алгоритмы «черного ящика» также можно отнести в эту категорию, хотя нас интересовало даже не количество информации, а просто число раундов общения исполнителя алгоритма и «черного ящика», который владеет существенной для решения задачи информацией. Нас по-прежнему не интересуют ресурсы, которые тратят участники вычисления: мы оцениваем только количество передаваемой информации.

Начнем с основной для коммуникационной сложности модели.

Два участника вычисления, которых по традиции называют *Алиса* и *Боб*, владеют двумя непересекающимися частями входной информации и должны, обменявшись как можно меньшим количеством информации, решить задачу (см. рис. 1).

Стандартное уточнение такого вопроса имеет следующий вид. Считаем, что Алиса имеет x, а Боб — y, где $x, y \in \{0,1\}^n$ — двоичные строки длины n. Задача Алисы и Боба: вычислить $f(x,y)$, где $f\colon \{0,1\}^{2n} \to \{0,1\}$ — некоторая функция от $2n$ переменных.

Алиса и Боб по очереди посылают друг другу сообщения (двоичные строки). Каждое сообщение зависит от всей предыстории и является некоторой детерминированной функцией от входа и уже отправленных сообщений. После некоторого количества раундов Алиса (для определенности) должна сказать значение функции f.

Набор функций, которые по которым вычисляются сообщения, называется *коммуникационным протоколом*.

Наименьшее количество битов в сообщениях такого протокола, который правильно вычисляет $f(x,y)$ для всех пар x, y, и есть *детерминированная коммуникационная сложность* $C(f)$.

Раз есть детерминированная сложность, то неизбежно появляются вероятностная, а потом и квантовая. При этом возможны разные уточнения по поводу использования вероятностного и квантового ресурсов.

На рис. 2 изображена основная модель вероятностной коммуникации.

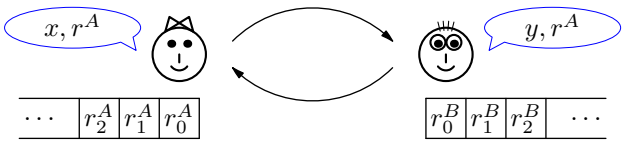


Рис. 2. Основная модель коммуникации: вероятностное вычисление, частные генераторы случайности

В отличие от предыдущего случая, Алиса и Боб могут теперь использовать генераторы случайных чисел. Причем в данной постановке предполагается, что у каждого участника есть свой персональный генератор и доступа к нему у другого участника нет. На английском эти предположения выражаются кратко

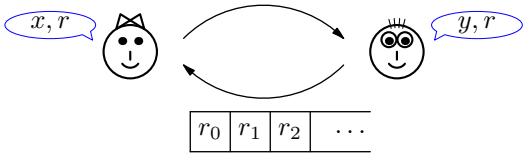


Рис. 3. Основная модель коммуникации: вероятностное вычисление, общий генератор случайности

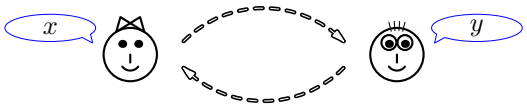


Рис. 4. Основная модель коммуникации: квантовое вычисление

private coin.

Теперь сообщения зависят не только от части входа, имеющейся у участника, и предыдущих сообщений, но и от значений r_k^* случайных битов этого участника.

Вероятностная коммуникационная сложность $R_\varepsilon(f)$ — это наименьшее количество передаваемых битов в вероятностном протоколе, гарантирующем вычисление $f(x, y)$ с ошибкой не более ε для любых x, y.

Здесь, как и ранее, можно легко понять, что многократным повторением протокола можно уменьшить вероятность ошибки от любой константы $< 1/2$ до любого желаемого ε. Поэтому обычно полагают $\varepsilon = 1/3$.

Вторая вероятностная модель коммуникации предполагает, что генератор случайности общий для участников (по-английски public coin), см. рис. 3.

Фактически в этой модели предполагается, что Алиса и Боб имеют общую информацию, хотя эта информация и представляет из себя набор случайных битов.

Соответствующую коммуникационную сложность обозначим $R_\varepsilon^{\mathrm{pub}}(f)$.

Теперь перейдем к квантовой коммуникации. У Алисы и Боба теперь квантовая память и обмениваются они также квантовыми сообщениями (рис. 4).

Квантовая коммуникационная сложность $Q_\varepsilon(f)$*:* наименьшая длина квантового протокола, гарантирующего вычисление $f(x, y)$ с ошибкой не более ε для любых x, y.

Как описать квантовый протокол формально? В самом общем виде это выглядит так. Квантовая память представляется набором кубитов, т. е. фазовое пространство имеет вид $(\mathbb{C}^2)^{\otimes N}$, где N — общее количество кубитов.

Действия участников описываются разложимыми унитарными операторами, которые затрагивают лишь часть кубитов, а на остальных действуют тождественно. Формально такой оператор записывается как $U[S]$, $S \subseteq \{1, 2, \ldots, N\}$. Если S — это начальный отрезок ряда $\{1, 2, \ldots, N\}$, действие оператора $U[S]$ на разложимых векторах записывается как

$$U[S]|s, t\rangle = U|s\rangle \otimes |t\rangle, \tag{4}$$

а на остальные продолжается по линейности. Если S произвольное множество, то записать такую же компактную формулу, как (4), не представляется возможным, но суть та же самая.

Упражнение 1. Напишите в вычислительном базисе матрицу оператора c-NOT$[1, 3]$, который действует на трех кубитах. Оператор c-NOT — это оператор обратимого копирования: c-NOT: $|x, y\rangle \mapsto |x, x \oplus y\rangle$.

Квантовый протокол — это разбиение кубитов на две группы $\{1, 2, \ldots, N\} = A \dot\cup B$; последовательность операторов

$$U_1[S_1], U_2[S_2], \ldots, U_\ell[S_\ell], \quad S_1 \subseteq A$$

(в некотором противоречии с древней китайской мудростью, мы считаем, что нечетные операторы исполняет Алиса, а четные — Боб); и указание на кубит результата (его значение после измерения будет ответом протокола на вопрос задачи). В начале работы кубиты из множества A находятся в состоянии $|x, 0\rangle$ (базисный вектор вычислительного базиса), а кубиты из множества B — в состоянии $|0, y\rangle$. Таким образом, начальное состояние описывается вектором из вычислительного базиса и, в частности, разложимо.

Как определить количество кубитов, которые пересылаются в таком протоколе? Кубиты из множества A изначально находятся у Алисы, а кубиты из множества B — у Боба. Множества S_j определяют, какие кубиты пересылаются: чтобы Алиса могла подействовать на кубит, который перед этим действием находится у Боба, нужно этот кубит ей переслать, и наоборот. Считаем, что первой передает Алиса, поэтому и накладываем ограничение $S_1 \subseteq A$; если его убрать, то единственное, что изменится — Боб получит возможность говорить первым. Формально это изменит некоторые утверждения (см. задачу 2 ниже), но изменения непринципиальны.

Длиной протокола считаем количество пересланных кубитов.

Неудобство такого протокола в том, что количество пересылаемых кубитов входит в него слишком неявно. Есть стандартное описание квантовых протоколов коммуникации, которое выглядит ограничительно, но на самом деле эквивалентно общему случаю.

В этом частном случае кубиты раз и навсегда делятся на три группы: кубиты Алисы, кубиты Боба и кубит сообщения. Каждый из участников имеет возможность действовать на свои кубиты и кубит сообщения (см. рис. 5).

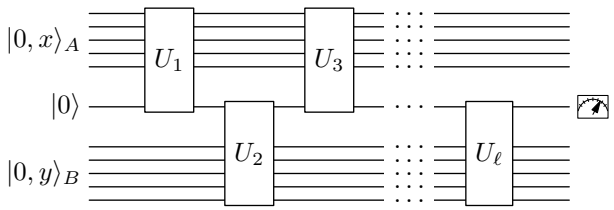


Рис. 5. Квантовый коммуникационный протокол с передачей по одному кубиту

На рис. 5 мы используем широко распространенный в квантовой информатике способ записи композиций унитарных операторов в тензорном пространстве. Горизонтальные линии соответствуют кубитам (тензорным сомножителям), оператор изображается прямоугольником, накрывающим то множество кубитов, на которое действует оператор (позже мы расширим эту символику).

Начальное состояния для этого протокола $|x, 0\rangle_A \otimes |0\rangle \otimes |0, y\rangle_B$. Результат работы определяется значением кубита сообщения (что указано на рисунке пиктограммой измерительного прибора).

Задача 2. Докажите, что общий протокол можно моделировать протоколом с передачей лишь по одному выделенному кубиту, причем длина этого протокола ограничена удвоенной длиной исходного протокола общего вида.

Указание: перестановка кубитов (тензорных сомножителей) — унитарный оператор.

В квантовом случае есть также аналог вероятностных протоколов с общим генератором случайности. Это коммуникация с предварительной сцепленностью, схема которой изображена на рис. 6 для модели передачи по одному кубиту. Как видно из рисунка, в данном случае начальное состояние имеет вид

$$|x\rangle \otimes |\psi\rangle \otimes |y\rangle, \tag{5}$$

где ψ — произвольное состояние, не обязательно разложимое. Такая сцепленность начальных состояний дает, вообще говоря, дополнительные возможности. Квантовой коммуникационной сложностью с предварительной сцепленностью $Q_\varepsilon^{\mathrm{ent}}(f)$ будем называть длину наименьшего протокола вычисления f в такой модели.

Заметим, что обычно под предварительной сцепленностью обычно понимают частный случай приведенного выше протокола, когда состояние ψ в (5) имеет вид

$$|\psi\rangle = \big(\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\big)^{\otimes m} \otimes |0\rangle, \tag{6}$$

причем первый кубит в каждом сомножителе достается Алисе, второй — Бобу, а последний кубит — это кубит сообщения (на рисунке он расположен между кубитами Алисы и Боба). Состояние двух кубитов

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

называется ЭПР-парой в честь Эйнштейна, Розена и Подольского, которые пытались опровергнуть квантовую механику, изучая эти состояния. ЭПР-пары являются одним из фундаментальных ресурсов в квантовой теории информации. Для квантовых алгоритмов роль и значение ЭПР-пар и вообще сцепленности меньше, поэтому в данном курсе мы о них не будем говорить подробно. Но в вопросах квантовой коммуникации они возникают совершенно неизбежно.

Между введенными величинами имеются очевидные соотношения.

Упражнение 3. Докажите, что для любой f выполнено

$$Q_\varepsilon^{\mathrm{ent}}(f) \leqslant R_\varepsilon^{\mathrm{pub}}(f) \leqslant R_\varepsilon(f) \leqslant D(f) \leqslant n,$$
$$Q_\varepsilon^{\mathrm{ent}}(f) \leqslant Q_\varepsilon(f) \leqslant R_\varepsilon(f).$$

Два неравенства приходится писать, поскольку связь между квантовой коммуникационной сложностью и вероятностной сложностью с общим генератором неясна. Видимо, открытым является вопрос, для любой ли функции f выполнено

$$Q_\varepsilon(f) \leqslant R_\varepsilon^{\mathrm{pub}}(f). \tag{7}$$

Замечание 1. Известно, впрочем, что (7) не выполняется в более простой модели коммуникации — модели одновременной передачи сообщений (SMP). Об этой модели мы скажем ниже, но нарушение (7) рассматривать не будем. Желающие могут ознакомиться с работой [43] (2004).

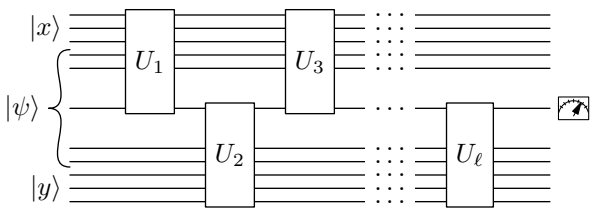


Рис. 6. Квантовая коммуникация с предварительной сцепленностью

Одной из основных открытых проблем в квантовой информатике является вопрос о полиномиальной эквивалентности классической (вероятностной) и квантовой коммуникационной сложности. Для ограниченных моделей коммуникации (в частности, SMP, о чем пойдет речь ниже) ответ на него отрицательный: квантовые протоколы оказываются экспоненциально короче. Но случай основной схемы, описанный выше по-прежнему открыт. Интригует здесь близость коммуникационной сложности к алгоритмам «черного ящика», причем не только формальная, но и по существу. Мы увидим ниже, как использовать идеи алгоритма Гровера для построения квантовых протоколов коммуникации. Известные нижние оценки для конкретных функций также напоминают рассуждения из теоремы о полиномиальной эквивалентности.

4.3. Задача о пересечении множеств

Рассмотрим пример задачи коммуникационной сложности для конкретной функции. Эта функция задается следующим образом:

$$\mathrm{DISJ}(x,y) = \begin{cases} 1, & \text{если } x_j y_j = 0 \text{ для любого } j, \\ 0, & \text{в противном случае.} \end{cases}$$

Если считать, что единицы в наборах x, y кодируют элементы двух подмножеств, то спрашивается о пустоте пересечения этих множеств.

Коммуникационная сложность для функции DISJ изучена полностью. Приведем соответствующие результаты.

Теорема 1 (Разборов [55], 1992). $R_\varepsilon(\mathrm{DISJ}) = \Omega(n)$.

Теорема 2 (Buhrman, Cleve, Wigderson [36], 1998). $Q_\varepsilon(\mathrm{DISJ}) = O(\sqrt{n}\log n)$.

Замечание 2. Впоследствии Ааронсон и Амбайнис [18] усилили этот результат до $Q_\varepsilon(\mathrm{DISJ}) = O(\sqrt{n})$.

Теорема 3 (Разборов [14], 2002)). $Q_\varepsilon(\mathrm{DISJ}) = \Omega(\sqrt{n})$.

Самый простой из этих результатов — теорема Бурмана, Клива и Вигдерсона. Идея их протокола состоит в том, чтобы использовать алгоритм Гровера.

Протокол вычисления $\neg\mathrm{DISJ}$**.** (Ясно, что протоколы вычисления функции и ее отрицания имеют одинаковую коммуникационную сложность.)

1°. Алиса выполняет шаги алгоритма вычисления дизъюнкции $\bigvee_{x_j=1} y_j$.

2°. Вместо запроса к «черному ящику» она общается с Бобом:

   а) Алиса посылает регистр адреса Бобу;

   б) Боб применяет оператор фазового запроса $O_y\colon |k\rangle \mapsto (-1)^{y_k}|k\rangle$ и возвращает регистр Алисе.

В этом протоколе передается $O(\sqrt{n}\log n)$ кубитов, так как количество запросов $O(\sqrt{n})$, как мы уже проверяли, а длина каждого запроса $O(\log n)$.

Лекция 5. Нижние оценки квантовой коммуникационной сложности. Другие модели коммуникации

5.1. Нижние оценки квантовой коммуникационной сложности

В основе построения нижних оценок квантовой коммуникационной сложности лежит лемма, которая описывает структуру квантового состояния после ℓ раундов обмена информацией по одному кубиту (см. рис. 4.5).[1)] Мы ограничимся только протоколами без предварительной сцепленности.

Перед началом работы состояние системы $|x,0\rangle_A \otimes |0\rangle \otimes |0,y\rangle_B$ принадлежит вычислительному базису и, в частности, является разложимым вектором относительно тензорного произведения $A \otimes C \otimes B$, где A — пространство состояний кубитов Алисы, B — Боба, а C — кубита сообщения. В дальнейшем состояние системы перестает быть разложимым. Однако оказывается, что после небольшого числа раундов коммуникации состояние системы является суммой сравнительно небольшого числа разложимых векторов.

Лемма 1 (разложение Яо – Кремера). *Состояние квантовой памяти после ℓ раундов общения с передачей по одному кубиту представляется в виде*

$$|\psi\rangle = \sum_{m \in \{0,1\}^\ell} |\alpha_m\rangle_A \otimes |m_\ell\rangle \otimes |\beta_m\rangle_B, \tag{1}$$

где $|\alpha_m| \leqslant 1$, $|\beta_m| \leqslant 1$, а m_ℓ — последний бит двоичной строки m.

Доказательство. Индукция по длине протокола.

Как уже говорилось, начальное состояние $|x,0\rangle_A \otimes |0\rangle \otimes |0,y\rangle_B$ разложимо.

Пусть после ℓ раундов состояние имеет вид

$$|\psi\rangle = \sum_{m \in \{0,1\}^\ell} |\alpha_m\rangle_A \otimes |m_\ell\rangle \otimes |\beta_m\rangle_B.$$

Рассмотрим случай четного ℓ (следующий ход Алисы). Получаем:

$$\begin{aligned}
&U_{\ell+1}|\alpha_m\rangle_A \otimes |m_\ell\rangle = |\alpha'_{m0}\rangle \otimes |0\rangle + |\alpha'_{m1}\rangle \otimes |1\rangle,\\
&|\alpha'_{m0}|^2 + |\alpha'_{m1}|^2 = |\alpha_m|^2 \leqslant 1,\\
&U_{\ell+1}|\psi\rangle = \sum_{m \in \{0,1\}^{\ell+1}} |\alpha'_m\rangle_A \otimes |m_\ell\rangle \otimes |\beta'_m\rangle_B,\\
&\text{где } |\beta'_{m0}\rangle = |\beta'_{m1}\rangle = |\beta_m\rangle.
\end{aligned}$$

В случае хода Боба вычисления аналогичны. □

Теперь введем матрицу положительных ответов P. Ее элементы p_{xy} — это вероятности ответа 1 на входе x, y, где $x \in X \subseteq \{0,1\}^n$, $y \in Y \subseteq \{0,1\}^n$ (бывает удобно ограничиться не всеми возможными парами входных данных — отсюда возникают множества X, Y).

Из разложения Яо – Кремера можно заметить, что матрица положительных ответов для короткого протокола имеет весьма ограничительный вид.

Лемма 2. *Для протокола длины ℓ выполняется*

$$p_{xy} = \sum_{k=0}^{2^{2\ell-2}-1} a_{x,k} \cdot b_{y,k}, \quad |a_{x,k}| \leqslant 1,\ |b_{x,k}| \leqslant 1.$$

[1)] Изложение основано на лекции О. Регева в университете Тель-Авива, 2006.

Доказательство. Запишем разложение Яо – Кремера

$$|\psi\rangle = \sum_{m\in\{0,1\}^\ell} |\alpha_m\rangle_A \otimes |m_\ell\rangle \otimes |\beta_m\rangle_B =$$
$$= \sum_{m\in\{0,1\}^{\ell-1}} |\alpha_{m0}\rangle_A \otimes |0\rangle \otimes |\beta_{m0}\rangle_B + \sum_{m\in\{0,1\}^{\ell-1}} |\alpha_{m1}\rangle_A \otimes |1\rangle \otimes |\beta_{m1}\rangle_B \quad (2)$$

Из (2) вероятность наблюдения исхода 1

$$p_{xy} = \sum_{m',m''} \langle\alpha_{m'1}|\alpha_{m''1}\rangle \cdot \langle\beta_{m'1}|\beta_{m''1}\rangle,$$

что и означает утверждение леммы. □

Лемма 2 утверждает, что матрица положительных ответов для протокола длины ℓ является суммой не более чем $2^{2\ell-2}-1$ матриц ранга 1, т. е. матриц вида $|a\rangle\langle b|$, причем векторы a, b имеют компоненты, по модулю не превосходящие 1.

Дальнейшее получение нижних оценок основано на оценках норм матриц указанного вида. Нам потребуется сразу несколько матричных норм.

Прежде всего введем скалярное произведение операторов (или матриц). Оно называется произведением Фробениуса:

$$\langle A, B\rangle = \mathrm{Tr}(A^\dagger B). \quad (3)$$

Из линейности следа легко следует, что это действительно эрмитово произведение (т. е. полулинейная форма).

Нормой Фробениуса называется норма относительно этого скалярного произведения:

$$\|A\|_F = \sqrt{\langle A, A\rangle}. \quad (4)$$

Если зафиксировать базис в пространстве операторов, т. е. рассматривать матрицы, то норма Фробениуса — это корень квадратный из суммы квадратов модулей матричных элементов. Такое определение совпадает с определением нормы вектора в координатном пространстве.

Наиболее естественная норма для операторов имеет, однако, другой вид (*операторная норма*):

$$\|A\| = \max_{x:|x|=1} |Ax|. \quad (5)$$

Помимо этих двух норм в квантовой информатике оказывается крайне полезной еще одна, так называемая *следовая норма*:

$$\|A\|_{\mathrm{tr}} = \max_{\|X\|=1} \big|\langle A, X\rangle\big| \quad (6)$$

Утверждение 1. *Норма Фробениуса, операторная норма и следовая норма удовлетворяют свойствам нормы:*

(1) $\|x\| \geqslant 0$,

(2) $\|\lambda x\| = |\lambda| \cdot \|x\|$,

(3) $\|x + y\| \leqslant \|x\| + \|y\|$.

Доказательство этого утверждения оставляется в качестве упражнения.

Для работы с нормами операторов полезна теорема о сингулярном разложении.

Теорема 1 (теорема о сингулярном разложении). *Любой оператор представляется в виде*

$$A = \sum_{k=1}^{r} s_k |\xi_k\rangle\langle\psi_k| \quad \textit{или для матриц} \quad A = UDV,$$

где $\{\xi_k\}$, $\{\psi_k\}$ — *ортонормированные системы векторов,*
$s_k \geqslant 0$ — сингулярные числа, r — ранг *оператора,*
U, V — *унитарные матрицы,* D — *неотрицательная диагональная.*

Теорема о сингулярном разложении выводится из свойств эрмитовых операторов.

Упражнение 1. Выведите теорему о сингулярном разложении из теоремы о существовании для эрмитова оператора ортонормированного базиса из собственных векторов.

Указание: используйте тот факт, что для любого оператора A оператор $A^\dagger A$ эрмитов.

Выполнив это упражнение, вы поймете, что сингулярные числа оператора A — это квадратные корни из собственных чисел оператора $A^\dagger A$.

Введенные выше операторные нормы выражаются через сингулярные числа. Нам потребуется выражение для следовой нормы.

Утверждение 2. $\|A\|_{\mathrm{tr}} = \sum_k s_k$.

Доказательство. Используем теорему о сингулярном разложении. Пусть $A = UDV$.

Тогда

$$\langle A, UV\rangle = \mathrm{Tr}(A^\dagger UV) = \mathrm{Tr}(V^\dagger D U^\dagger UV) = \mathrm{Tr}(D) = \sum_k s_k.$$

С другой стороны,

$$\big|\,\mathrm{Tr}(A^\dagger X)\big| \leqslant \sum_k s_k \big|\,\mathrm{Tr}(|\psi_k\rangle\langle\xi_k|X)\big| = \sum_k s_k \big|\,\mathrm{Tr}(\langle\xi_k|X|\psi_k\rangle)\big| =$$
$$= \sum_k s_k \big|\langle\xi_k|X|\psi_k\rangle\big| \leqslant \sum_k s_k \big|X|\psi_k\rangle\big| \leqslant \|X\| \sum_k s_k.$$

□

Лемма 2 позволяет оценить следовую норму матрицы положительных ответов P.

Лемма 3. $\|P\|_{\mathrm{tr}} \leqslant 2^{2\ell-2}\sqrt{|X|\cdot|Y|}$.

Доказательство. Из леммы 2 получаем

$$p_{xy} = \sum_{k=0}^{2^{2\ell-2}-1} a_{x,k}\cdot b_{y,k}, \quad |a_{x,k}| \leqslant 1,\ |b_{x,k}| \leqslant 1.$$

Введем векторы $\boldsymbol{a}_k = (a_{x,k})$, $\boldsymbol{b}_k = (b^*_{y,k})$. Тогда $P = \sum_k |\boldsymbol{a}_k\rangle\langle\boldsymbol{b}_k|$ и

$$\|P\|_{\mathrm{tr}} \leqslant \sum_{k=0}^{2^{2\ell-2}-1} \big\||\boldsymbol{a}_k\rangle\langle\boldsymbol{b}_k|\big\|_{\mathrm{tr}} = \sum_{k=0}^{2^{2\ell-2}-1} |\boldsymbol{a}_k|\cdot|\boldsymbol{b}_k| \leqslant 2^{2\ell-2}\sqrt{|X|\cdot|Y|}.$$

□

В некоторых случаях непосредственное применение леммы 3 дает нижнюю оценку на квантовую коммуникационную сложность.

Рассмотрим такой пример. Функция скалярного произведения (над полем из двух элементов) определяется как

$$\mathrm{IP}(x,y) = \bigoplus_{k=1}^{n} x_k y_k.$$

Утверждение 3. $Q_{1/3}(\mathrm{IP}) = \Omega(n)$.

Доказательство. Полагаем $X = Y = \{0,1\}^n$.

Напишем матрицу скалярных произведений

$$M = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\otimes n}. \tag{7}$$

Более точно, $M_{xy} = (-1)^{\mathrm{IP}(x,y)}$ (проверьте это, выписав матричный элемент (7), или посмотрите решение задачи Дойча – Джоза).

Пусть P — матрица положительных ответов для квантового протокола, вычисляющего $\neg\mathrm{IP}$ с ошибкой не более $1/3$. (Коммуникационная сложность функции и ее отрицания одинакова.) Это означает, что из $M_{xy} = 1$ следует $P_{xy} > 2/3$, а из $M_{xy} = -1$ следует $P_{xy} < 1/3$. Но тогда

$$\langle P, M\rangle = \sum_{x,y} P_{xy} M_{xy} \geqslant \frac{2^{2n}}{6}\,. \tag{8}$$

Чтобы применить лемму 3, нужно оценить операторную норму M. Это легко сделать, если заметить, что

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \sqrt{2}H,$$

а оператор Адамара H — унитарный, т. е. его норма равна 1.

Норма тензорной степени равна степени нормы (упражнение), откуда заключаем, что $\|M\| = 2^{n/2}$ и

$$\langle P, M\rangle \leqslant \|P\|_{\mathrm{tr}} \|M\| = 2^{n/2} \|P\|_{\mathrm{tr}} \leqslant 2^{3n/2} 2^{2\ell-2}.$$

Комбинируя с (8), получаем $2\ell - 2 \geqslant n/2 + c$. □

5.2. О нижней оценке для функции пересечения множеств

На прошлой лекции уже приводилась оценка Разборова для функции DISJ (которая, с точностью до отрицания, совпадает с функцией пересечения множеств) $Q_{1/3}(\mathrm{DISJ}) = \Omega(\sqrt{n})$. На этой функции достигается самый большой известный разрыв между квантовой и классической коммуникационными сложностями.

Разборов получил эту оценку, применив лемму 3 более изощренным способом. Сейчас известно альтернативное доказательство Шерстова, которое во многих отношениях выглядит более привлекательным. Я ограничусь только изложением идей исходного доказательства, про метод Шерстова можно прочитать в его диссертации [57], которая содержит и много других результатов, полученных применением Фурье-анализа.

Для применения леммы 3 выберем в качестве X и Y множество слов длины n и веса $n/4$ (вес — количество единиц).

После этого подберем матрицы μ_s, $0 \leqslant s \leqslant n/4$, со следующими свойствами:

1. $\langle P, \mu_0 \rangle \in (2/3; 1]$;
2. $\langle P, \mu_s \rangle \in [0; 1/3)$ для $s = 1, \ldots, n/4$;
3. для любого $d \leqslant n/4$ существует многочлен p степени d такой, что

$$|p(s) - \langle P, \mu_s \rangle| \leqslant \frac{\|P\|_{\mathrm{tr}}}{\binom{n}{n/4} 2^{d/4}} \quad \text{для } 0 \leqslant s \leqslant n/8. \tag{9}$$

Полагаем $d = 8\ell + 8$ и применяем лемму 3 к правой части (9), получаем

$$|p(s) - \langle P, \mu_s \rangle| \leqslant \frac{2^{2\ell-2}\binom{n}{n/4}}{\binom{n}{n/4} 2^{2\ell-2}} = \frac{1}{2^4}.$$

Но это означает, что к многочлену $p(\cdot)$ можно применить теорему о нижней оценке степени многочлена (теорема 3.6), из которой следует $d = \Omega(\sqrt{n})$ и $\ell = \Omega(\sqrt{n})$.

Чтобы реализовать этот план, нужно найти подходящие матрицы μ_s. Они выбираются следующим образом:

$$\mu_s = \frac{1}{\binom{n}{n/4}\binom{n/4}{s}\binom{n-n/4}{n/4-s}} J_{n,n/4,s},$$

где $J_{n,k,s}$ — матрица из схемы отношений Джонсона

$$J_{n,k,s}(x,y) = \begin{cases} 1, & \text{если } |x \cap y| = s, \\ 0, & \text{в противном случае,} \end{cases} \qquad x, y \in \{0,1\}^n;\ |x| = |y| = k.$$

Фактически матрица μ_s задает равномерное распределение на тех парах входных данных x, y, для которых размер пересечения в точности равен s. Поэтому для протокола, вычисляющего $\mathrm{DISJ}(x, y)$, выполняются условия 1–2:

$$\langle P, \mu_0 \rangle \in (2/3; 1], \quad \langle P, \mu_s \rangle \in [0; 1/3) \text{ для } s = 1, \ldots, n/4.$$

Для проверки условия 3 нужно изучить спектры матриц из схемы отношений Джонсона (и это было сделано задолго до возникновения квантовой информатики).

Я лишь намечу общий план рассуждения, оставляя выполнение его шагов в качестве самостоятельной работы (не всегда простой, но весьма полезной).

Упражнение 2. Матрицы $J_{n,k,s}$ попарно коммутируют.

Из этого факта следует, что у матриц $J_{n,k,s}$ есть общая система собственных пространств $E_0, \ldots, E_k$. Нумерация собственных пространств не произвольная, как показывает следующая, довольно трудная, задача.[2)]

Задача 3. Докажите, что собственное число $\lambda_{s,t}$ матрицы $J_{n,k,s}$, отвечающее пространству E_t, выражается формулой

$$\lambda_{s,t} = \sum_r (-1)^{t-r} \binom{t}{r} \binom{k-r}{s-r} \binom{n-k-t+r}{k-s-t+r}.$$

Задача 4. Докажите, что соответствующие собственные числа $\lambda_{s,t}(\mu_s)$ матриц μ_s задаются многочленом от s степени t.

Задача 5. Докажите, что при $k \leqslant n/4$ и $s \leqslant k/2$ выполняется неравенство

$$|\lambda_{s,t}(\mu_s)| \leqslant \binom{n}{k}^{-1} 2^{-t/4}.$$

Задача 6. Получите условие 3 из приведенных выше фактов.

Указание: перейдите к базису, в котором матрицы $J_{n,k,s}$ диагонализуются.

[2)] Подробное решение этой задачи написано Д. Кнутом [50].

5.3. Модель коммуникации SMP

Как уже было сказано, вопрос о возможности сверхполиномиального разрыва между классической и квантовой коммуникационными сложностями в основной модели коммуникации пока открыт.

Для нескольких ограниченных моделей коммуникации доказано существование экспоненциального разрыва. Рассмотрим одну из таких моделей. Это модель одновременной передачи сообщений (SMP).

Теперь у нас три действующих лица. Алиса знает x, а Боб знает y. Они не могут общаться непосредственно и должны передать данные Чарли, чтобы тот вычислил $f(x, y)$. Аналогично предыдущему формулируются вероятност-

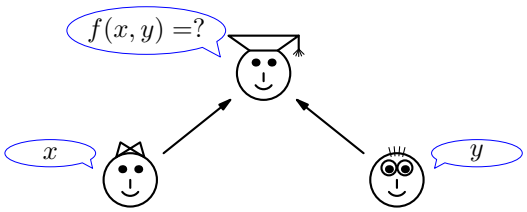


Рис. 1. Одновременная передача сообщений

ная модель (с частными или общими генераторами) и квантовая (с предварительной сцепленностью или без).

Примером функции, для которой доказан экспоненциальный разрыв между квантовой и классической моделями одновременной передачи сообщений, является функция равенства:

$$\mathrm{EQ}(x, y) = \begin{cases} 1, & \text{если } x = y, \\ 0, & \text{иначе.} \end{cases}$$

Коммуникационная сложность проверки равенства в основной модели задается следующим утверждением.

Утверждение 4. $D(\mathrm{EQ}) = n; \quad R_\varepsilon(\mathrm{EQ}) = O(\log n)$.

В модели SMP ситуация изменяется.

Утверждение 5. $R_\varepsilon^{SMP}(\mathrm{EQ}) = \Omega(\sqrt{n}); \quad Q_\varepsilon^{SMP}(\mathrm{EQ}) = O(\log n)$.

Нижняя оценка на вероятностную сложность проверки равенства в модели SMP вытекает из следующей теоремы.

Теорема 2 (Babai, Kimmel [28], 1996). *В SMP модели разрыв между вероятностной и детерминированной коммуникационной сложностью не более чем квадратичный для любой функции.*

Для проверки равенства слов полезными оказываются хорошие коды.

Определение. Семейство подмножеств $C_n \subset \{0,1\}^n$ называется (асимптотически) *хорошим кодом*, если найдутся такие две константы δ, r, что

- $|C_n| \geqslant 2^{rn}$;
- $\|x \oplus y\| \geqslant \delta n$ при $x, y \in C_n$, $x \neq y$.

r назовем *пропускной способностью*,
δ — (относительным) *кодовым расстоянием*.

Задача 7. Докажите, что хорошие коды существуют при $r + H_2(\delta) < 1$, $\delta < 1/2$.

Здесь $H_2 = -\delta \log_2 \delta - (1-\delta)\log_2 \delta$ — функция двоичной энтропии.

Указание: достаточно использовать жадный метод построения кода — добавляйте к уже построенным кодовым словам новое слово, которое далеко от всех них.

Задача 8. Докажите, используя хорошие коды, что $R_\varepsilon(\mathrm{EQ}) = O(\log n)$.

Задача 9. Докажите, что $R_\varepsilon^{\mathrm{SMP,\ pub}}(\mathrm{EQ}) = O(\log n)$ (хорошие коды и здесь полезны).

В квантовом случае не нужна предварительная сцепленность. Мы сейчас опишем протокол, который позволяет проверять равенство при логарифмическом объеме сообщений.[3)]

Алиса, Боб и Чарли заранее (перед началом работы) выбирают хорошее семейство кодов C_m с пропускной способностью r и кодовым расстоянием δ. (Для определенности $\delta = 1/4$, $r = 1/8$.) Кроме того, они фиксируют некоторую кодирующую функцию (инъективное отображение)

$$c_n \colon \{0,1\}^n \to C_{n/r}.$$

Теперь собственно протокол. Алиса готовит свое сообщение следующим образом: по x находит $u^{(x)} = c_n(x)$, где $n = |x|$, после чего порождает квантовое состояние

$$|\psi_x\rangle = \frac{1}{\sqrt{m}} \sum_{k=1}^{m} |k, u_k^{(x)}\rangle, \quad m = n/r, \quad u_k^{(x)} \text{ — } k\text{-й бит } u^{(x)}.$$

Здесь k представляется двоичной записью, поэтому в этом состоянии используется $O(\log m)$ кубитов.

Боб готовит аналогичное сообщение $|\psi_y\rangle$, используя имеющееся у него слово y.

Чарли, получив от Алисы и Боба $|\psi_x\rangle$ и $|\psi_y\rangle$, выполняет с ними действия, изображенные схемой на рисунке 2.

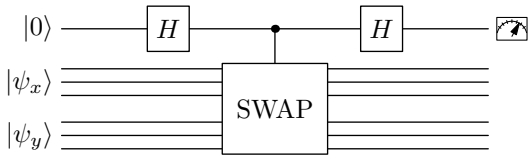
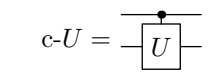


Рис. 2. Действия Чарли

На этом рисунке через H обозначен уже известный нам оператор Адамара

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

В средней части схемы появляется новое действующее лицо: *условный оператор*. Общее определение условного оператора U, контролируемого кубитом 1 (для простоты записи), таково:



$$\text{c-}U \colon |0\rangle \otimes |\psi\rangle \to |0\rangle \otimes |\psi\rangle$$
$$\text{c-}U \colon |1\rangle \otimes |\psi\rangle \to |1\rangle \otimes U|\psi\rangle$$

[3)] Этот протокол предложен в работе [35], 2001.

Из этого определения название становится понятным: оператор U применяется при условии, что первый кубит находится в состоянии $|1\rangle$, и не применяется, если первый кубит находится в состоянии $|0\rangle$.

Упражнение 10. Проверьте, что для любого унитарного оператора U оператор c-U также унитарен.

Последнее, что нужно объяснить в схеме действий Чарли, это оператор SWAP. Он меняет местами два квантовых регистра (группы кубитов, в данном случае — кубиты, присланные Алисой, и кубиты, присланные Бобом)

$$\mathrm{SWAP}\colon |u\rangle \otimes |v\rangle \mapsto |v\rangle \otimes |u\rangle$$

(на остальные векторы определение продолжается по линейности).

Замечание 1. Действия Чарли выглядят на первый взгляд странно. На самом деле, схема на рисунке 2 является частным случаем процедуры измерения собственного числа оператора. Мы еще вернемся к этой процедуре во второй части курса.

Теперь проанализируем описанный выше протокол. Во-первых, из свойств хорошего кода получаем такую оценку скалярного произведения сообщений Алисы и Боба при $x \neq y$:

$$\langle \psi_x | \psi_y \rangle = \frac{1}{m} \#(k : u_k^{(x)} = u_k^{(y)}) \leqslant 1 - \delta.$$

При $x = y$ сообщения Алисы и Боба совпадают и потому $\langle \psi_x | \psi_x \rangle = 1$.

Схема действий Чарли с заметной долей вероятности обнаруживает разницу в случае $x \neq y$. Вычисления:

$$\begin{aligned}
|0\rangle \otimes |\psi_x\rangle \otimes |\psi_y\rangle &\overset{H}{\longmapsto} \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) \otimes |\psi_x\rangle \otimes |\psi_y\rangle \\
&\xrightarrow{\text{c-SWAP}} \frac{1}{\sqrt{2}}|0\rangle \otimes |\psi_x\rangle \otimes |\psi_y\rangle + \frac{1}{\sqrt{2}}|1\rangle \otimes |\psi_y\rangle \otimes |\psi_x\rangle \\
&\overset{H}{\longmapsto} \frac{1}{2}\big(|0\rangle + |1\rangle\big) \otimes |\psi_x\rangle \otimes |\psi_y\rangle + \frac{1}{2}\big(|0\rangle - |1\rangle\big) \otimes |\psi_y\rangle \otimes |\psi_x\rangle = \\
&= \frac{1}{2}|0\rangle \otimes \big(|\psi_x\rangle \otimes |\psi_y\rangle + |\psi_y\rangle \otimes |\psi_x\rangle\big) + \frac{1}{2}|1\rangle \otimes \big(|\psi_x\rangle \otimes |\psi_y\rangle - |\psi_y\rangle \otimes |\psi_x\rangle\big)
\end{aligned}$$

Вероятность исхода 1 при измерении первого (управляющего условным обменом) кубита равна

$$\begin{aligned}
\mathbf{Pr}(1) &= \frac{1}{4}\big(\langle\psi_x| \otimes \langle\psi_y| - \langle\psi_y| \otimes \langle\psi_x|\big)\big(|\psi_x\rangle \otimes |\psi_y\rangle - |\psi_y\rangle \otimes |\psi_x\rangle\big) = \\
&= \frac{1}{4}\big(2 - \langle\psi_x|\psi_y\rangle\langle\psi_y|\psi_x\rangle - \langle\psi_y|\psi_x\rangle\langle\psi_x|\psi_y\rangle\big) = \frac{1}{2} - \frac{1}{2}|\langle\psi_x|\psi_y\rangle|^2.
\end{aligned}$$

Итак, в случае $x = y$ вероятность исхода 1 равна 0, а в случае $x \neq y$ она не меньше

$$\frac{1}{2} - \frac{1}{2}(1-\delta)^2 = \delta - \delta^2/2 > 1/5$$

(при выбранных выше значениях δ и r).

Чтобы достичь сколь угодно малой ошибки, Алиса и Боб должны приготовить несколько сообщений такого вида, а Чарли должен их обработать независимо и сказать «$x \neq y$», если хотя бы один из наблюдаемых исходов равен 1. Оценка вероятности ошибки аналогична уже проводившимся ранее.

5.4. Квантовая (псевдо-)телепатия

В заключение рассмотрим эффект предварительной сцепленности. Для физиков эффекты, возникающие из-за сцепленности играют большую роль, поскольку дают возможность провести эксперименты, проверяющие квантовую механику. В квантовой информатике сцепленность играет роль независимого ресурса при передаче сообщений.

Мы не обсуждали (и не будем обсуждать) квантовую теорию информации. Вместо этого рассмотрим пример экзотической коммуникации, когда задача, неразрешимая классическими средствами, решается использованием предварительной сцепленности.[4)]

Ситуацию, описанную ниже, можно принять за телепатию: участники протокола дают ответ на вопрос, на который они, по классическим представлениям, ответить не могут. С классической точки зрения такая согласованность может обеспечиваться передачей некоторой информации между участниками. Однако в квантовом мире вместо передачи информации можно использовать квантовую сцепленность.

Итак, опишем *игру Мермина*.

Условия: Алиса, Боб и Чарли рассаживаются по хорошо изолированным камерам. Каждый из них получает один бит: x_a, x_b, x_c. Им заранее известно, что либо ровно два значения из выданных битов равны 1, либо все биты равны 0. Другими словами,

$$x_a + x_b + x_c \equiv 0 \pmod 2.$$

Каждый из игроков должен сообщить один бит: y_a, y_b, y_c.

Цель: Игроки выигрывают, если

$$y_a + y_b + y_c \equiv (x_a + x_b + x_c)/2 \pmod 2.$$

Другими словами, если среди полученных битов есть единичные, то игроки должны сообщить нечетное число единиц (порядок неважен); а если все полученные биты равны 0, то четное количество.

Возможности классических протоколов исчерпываются утверждениями следующей задачи.

Задача 11.

(i) Докажите, что не существует вероятностной стратегии в игре Мермина, которая гарантировала бы (с вероятностью 1) победу игроков, даже если они используют общий генератор случайности.

(ii) Постройте стратегию с общим генератором случайности, при которой вероятность выигрыша 3/4.

Однако при использовании предварительной сцепленности, участники могут гарантировать выигрыш.

Опишем действия игроков.

Пока их не рассадили по камерам, игроки создают состояние GHZ

$$\frac{1}{\sqrt{2}}\big(|000\rangle + |111\rangle\big)\ .$$

Каждый из них забирает в камеру один из битов этого состояния.

Получив бит x, игрок применяет условный фазовый сдвиг к своему кубиту:

$$S\colon |0\rangle \mapsto |0\rangle, \quad S\colon |1\rangle \mapsto i^x|1\rangle,$$

[4)] Этот пример взят из работы [33], 2004.

затем применяет преобразование Адамара H, после чего производит измерение своего кубита и сообщает наблюдаемый исход.

Проанализируем этот протокол. Перед условным фазовом сдвигом состояние трех кубитов

$$\frac{1}{\sqrt{2}}\big(|000\rangle + |111\rangle\big).$$

После условного фазового сдвига оно изменяется на

$$\frac{1}{\sqrt{2}}\big(|000\rangle + i^{x_a+x_b+x_c}|111\rangle\big) =$$

$$= \begin{cases} \dfrac{1}{\sqrt{2}}\big(|000\rangle + |111\rangle\big), & \text{если } x_a + x_b + x_c = 0 \text{ (случай \textbf{(ч)})}; \\ \dfrac{1}{\sqrt{2}}\big(|000\rangle - |111\rangle\big), & \text{если } x_a + x_b + x_c = 2 \text{ (случай \textbf{(н)})}. \end{cases}$$

Заметим, что полученные состояния ортогональны и отвечают двум возможным случаях. Поэтому уже неудивительная возможность безошибочного ответа.

При решении последней задачи полезно вспомнить формулу для тензорной степени преобразования Адамара, которая использовалась при анализе решения задачи Дойча – Джоза.

Задача 12. Проверьте, что после применения преобразования Адамара к каждому из кубитов в случае **(ч)** получается сумма базисных состояний с четным числом единиц, а в случае **(н)** — с нечетным числом единиц.

Замечание 2. Известно, что информация не может передаваться быстрее скорости света. Участники игры Мермина могут быть сколь угодно далеко разнесены в пространстве (нужно только предохранять квантовую сцепленность от помех).

Возникает кажущийся парадокс: участники согласуют свои действия, не передавая при этом никакой информации. Предлагаю слушателям разобраться в этом парадоксе самостоятельно.

Лекция 6. Квантовые схемы

Пока мы рассматривали возможности квантовых алгоритмов, не затрагивая вопроса об их эффективности (сколько ресурсов необходимо для исполнения алгоритма). Здесь мы обсудим один из самых важных ресурсов — время вычисления. Оказывается, что делать это удобнее на языке схем.

6.1. Трудоемкость квантового вычисления

В лекции 1 была изображена общая схема использования квантового ресурса в алгоритмах (рис. 1.3). Напомним, что после применения квантового преобразования к квантовой памяти производится измерение, результаты которого используются в дальнейшей работе. Таким образом, применение квантового преобразования с точки зрения алгоритма, использующего квантовый ресурс, порождает вероятностное распределение на исходах измерения. Поэтому сразу же возникают два вопроса:

1. Как определить ресурсы (например, время) для порождения распределения p квантовым устройством?

2. Насколько сложно породить близкое распределение классическими средствами?

Начнем с первого вопроса. Мы будем представлять квантовое преобразование (точнее, соответствующий ему унитарный оператор) как композицию некоторого количества *элементарных действий* — тех преобразований, которые возможно выполнить за один такт работы устройства.

Набор унитарных операторов, отвечающих элементарным действиям, будем называть *базисом*.

Есть два естественных физических ограничения на элементарные действия:

1. Элементарное действие *локально*, т. е. оно нетривиально действует лишь на небольшое количество кубитов (один, два, три, ..., $O(1)$).

2. Если два элементарных действия совершаются *одновременно*, то они нетривиально действуют на *разных* наборах кубитов.

Чтобы обсуждать эти ограничения, введем формальное понятие квантовой схемы.

Определение. *Квантовая схема над базисом* $\mathcal{B}$ — это последовательность операторов

$$U_1[S_1], U_2[S_2], \ldots, U_\ell[S_\ell], \tag{1}$$

где $U_k \in \mathcal{B}$, $S_k \subseteq \{1, \ldots, n\}$, n — общее количество используемых кубитов.

Квантовые схемы обычно изображаются графически так, как показано на рис. 1.

В (1) использовано обозначение $U[S]$ для оператора, который действует нетривиально на кубитах из множества S. Как объяснялось раньше в лекции 1, такой оператор является тензорным произведением оператора U и единичного оператора I. Более формально, матричные элементы оператора $U[S]$ имеют следующий вид.

Пусть

$$U = \sum_{x,y \in \{0,1\}^d} u_{x,y} |x\rangle\langle y|, \quad S = \{j_1, \ldots, j_d\}, 1 \leqslant j_1 < j_2 < \cdots < j_d \leqslant n.$$

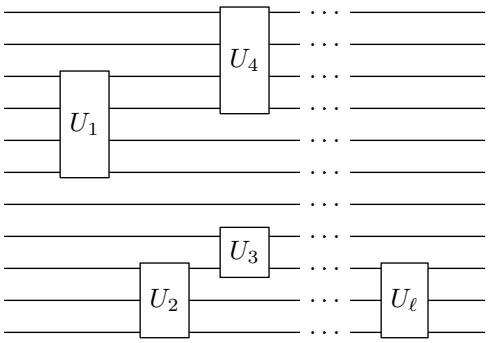


Рис. 1. Изображение квантовой схемы

Обозначим $x[S]$ подпоследовательность битов, стоящих на местах из множества S. Тогда оператор $U[S]$ записывается как

$$U[S] = \sum_{x,y\in\{0,1\}^n : x[\bar{S}]=y[\bar{S}]} u_{x[S],y[S]} |x\rangle\langle y|.$$

Две основные меры сложности схемы — *размер* и *глубина*.

Размер схемы — это количество элементов ℓ. Неформально размер схемы описывает время вычисления на последовательном устройстве, которое выполняет элементарные действия одно за другим.

Формальное определение глубины схемы более замысловатое. По определению, глубиной схемы называется *наименьшее количество слоев*, в которые можно расположить элементы схемы при соблюдении условий:

1. элементы, которые стоят в схеме после j-го элемента, не попадают в слои, предшествующие слою, в котором находится j-й элемент;

2. элементы из одного слоя действуют на непересекающиеся множества кубитов.

Таким образом, глубина отвечает за время вычисления на устройстве, которое способно выполнять одновременно несколько разных действий (*параллельное вычисление*). Второе условие отвечает указанному выше физическому ограничению — в один момент времени кубит может быть вовлечен лишь в одно элементарное действие.

Далее мы ограничимся лишь оценками размера схем.

Используя понятие квантовой схемы, уточним порядок применения квантового ресурса как показано на рис. 2. Входные кубиты приготавливаются в состоянии $|0\rangle$, затем применяется схема, после чего часть кубитов измеряется.

Такой порядок действий порождает несколько вопросов:

1. Почему можно использовать начальное состояние $|0^n\rangle$?

2. Какие начальные состояния помимо $|0^n\rangle$ можно использовать?

3. Как зависит трудоемкость от выбора базиса?

Проще всего ответить на вопрос 1. Имеет место следующий факт: *если есть прибор, измеряющий в вычислительном базисе, то можно приготавливать состояния из вычислительного базиса*.

Действительно, приготовить состояние $|0\rangle$ можно, руководствуясь следующим рецептом:

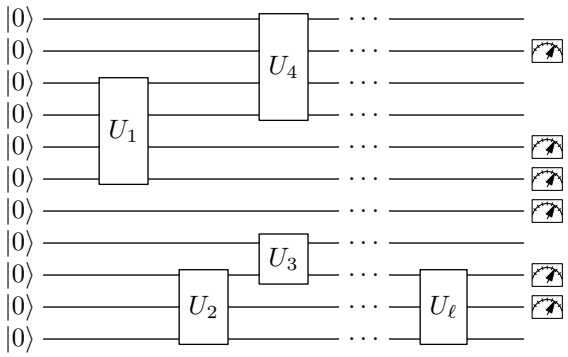


Рис. 2. Использование квантового ресурса

1. Берем случайное состояние.
2. Измеряем его в вычислительном базисе.
3. Если наблюдаем исход 0, то кубит находится в состоянии $|0\rangle$: готово.
4. В противном случае повторяем процедуру.

Здесь нужно объяснить два момента. Во-первых, откуда берется случайное состояние. С физической точки зрения ответ на этот вопрос прост: достаточно «горячая» система находится в случайном состоянии.

Во-вторых, описанная схема действий потенциально бесконечна. Ведь возможен такой сценарий, при котором результаты измерений постоянно дают $|1\rangle$. Однако здесь вступают в силу обычные свойства вероятности: вероятность многократного повторения единиц экспоненциально убывает, так что вероятность того, что «печка» будет долго готовить состояние $|0\rangle$, пренебрежимо мала.

С вопросом номер 2 дело обстоит сложнее. Ясно, что начальное состояние нужно уметь приготавливать достаточно быстро. Годятся любые состояния из вычислительного базиса и (более общо) те, которые получаются из них действием достаточно коротких схем. Но использовать такие состояния неинтересно: фактически часть схемы прячется в схему приготовления состояния.

Возможно, однако, что какие-то физические процессы способны создавать сложные состояния за небольшое время. Таким образом, ответ на этот вопрос зависит от законов физики. По-видимому, ничего принципиально отличного от состояний из вычислительного базиса приготовить нельзя, основываясь на известных законах природы.

Остался последний вопрос 3 о роли выбора базиса для квантовых схем. Этим вопросом мы и займемся на довольно длительное время. Сразу анонсирую основные пункты ответа на этот вопрос:

- Базис из операторов, действующих на одном кубите, неинтересен (моделируется классически).
- Базисы из операторов, действующих на k кубитах, эффективно эквивалентны для всех k.
- Более того, существует *конечный базис* из операторов, действующих не более чем на двух кубитах, который им всем эффективно эквивалентен (при разумных предположениях).

6.2. Точная реализация унитарных операторов квантовыми схемами

Мы начнем с того, что рассмотрим более сильную задачу, чем реализация некоторого вероятностного распределения на исходах наблюдения. А именно, займемся таким вопросом: какие унитарные операторы можно реализовать в некотором базисе?

Определение. Схема $U_1[S_1], U_2[S_2], \ldots, U_\ell[S_\ell]$ реализует оператор

$$U = U_\ell[S_\ell] \ldots U_2[S_2] U_1[S_1]$$

(обратите внимание на порядок).

Схема $U_1[S_1], U_2[S_2], \ldots, U_\ell[S_\ell]$ реализует оператор U *в расширенном смысле*, если

$$U_\ell[S_\ell] \ldots U_2[S_2] U_1[S_1] \colon |\psi\rangle \otimes |0^N\rangle \mapsto U|\psi\rangle \otimes |0^N\rangle$$

для всех $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$.

Сложностью реализации оператора U (в расширенном смысле) в базисе $\mathcal{B}$ называется наименьший размер схемы в базисе $\mathcal{B}$, реализующей U (в расширенном смысле).

Сложность бесконечна, если реализации не существует.

Реализация в расширенном смысле предполагает очень ограничительное использование вспомогательных кубитов: в начале и конце вычисления состояние вспомогательных кубитов должно быть одно и то же. Такое ограничение связано с тем, что нас, в сущности, интересует распределение результатов измерения в вычислительном базисе состояния $U(|0\rangle \otimes |\text{ancilla}\rangle)$. Если состояние $|\text{ancilla}\rangle$ одно и то же в начале и конце вычисления, то распределение исходов измерения первого регистра такой расширенной системы будет совпадать с распределением исходов при измерении системы в состоянии $U|0\rangle$.

Достаточны и более слабые условия, например,

$$U_\ell[S_\ell] \ldots U_2[S_2] U_1[S_1] \colon |\psi\rangle \otimes |0^N\rangle \mapsto U|\psi\rangle \otimes V|\psi\rangle. \tag{2}$$

Однако, это условие не слишком добавляет общности. Из линейности и унитарности следует, что состояние второго регистра обязано быть одинаковым для всех $|\psi\rangle$.

Задача 1. Докажите, что из выполнения (2) следует $V|\psi\rangle = |\xi\rangle$ для любого $|\psi\rangle$.

Дополнительным преимуществом условия реализации в расширенном смысле является сохранение этого свойства при композициях.

6.2.1. Обратимые вычисления: мостик между классическими и квантовыми

Начнем наш анализ с важного частного случая.

Определение. Унитарный оператор назовем *перестановочным*, если он сохраняет множество базисных векторов.

Раз перестановочный оператор сохраняет вычислительный базис, он действует классически. Если ограничить базис только перестановочными операторами, то предыдущие определения дают понятие *обратимого вычисления*, схем для обратимого вычисления и реализации (в расширенном смысле) перестановок обратимыми схемами.

Мы уже обсуждали в лекции 2 соотношение между обратимыми (на микроуровне) действиями и необратимыми на макроуровне операциями. Сейчас мы повторим их для обратимых схем.

Прежде всего заметим, что если разрешить вместо перестановок любые отображения, получим обычные классические схемы. В этом случае обычно считают без ограничения общности, что элементы схем вычисляют булевы функции (область значений $\{0,1\}$).

Хорошо известно, что базис из отображений на двух битах полный, т. е. в нем реализуются любые отображения. Достаточно даже использовать в качестве базисных функций конъюнкцию и отрицание. (На самом деле, существует даже полный базис из одной булевой функции, но нам он не понадобится.)

Теперь опишем моделирование необратимого вычисления обратимым. По классическому базису $\mathcal{B} = \{f_1, \ldots, f_m\}$ из функций $f_k\colon \{0,1\}^{d_k} \to \{0,1\}$ построим обратимый базис $\mathcal{B}_\oplus = \{f_k^\oplus, \text{c-NOT}\}$, где

$$f_k^\oplus\colon (x,y) \mapsto (x, y \oplus f_k(x)), \quad \text{c-NOT}\colon (x,y) \mapsto (x, x \oplus y).$$

Теорема 1 (реализация в расширенном смысле). *Если отображение*

$$F\colon \mathbb{B}^n \to \mathbb{B}^m$$

реализуется булевой схемой размера L в базисе $\mathcal{B}$, то существует схема размера $O(L+m)$ в базисе $\mathcal{B}_\oplus$, которая реализует отображение

$$F^\oplus\colon (x, y, 0^L) \mapsto (x, y \oplus F(x), 0^L).$$

Доказательство. Опишем последовательность действий по построению указанной в теореме обратимой схемы.

Представим схему, реализующую в базисе $\mathcal{B}$ отображение F, как последовательность присваиваний

$$z_j := f_{k_j}(\text{предыдущие значения и входные переменные}).$$

Сопоставим битам из третьего регистра вспомогательные переменные схемы z_j (результаты присваиваний).

Заменим каждое присваивание на применение соответствующего $f_{k_j}^\oplus$ к соответствующим битам.

Далее скопируем, используя перестановочный оператор c-NOT, биты ответа во второй регистр.

А теперь откатим все шаги, кроме последнего (т. е. повторим их в порядке, обратном первоначальному — в программах такая опция называется ‘undo’).

Почему откатка вернет нули в третьем регистре? Дело в том, что перестановка $f^\oplus$ инволютивна:

$$(x,y) \overset{f^\oplus}{\longmapsto} (x, y \oplus f(x)) \overset{f^\oplus}{\longmapsto} (x, y \oplus f(x) \oplus f(x)) = (x,y).$$

Итак, во втором регистре мы получили $y \oplus F(x)$, а в третьем — нули. Искомая схема построена. Дополнительные m действий возникают из-за необходимости скопировать результат вычислений необратимой схемой перед откаткой. □

Поскольку конъюнкция и отрицание образуют полный базис для необратимых схем, получаем такое следствие для обратимых схем:

Следствие (полнота NOT-базиса). *Любое отображение $F\colon \mathbb{B}^n \to \mathbb{B}^m$ реализуется в расширенном смысле (с использованием вспомогательных битов, не меняющих своего значения после вычисления) в базисе*

$$\sigma_x = \text{NOT}\colon x \mapsto 1 \oplus x; \quad \text{c-NOT}\colon (x,y) \mapsto (x, x \oplus y);$$
$$\text{cc-NOT}\colon (x,y,z) \mapsto (x, y, z \oplus xy) \text{ (элемент Тоффоли).}$$

Полученный базис содержит элемент Тоффоли, который действует на трех битах. Оказывается, в случае перестановочных операторов это неизбежно.

Задача 2. Докажите, что в базисе из перестановок двух битов не все отображения реализуемы в расширенном смысле.

Использование вспомогательных битов (реализация в расширенном смысле) также неизбежно, как показывает следующая задача.

Задача 3. Докажите, что без использования вспомогательных битов невозможно реализовать отображение

$$\mathrm{c}^{(n)}\text{-NOT}\colon (x_1, \ldots, x_n, y) \mapsto (x_1, \ldots, x_n, y \oplus x_1 x_2 \ldots x_n)$$

в базисе из перестановок n битов.

Эта задача показывает, что вспомогательные биты оказываются необходимыми как только базис не содержит перестановки, действующие на все биты сразу.

6.2.2. Базис из операторов, действующих на одном кубите

Теперь перейдем к произвольным унитарным операторам и рассмотрим для начала случай базиса $\mathcal{B}_1$, состоящего из операторов, действующих на одном кубите. В этом случае действия операторов в схеме коммутируют

$$(U_1 \otimes I)(I \otimes U_2) = U_1 \otimes U_2 = (I \otimes U_2)(U_1 \otimes I),$$

так что реализуются лишь операторы вида

$$U_1 \otimes U_2 \otimes \cdots \otimes U_n.$$

Поэтому исходы при наблюдении распределены независимо для каждого кубита. Такое распределение моделируется классически: для каждого кубита нужно использовать подходящим образом «подпорченную монету», которая дает нужное распределение нуля и единицы.

Здесь важно указать на следующее обстоятельство, общее для вероятностных и квантовых вычислений. Использование произвольных распределений на $\{0, 1\}$ делает некоторые невычислимые функции вычислимыми. Проверьте, что для этого достаточно, чтобы биты в двоичной записи вероятности единицы $p(1)$ образовывали невычислимую последовательность.

В случае вероятностных вычислений стандартный выход из этого затруднения состоит в использовании равномерного распределения на некотором числе кубитов. Тогда уже не все распределения удается смоделировать.

В квантовом случае стандартным выходом является переход к конечному базису и приближенной реализации унитарных операторов. Мы обсудим его позже.

6.2.3. Базис из операторов, действующих на двух кубитах

Теорема 2 (универсальность двухкубитовых операторов). *Любой унитарный оператор точно реализуется в расширенном смысле в базисе $\mathcal{B}_2$, состоящем из всех операторов, действующих на двух кубитах.*

Доказательство теоремы довольно длинное, но распадается на несколько шагов. Мы докажем по очереди следующие утверждения.

1. Любой унитарный оператор — композиция подкрученных транспозиций.

2. Любая подкрученная транспозиция реализуется в базисе, который содержит все операторы, действующие на одном кубите, и NOT-базис (NOT, c-NOT, cc-NOT).

3. Элемент Тоффоли cc-NOT реализуется в базисе $\mathcal{B}_2$.

Начнем с того, что введем подкрученные транспозиции. Напомним, что транспозицией называется перестановка, которая меняет местами два элемента и оставляет на месте остальные. Хорошо известно, что любая перестановка является композицией транспозиций.

В случае унитарных операторов справедливо аналогичное утверждение для подходящего обобщения понятия транспозиции.

Определение. *Подкрученной транспозицией* называется унитарный оператор $U\colon \mathbb{C}^M \to \mathbb{C}^M$, матрица которого имеет вид:

$$\begin{pmatrix} 1 & 0 & \dots & & & & & 0 \\ \vdots & \ddots & 0 & \dots & & & & 0 \\ 0 & \dots & a & 0 & \dots & 0 & b & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 0 & \dots & & 0 \\ \dots & & & & \ddots & & & & \dots \\ 0 & \dots & 0 & 0 & \dots & 1 & \dots & & 0 \\ 0 & \dots & c & 0 & \dots & d & \dots & & 0 \\ \dots & & & & & & \ddots & & 0 \\ 0 & \dots & & & & & & & 1 \end{pmatrix}.$$

При $a = d = 0$ и $b = c = 1$ получаем обычную транспозицию.

Лемма 1. *Любой унитарный оператор $U\colon \mathbb{C}^M \to \mathbb{C}^M$, где $M \geqslant 2$, является композицией подкрученных транспозиций.*

Доказательство. Доказываем индукцией по порядку матрицы аналогично тому, как доказывается, что транспозиции порождают все перестановки.

База индукции очевидна: при $M = 2$ все операторы являются подкрученными транспозициями по определению.

Для индуктивного перехода будем использовать следующее наблюдение:

Для любых c_1, c_2 существует такая унитарная матрица V размера 2×2, что

$$V \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} \sqrt{|c_1|^2 + |c_2|^2} \\ 0 \end{pmatrix}.$$

Упражнение 4. Докажите справедливость этого наблюдения.

Основываясь на сделанном наблюдении, заключаем, что для любого $|\xi\rangle \in \mathbb{C}^M$ существует последовательность $V^{(1)}, \dots, V^{(M-1)}$ такая, что

$$V^{(1)} \cdot \dots \cdot V^{(M-1)} |\xi\rangle = |1\rangle,$$

где $V^{(s)}$ — подкрученная транспозиция, которая действует на подпространстве $\mathbb{C}(|s\rangle, |s+1\rangle)$. (После применения $V^{(s)}$ все координаты от $(s+1)$-й до M-й равны 0.)

Поэтому умножениями на подкрученные транспозиции можно перевести первый столбец любой унитарной матрицы в единичный. Из условия унитарности следует, что после этого и первая строка станет единичной:

$$V^{(1)} \cdot \dots \cdot V^{(M-1)} U = \begin{pmatrix} 1 & 0^{M-1} \\ 0^{M-1} & U_1 \end{pmatrix}$$

Осталось применить утверждение леммы к U_1 (для которого оно справедливо в силу предположения индукции). $\square$

Теперь перейдем ко второму шагу в доказательстве теоремы 2. Рассмотрим операторы, управляемый n кубитами:

$$\mathrm{c}^{(n)}\text{-}U\colon |x_1,\dots,x_n\rangle\otimes|\psi\rangle = \begin{cases} |x_1,\dots,x_n\rangle\otimes U|\psi\rangle, & \text{если } x_1x_2\dots x_n=1;\\ |x_1,\dots,x_n\rangle\otimes|\psi\rangle, & \text{иначе.}\end{cases}$$

Они реализуются в базисе $\mathcal{B}_2\cup\{\mathrm{NOT},\mathrm{c\text{-}NOT},\mathrm{cc\text{-}NOT}\}$ схемами следующего вида Существование схемы в базисе $\{\mathrm{NOT},\mathrm{c\text{-}NOT},\mathrm{cc\text{-}NOT}\}$, вычисляющей конъ-

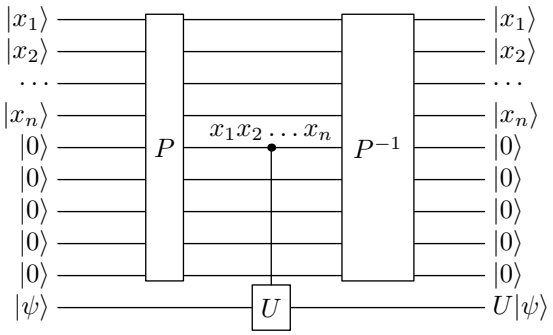


Рис. 3. Схема P вычисляет конъюнкцию $x_1x_2\dots x_n$

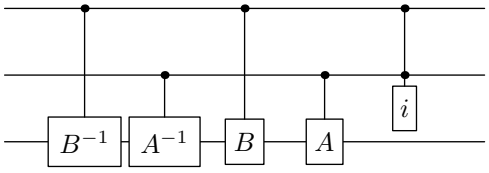
юнкцию, следует из замечаний, сделанных после доказательства теоремы 1. В середине схемы помещен управляемый оператор U, управляющий кубит которого совпадает с кубитом результата схемы P.

Заметим, что операторы $\mathrm{c}^{(n)}$-U являются подкрученными транспозициями, действующими на пространстве $\mathbb{C}(|1\dots10\rangle,|1\dots11\rangle)$. Любая другая подкрученная транспозиция T получается из $\mathrm{c}^{(n)}$-U сопряжением некоторым перестановочным оператором:

$$T = P\,\mathrm{c}^{(n)}\text{-}UP^{-1},\quad P\colon|1\dots10\rangle\mapsto|x\rangle,\quad P\colon|1\dots11\rangle\mapsto|y\rangle.$$

Такой оператор P реализуется в NOT-базисе в расширенном смысле в силу теоремы 1.

Итак, любая подкрученная транспозиция реализуется в базисе $\mathcal{B}_2$, к которому добавлен элемент Тоффоли. Последний шаг доказательства теоремы 2 состоит в реализации элемента Тоффоли в базисе $\mathcal{B}_2$. Такая реализация представлена на рис. 4.



$$A=\frac{1}{\sqrt{2}}\begin{pmatrix}-i & -1\\ 1 & i\end{pmatrix};\quad B=\begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}$$

Рис. 4.

Последний оператор в этой схеме умножает на i базисный вектор $|11\rangle$, а на остальных векторах вычислительного базиса действует тождественно.

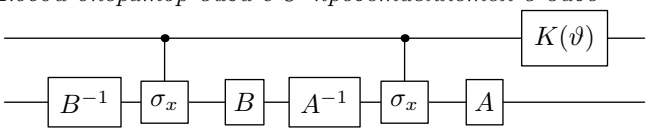
Проверка того, что указанная схема и впрямь реализует элемент Тоффоли оставляется в качестве упражнения.

Упражнение 5. Проверьте, что $A^2 = -I, \quad B^2 = -I, \quad ABA^{-1}B^{-1} = -i\sigma_x$.

На этом доказательство теоремы 2 закончено. Заметим, что из доказательства видно, что любой унитарный оператор реализуется в расширенном смысле в базисе, который содержит все однокубитовые операторы и все операторы вида с-U, где U — однокубитовый.

Оказывается, из второй группы операторов достаточно оставить только оператор обратимого копирования кубита с-NOT = с-σ_x.

Теорема 3. *Любой оператор вида* с-U *представляется в виде*



$$\text{где } K(\vartheta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\vartheta} \end{pmatrix}$$

Для доказательства этой теоремы нужно более подробно изучить унитарные операторы в двумерном пространстве.

6.3. Об унитарных преобразованиях одного кубита

Непосредственно проверяется, что следующие матрицы (*матрицы Паули*) являются одновременно и унитарными, и эрмитовыми

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Матрицы Паули имеют прямое отношение к упоминавшейся в лекции 1 сфере Блоха.

Утверждение 1. *Если* $|\psi\rangle = a|0\rangle + b|1\rangle$ *— состояние кубита, то*

$$|\psi\rangle\langle\psi| = \frac{1}{2}(I + x\sigma_x + y\sigma_y + z\sigma_z), \quad x^2 + y^2 + z^2 = 1,\ x, y, z \in \mathbb{R}. \tag{3}$$

Доказательство. Поскольку общий фазовый множитель не меняет $|\psi\rangle\langle\psi|$, без ограничения общности можно считать, что $a \in [0; 1]$.

Тогда полагаем $a = \cos(\theta/2)$, $b = e^{i\varphi}\sin(\theta/2)$, $\theta \in [0; \pi]$, $\varphi \in [0; 2\pi)$,

$$x = \cos\varphi\sin\theta; \quad y = \sin\varphi\sin\theta; \quad z = \cos\theta$$

(сферические координаты).

Равенство (3) проверяется прямым вычислением

$$|\psi\rangle\langle\psi| = \begin{pmatrix} \cos(\theta/2) \\ e^{i\varphi}\sin(\theta/2) \end{pmatrix} \begin{pmatrix} \cos(\theta/2) & e^{-i\varphi}\sin(\theta/2) \end{pmatrix} =$$

$$= \begin{pmatrix} \cos^2(\theta/2) & e^{-i\varphi}\cos(\theta/2)\sin(\theta/2) \\ e^{i\varphi}\cos(\theta/2)\sin(\theta/2) & \sin^2(\theta/2) \end{pmatrix} =$$

$$= \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{2}\cos\theta\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + \frac{1}{2}\cos\varphi\sin\theta\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + \frac{1}{2}\sin\varphi\sin\theta\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} =$$

$$= \frac{1}{2}(I + x\sigma_x + y\sigma_y + z\sigma_z)$$

□

В лекции 1 упоминалось, что помимо эрмитова произведения на сфере Блоха есть еще одно скалярное произведение — обычное произведение векторов в трехмерном пространстве. Пусть $|\psi\rangle$ на сфере Блоха попадает в (x', y', z'), $|\xi\rangle$ — в (x'', y'', z'').

Матрицы Паули (включая единичную) ортогональны относительно произведения Фробениуса: $\frac{1}{2}\operatorname{Tr}(\sigma_\alpha\sigma_\beta) = \delta_{\alpha\beta}$ (проверьте).

Поэтому

$$|\langle\psi|\xi\rangle|^2 = \operatorname{Tr}(|\psi\rangle\langle\psi|\ |\xi\rangle\langle\xi|) = \frac{1}{2}(1 + x'x'' + y'y'' + z'z''). \tag{4}$$

Из (4) можно заключить несколько следствий. Во-первых, любая пара ортогональных состояний попадает на сфере Блоха в пару диаметрально противоположных точек.

Во-вторых, действие унитарных преобразований на сфере Блоха по правилу

$$U\colon |\psi\rangle\langle\psi| \mapsto U|\psi\rangle\langle\psi|U^\dagger \tag{5}$$

является движением трехмерного пространства, которое сохраняет центр сферы Блоха.

Из формулы (5) для действия унитарного оператора на сфере Блоха следует, что скалярные операторы $|\psi\rangle \mapsto e^{i\alpha}|\psi\rangle$ действуют на сфере Блоха тождественно.

Верно и обратное: если унитарный оператор U действует на сфере Блоха тождественно, то каждый вектор $|\psi\rangle$ в пространстве $\mathbb{C}^2$ — собственный для U, поэтому оператор U — скалярный.

Напомним, что любой унитарный оператор имеет ортонормированный базис из собственных векторов. Соответствующая пара точек на сфере Блоха не меняется при действии этого оператора. Отсюда заключаем, что нетривиальное действие унитарного оператора на сфере Блоха — движение, у которого есть ровно одна пара неподвижных точек на сфере Блоха. Значит, это поворот.

Фактически мы построили изоморфизм между группой $\mathbf{U}(2)/\mathbf{U}(1)$ унитарных матриц второго порядка с точностью до скалярного множителя и группой $\mathbf{SO}(3)$ ортогональных матриц третьего порядка (поворотов).

Для доказательства теоремы 3 из предыдущего раздела полезно изучить, какие повороты реализуются матрицами Паули. Оказывается, это повороты на угол π вокруг координатных осей. Действительно, $\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = I$ и

$$\begin{aligned}
\sigma_x\frac{1}{2}(I + \sigma_x)\sigma_x^\dagger &= \frac{1}{2}(I + \sigma_x),\\
\sigma_y\frac{1}{2}(I + \sigma_y)\sigma_y^\dagger &= \frac{1}{2}(I + \sigma_y),\\
\sigma_z\frac{1}{2}(I + \sigma_z)\sigma_z^\dagger &= \frac{1}{2}(I + \sigma_z).
\end{aligned}$$

Доказательство теоремы 3. Достаточно доказать, что для любого U есть представление вида $U = e^{i\vartheta}A\sigma_xA^{-1}B\sigma_xB^{-1}$.

Но операторы $A\sigma_xA^{-1}$ и $B\sigma_xB^{-1}$ — это произвольные повороты на угол π. Поэтому для завершения доказательства нужен простой факт из элементарной геометрии.

Задача 6. Докажите, что любой поворот трехмерного пространства является композицией двух поворотов на угол π.

□

Лекция 7. Конечные базисы

Если базис содержит лишь конечное количество элементов, точная реализация произвольного оператора становится невозможной просто из-за того, что схем в конечном базисе счетное множество, а операторов — несчетное.

Поэтому нужно ослабить условия реализации и приближать операторы схемами.

7.1. Приближенная реализация унитарных операторов

Определение. Оператор $\tilde{U}$ представляет оператор U с точностью δ, если

$$\|\tilde{U} - U\| < \delta. \tag{1}$$

Здесь используется операторная норма $\|A\| = \max_{x:|x|=1} |Ax|$.

Приведем два важных свойства приближений унитарных операторов.

Утверждение 1. *Если унитарный оператор $\tilde{U}$ приближает U с точностью δ, то $\tilde{U}^{-1}$ приближает U^{-1} с такой же точностью δ.*

Утверждение 2 (линейное накопление ошибок). *Если $\|\tilde{U}_k - U_k\| < \delta_k$, то*

$$\left\|\tilde{U}_L \cdot \ldots \cdot \tilde{U}_1 - U_L \cdot \ldots \cdot U_1\right\| \leq \sum_k \delta_k.$$

Для доказательства этих утверждений нам потребуются следующий свойства операторной нормы:

1^{op}. $\|X\|^2 = \max_{x:|x|=1} \langle x|X^\dagger X|x\rangle$ — наибольшее собственное число оператора $X^\dagger X$ (так как $|Xx|^2 = \langle x|X^\dagger X|x\rangle$).

2^{op}. $\|XY\| \leqslant \|X\|\,\|Y\|$ (так как $|XYx| \leqslant \|X\|\,|Yx| \leqslant \|X\|\,\|Y\|\,|x|$).

3^{op}. $\|X \otimes Y\| = \|X\|\,\|Y\|$ (уже было раньше, следует из 1^{op}).

4^{op}. Для унитарного оператора $\|U\| = 1$ (из определения).

Доказательство утверждения 1. Пусть $\|\tilde{U} - U\| \leqslant \delta$. Тогда

$$\|U^{-1} - \tilde{U}^{-1}\| = \|\tilde{U}^{-1}(\tilde{U} - U)U^{-1}\| \overset{(2^{\text{op}},\,4^{\text{op}})}{\leqslant} \|\tilde{U} - U\| \leqslant \delta.$$

В этой выкладке существенно, что норма унитарного оператора равна 1. В противном случае точность приближения обратного оператора может быть сколь угодно плоха. □

Доказательство утверждения 2. Приведем доказательство для случая двух операторов, общий случай легко получается по индукции. Имеем следующую цепочку неравенств, основанную на применении неравенства 2^{op} и общих свойств нормы:

$$\begin{aligned}\left\|\tilde{U}_2\tilde{U}_1 - U_2U_1\right\| &= \left\|\tilde{U}_2(\tilde{U}_1 - U_1) + (\tilde{U}_2 - U_2)U_1\right\| \leqslant \\ &\leqslant \left\|\tilde{U}_2(\tilde{U}_1 - U_1)\right\| + \left\|(\tilde{U}_2 - U_2)U_1\right\| \leqslant \\ &\leqslant \left\|\tilde{U}_2\right\|\left\|\tilde{U}_1 - U_1\right\| + \left\|\tilde{U}_2 - U_2\right\|\left\|U_1\right\| = \\ &= \left\|\tilde{U}_1 - U_1\right\| + \left\|\tilde{U}_2 - U_2\right\|.\end{aligned}$$

В последнем равенстве используем 4^{op}. В случае операторов общего ошибки накапливаются экспоненциально быстро. □

Как и раньше, мы используем также и реализацию в расширенном смысле (с привлечением дополнительных кубитов).

Определение. Оператор $U\colon (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n}$ *приближается в расширенном смысле* оператором $\tilde{U}\colon (\mathbb{C}^2)^{\otimes N} \to (\mathbb{C}^2)^{\otimes N}$ с точностью δ, если для любого $|\xi\rangle$ из $(\mathbb{C}^2)^{\otimes n}$ выполнено

$$\big|\tilde{U}\left(|\xi\rangle \otimes |0^{N-n}\rangle\right) - U|\xi\rangle \otimes |0^{N-n}\rangle\big| \leq \delta\big|\xi\big|.$$

Для приближений в расширенном смысле выполняются свойства, аналогичные утверждениям 1, 2.

Задача 1. Докажите, что если U_1 приближается в расширенном смысле $\tilde{U}_1$ с точностью δ_1, а U_2 приближается в расширенном смысле $\tilde{U}_2$ с точностью δ_2, то U_1^{-1} приближается в расширенном смысле $\tilde{U}_1^{-1}$ с точностью δ_1, а U_1U_2 приближается в расширенном смысле $\tilde{U}_1\tilde{U}_2$ с точностью $\delta_1 + \delta_2$.

Теперь обсудим, почему указанные выше приближения пригодны для квантовых схем. От квантовой схемы нам нужно только вероятностное распределение на результатах измерения. Поэтому естественнее потребовать близости вероятностных распределений. Оказывается, что близкие операторы порождают близкие распределения.

Напомним, что у нас есть выделенный вычислительный (ортонормированный) базис. Для вектора $|\psi\rangle = \sum_x c_x|x\rangle$ вероятность $\mathbf{Pr}(|\psi\rangle, x)$ исхода x равна $|c_x|^2$. Тогда вероятность произвольного события A равна

$$\mathbf{Pr}(|\psi\rangle, A) = \sum_{x\in A} |c_x|^2 = \big|\Pi_A|\psi\rangle\big|^2, \tag{2}$$

где Π_A — оператор ортогонального проектирования на подпространство, порожденное векторами $|x\rangle$, $x \in A$.

Нас интересует вероятностное распределение, порождаемое измерением вектора $U|0^n\rangle$. Запишем ортогональное разложение

$$U|0^n\rangle = |\psi'\rangle_A + |\psi''\rangle_{\bar{A}},$$

где в первом слагаемом собрана линейная комбинация тех базисных векторов, которые отвечают событию A.

Аналогичное разложение для приближающего оператора

$$\tilde{U}(|0^n\rangle \otimes |0^N\rangle) = U|0^n\rangle \otimes |0^N\rangle + |\Delta\rangle = |\psi'\rangle_A \otimes |0^N\rangle + |\psi''\rangle_{\bar{A}} \otimes |0^N\rangle + |\Delta\rangle,$$

здесь $|\Delta| < \delta$.

Запишем также ортогональное разложение для вектора ошибки:

$$|\Delta\rangle = |\Delta'\rangle_A + |\Delta''\rangle_{\bar{A}}.$$

Теперь можно записать вероятности события A в двух случаях: при действии оператора U и приближающего оператора $\tilde{U}$:

$$\mathbf{Pr}(U|0^n\rangle, A) = |\psi'|^2$$
$$\mathbf{Pr}(\tilde{U}(|0^n\rangle \otimes |0^N\rangle), A) = \big||\psi'\rangle \otimes |0^N\rangle + |\Delta'\rangle\big|^2$$

Поскольку $|\Delta'| < \delta$, имеем

$$|\psi'| - \delta < \big||\psi'\rangle \otimes |0^N\rangle + |\Delta'\rangle\big| < |\psi'| + \delta.$$

Поэтому при $|\psi'| \geqslant \delta$

$$\mathbf{Pr}(U|0^n\rangle, A) - 2\delta|\psi'| + \delta^2 < \mathbf{Pr}(\tilde{U}(|0^n\rangle \otimes |0^N\rangle), A) < \mathbf{Pr}(U|0^n\rangle, A) + 2\delta|\psi'| + \delta^2.$$

Значит,

$$|\,\mathbf{Pr}(U|0^n\rangle, A) - \mathbf{Pr}(\tilde{U}(|0^n\rangle \otimes |0^N\rangle), A)| < 2\delta|\psi'| + \delta^2 \leqslant 2\delta(1 + \delta/2) \leqslant 4\delta.$$

(Здесь использован тот факт, что норма разности унитарных операторов не превосходит 2.)

Упражнение 2. Проверьте, что при $|\psi'| < \delta$ выполнено

$$|\,\mathbf{Pr}(U|0^n\rangle, A) - \mathbf{Pr}(\tilde{U}(|0^n\rangle \otimes |0^N\rangle), A)| < 3\delta^2 < 6\delta.$$

Таким образом, доказано следующее утверждение о статистической близости распределений, порождаемых оператором и его приближением.

Утверждение 3. *Если унитарный оператор U приближается в расширенном смысле унитарным оператором $\tilde{U}$ с точностью δ, то для любого события A вероятности этого события при измерении $U|0^n\rangle$ и $\tilde{U}(|0^n\rangle \otimes |0^N\rangle$ различаются на $O(\delta)$.*

Замечание 1. Обратное к утверждению 3, разумеется, неверно. Например, операторы

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{и} \quad \begin{pmatrix} i & 0 \\ 0 & 1 \end{pmatrix}$$

порождают одинаковые распределения в вычислительном базисе при действии на вектор $|0\rangle$. Поэтому нас, вообще говоря, устроило бы и более слабое определение близости операторов. Но пока неясно, какую пользу приносит такое ослабление.

7.2. Конечные универсальные базисы

Теперь вернемся к конечным базисам. Будем использовать следующее определение.

Определение. Конечный базис $\mathcal{B}$ называется *универсальным*, если любой унитарный оператор U с точностью до скалярного множителя приближается в расширенном смысле с любой точностью ε схемами в базисе $\mathcal{B}$.

В этом определении мы учитываем, что скалярный множитель не меняет вероятностного распределения, порождаемого оператором.

Основной факт состоит в том, что конечные универсальные базисы существуют.

Теорема 1 (об универсальном конечном базисе). *Базис* $\{\text{c-NOT}, H, K(\pi/4)\}$ — *универсальный. Здесь*

$$\text{c-NOT}\colon |x, y\rangle \mapsto |x, y \oplus x\rangle, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad K(\pi/4) = \begin{pmatrix} 1 & 0 \\ 0 & e^{\pi i/4} \end{pmatrix}.$$

Из теоремы предыдущей лекции 6.3 (любой оператор выражается в базисе из однокубитовых операторов и c-NOT) следует, что для доказательства теоремы 1 достаточно доказать, что любой однокубитовый оператор приближается произведениями H и $K(\pi/4)$.

А поскольку фазовый множитель несущественен, достаточно приближать операторы с определителем 1. Но, как было показано в прошлый раз, $\mathbf{SU}(2) \cong \mathbf{SO}(3)$. Поэтому утверждение теоремы сводится к утверждению о приближении поворотов с помощью некоторых двух данных поворотов.

Начнем с замечания о порождении всей подгруппы $\mathbf{SO}(3)$.

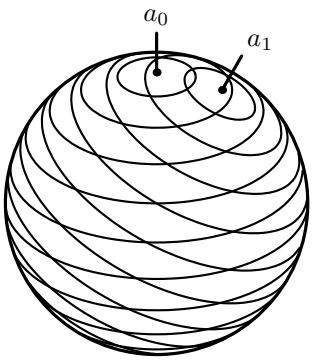


Рис. 1.

Утверждение 4. *Группа* $\mathbf{SO}(3)$ *поворотов трехмерного пространства порождается поворотами относительно любых двух неколлинеарных осей.*

Доказательство легко увидеть из рисунка 1, если заметить, что достаточно проверить транзитивность действия на сфере группы, порожденной поворотами вокруг двух осей. Действительно, если $U\colon a \mapsto a_0$, то

$$R_\varphi(a) = U^{-1} R_\varphi(a_0) U.$$

Транзитивность видна из рисунка 1: двигаясь по двум несовпадающим системам параллелей, можно добраться в любую точку сферу из полюса.

Задача 3. Докажите, что для перевода любого вектора в a_0 достаточно $O(1/\vartheta)$ поворотов вокруг a_0, a_1, где ϑ — угол между a_0 и a_1.

Итак, повороты вокруг двух осей порождают всю группу поворотов $\mathbf{SO}(3)$. Нам, однако, хочется иметь конечное множество поворотов, которое порождает всюду плотное множество в группе $\mathbf{SO}(3)$. Достаточно построить такое множество для группы поворотов $\mathbf{SO}(2)$.

Приведем хорошо известный факт.

Теорема 2. *Если угол* α *несоизмерим с* π*, то любой поворот* R_φ *приближается некоторым кратным* R_α^n *с точностью* ε*.*

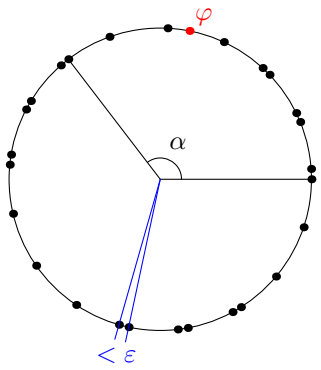


Рис. 2.

Напомним доказательство этой теоремы (см. рисунок 2). Если $m > 2\pi/\varepsilon$, то найдутся $0 < m', m'' \leqslant m$ такие, что $|R_\alpha^{m'}(0) - R_\alpha^{m''}(0)| < \varepsilon$, т. е. $R_\alpha^{m''-m'}$ — поворот на угол $< \varepsilon$. Однако этот угол отличен от нуля, так как α несоизмерим

с π. Откладывая кратные этого угла, получим приближение любого поворота с точностью по крайней мере ε.

Для доказательства теоремы 1 достаточно проверить, что операторы H и K порождают два поворота вокруг разных осей на угол, несоизмеримый с π. Для этого нужно выполнить некоторые вычисления.

Упражнение 4. Проверьте, что

$$\begin{aligned} &H\sigma_x H^\dagger = \sigma_z; && K(\frac{\pi}{4})\sigma_x K(\frac{\pi}{4})^\dagger = \cos\frac{\pi}{4}\sigma_x + \sin\frac{\pi}{4}\sigma_y \\ &H\sigma_y H^\dagger = -\sigma_y; && K(\frac{\pi}{4})\sigma_y K(\frac{\pi}{4})^\dagger = -\sin\frac{\pi}{4}\sigma_x + \cos\frac{\pi}{4}\sigma_y \\ &H\sigma_z H^\dagger = \sigma_x; && K(\frac{\pi}{4})\sigma_z K(\frac{\pi}{4})^\dagger = \sigma_z \end{aligned} \tag{3}$$

Из соотношений (3) следует, что H действует как поворот на π вокруг оси $(1,0,1)$, а $K(\frac{\pi}{4})$ — как поворот на $-\pi/4$ вокруг оси σ_z.

Теперь вычислим действие композиции $K(\frac{\pi}{4})H$:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \overset{H}{\longmapsto} \begin{pmatrix} z \\ -y \\ x \end{pmatrix} \overset{K(\frac{\pi}{4})}{\longmapsto} \begin{pmatrix} z\cos\frac{\pi}{4} - y\sin\frac{\pi}{4} \\ -z\sin\frac{\pi}{4} - y\cos\frac{\pi}{4} \\ x \end{pmatrix}$$

Ось поворота находим из системы уравнений

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} z\cos\frac{\pi}{4} - y\sin\frac{\pi}{4} \\ -z\sin\frac{\pi}{4} - y\cos\frac{\pi}{4} \\ x \end{pmatrix}.$$

Получаем, что $K(\frac{\pi}{4})H$ действует как поворот вокруг оси

$$\begin{pmatrix} 1 \\ -\sqrt{2}+1 \\ 1 \end{pmatrix}.$$

Чтобы найти угол поворота, подействуем на вектор, перпендикулярный оси поворота:

$$K(\frac{\pi}{4})H\colon \vec{v} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \mapsto \begin{pmatrix} -\cos\frac{\pi}{4} \\ \sin\frac{\pi}{4} \\ 1 \end{pmatrix} = \vec{u}. \tag{4}$$

Из (4) для угла поворота α получаем соотношение

$$\cos\alpha = \frac{1}{2}(\vec{v},\vec{u}) = -\frac{1+\cos(\pi/4)}{2} = -\cos^2\frac{\pi}{8} = -\frac{2+\sqrt{2}}{4}.$$

Задача 5. Докажите, что если $\cos\alpha = -\cos^2\frac{\pi}{8}$, то α несоизмерим с π.

Указание: проверьте по индукции, что $\cos(n\alpha)$ имеет вид $\frac{p_n}{2^{k_n}} + q_n\sqrt{2}$, где p_n — нечетные целые.

Замечание 2. Справедлива следующая теорема Влодарского: если β не является целым кратным $\pi/4$ и $\cos\alpha = \cos^2\beta$, то хотя бы один из углов α, β несоизмерим с π.

Итак, $K(\frac{\pi}{4})H$ действует как поворот на иррациональный угол. Но тогда и $HK(\frac{\pi}{4}) = H(K(\frac{\pi}{4})H)H$ — поворот на тот же угол, но вокруг другой оси (так как ось вращения $K(\frac{\pi}{4})H$ не сохраняется H).

Значит, композиции операторов $K(\frac{\pi}{4})$ и H порождают всюду плотное множество в $\mathbf{SU}(2)$. Теорема об универсальном конечном базисе доказана.

Конечно, универсальных конечных базисов много. Приведем несколько примеров.

Теорема 3 (базис Китаева, [7]). *Базис* $\{\text{cc-NOT}, \text{c-NOT}, H, K(\pi/2)\}$ *— универсальный.*

Доказательство универсальности базиса Китаева технически более сложное и мы его разбирать не будем. Отмечу лишь, что оно основано на использовании следующей леммы, которая заменяет утверждение о порождении $\mathbf{SO}(3)$ поворотами вокруг двух разных осей.

Лемма 1. *Для вектора* $|\xi\rangle \neq 0$ *в унитарном пространстве размерности* $\geqslant 3$ *через* H *обозначим подгруппу унитарных операторов, сохраняющих одномерное подпространство* $\mathbb{C}(\xi)$.

Пусть V *— произвольный унитарный оператор, не сохраняющий подпространство* $\mathbb{C}(\xi)$. *Тогда* $H \cup V^{-1}HV$ *порождает всю группу унитарных операторов на этом пространстве.*

Усилением обеих приведенных теорем являются теоремы Ши.

Теорема 4 (Shi [58] 2002).

1. cc-NOT *и любой однокубитовый оператор, который сохраняет вычислительный базис, образуют универсальный базис.*
2. c-NOT *и любой однокубитовый* T *такой, что* T^2 *не сохраняет вычислительный базис, образуют универсальный базис.*

Следствие.

1. *Базис* $\{\text{cc-NOT}, H\}$ *— универсальный.*
2. *Базис* $\{\text{c-NOT}, F\}$, *где*
$$F = \frac{1}{5}\begin{pmatrix} 4 & -3 \\ 3 & 4 \end{pmatrix},$$
является универсальным.

Из приведенных следствий видно, что теоремы Ши нуждаются в комментарии. Рассмотрим, скажем, базис $\{\text{cc-NOT}, H\}$. В этом базисе все матричные элементы действительные. В каком же смысле этот базис универсальный? (Ясно, что всюду плотное множество в группе $\mathbf{SU}$ такие операторы не порождают.)

Ответ вполне естественный: в теоремах Ши, если сформулировать их аккуратно, речь идет о конечных подмножествах ортогональной группы, порождающих всюду плотное подмножество.

Оказывается, вместо унитарных операторов можно рассматривать ортогональные. Чтобы имитировать применение унитарных операторов с помощью ортогональных, нужно использовать дополнительный кубит, который будет хранить информацию о действительной и мнимой части амплитуды. А именно, вектору $(a+bi)|x\rangle$, $a, b \in \mathbb{R}$, будем сопоставлять $(a|0\rangle + b|1\rangle) \otimes |x\rangle$.

Проверка корректности такого кодирования не очень сложна и оставляется в качестве упражнения.

Упражнение 6.

(i) Проверьте, что для любого унитарного оператора U оператор
$$R(U) = \operatorname{Re}(U) - i\sigma_y[0]\operatorname{Im}(U)$$
является ортогональным в расширенном пространстве.

(ii) Проверьте, что это соответствие сохраняется при произведении операторов:
$$R(UV) = R(U)R(V).$$

(iii) Проверьте, что U и $R(U)$ порождают одинаковые вероятностные распределения при измерении всех кубитов, кроме нулевого (при действии на соответствующие вектора) .

7.3. Эффективные приближения

Результаты предыдущего раздела показывают, что конечного базиса достаточно, чтобы приблизить с любой наперед заданной точностью унитарный оператор композицией базисных операторов.

Однако эти результаты неудовлетворительны, если нас интересует приближение схем в одном базисе схемами в другом базисе. Действительно, из свойств приближений следует, что если каждый оператор в схеме размера ℓ приближается (в расширенном смысле) с точностью ε/ℓ, то реализуемый схемой оператор приближается с точностью ε (линейное накопление ошибок).

Таким образом, нам необходимо строить приближения с точностью, которая стремится к нулю, а не является попросту сколь угодно малой константой.

Посмотрим с этой точки зрения на доказательство теоремы об универсальном базисе. Проблема возникает при приближении произвольного поворота степенью поворота на угол α, несоизмеримый с π. Для приближения схем хотелось бы иметь усиление теоремы 2 примерно следующего вида:

> Оценка скорости приближения: *Если угол α несоизмерим с π, то любой поворот R_φ приближается некоторым кратным R_α^n с точностью ε при $n = O(1/\varepsilon)$.*

Такое утверждение, к сожалению неверно. Величина α может очень хорошо приближаться рациональными числами и тогда достижение точности ε может потребовать гораздо большего количества итераций.

Возникает вопрос: Верна ли оценка приближения $n = O(1/\varepsilon)$ при $\cos\alpha = \cos^2(\pi/8)$?

Положительный ответ на этот вопрос позволил бы усилить теорему об универсальном базисе. По крайней мере про этот базис (на самом деле, и базис Китаева, и базисы Ши, приведенные в следствии, анализируются аналогично и для них будет верно то же самое) можно было бы утверждать, что в нем хорошо приближаются любые другие операторы.

Я не знаю точного ответа на этот вопрос, но скорее всего он положительный. Дело в том, что $\cos\alpha$ и $\sin\alpha$ — алгебраические числа, которые, как и всякие алгебраические числа, плохо приближаются рациональными.

В данном случае «плохо» — это как раз «хорошо», поскольку знаменатели цепных дробей для $\cos\alpha$, $\sin\alpha$ растут не слишком быстро и невозможен указанный выше патологический эффект.

Задача 7 (неизвестной трудности). При $\cos\alpha = \cos^2(\pi/8)$ докажите оценку приближения вида $n = \mathrm{poly}(1/\varepsilon)$.

Однако отмеченная трудность может быть радикально преодолена. А именно имеет место следующая теорема.[1)]

Теорема 5 (Китаев – Соловей). *Для любого $\nu > 0$ справедливо следующее.*

Пусть имеется конечный базис $\mathcal{B}$, замкнутый относительно взятия обратного оператора, операторы которого порождают всюду плотное подмножество $\mathbf{SU}(M)$, $M \geqslant 2$.

Тогда любой оператор из $\mathbf{SU}(M)$ приближается с точностью δ схемой в базисе $\mathcal{B}$ размера $L = O(\exp(O(M^2))\log(1/\delta)^{3+\nu})$.

Более того, существует алгоритм, который порождает описание приближающей схемы за время $O(L)$.

[1)] По-видимому, первым обнаружил этот факт Р. Соловей [61] в 1995 году. Первое опубликованное доказательство появилось в обзоре Китаева [6], 1997. Доказательство можно найти в книге Нильсена и Чанга [11] и в исправленном и дополненном английском переводе книги [7].

Следствие. *Операторы любого конечного универсального базиса приближаются в любом другом конечном универсальном базисе схемами полилогарифмического размера от точности. (В данном случае $M = O(1)$.)*

Доказательство теоремы Китаева – Соловея довольно трудное и требует аккуратных оценок. Поэтому я ограничусь лишь некоторыми комментариями к ней.

Первый вопрос: куда девается отмеченная выше неэффективность? Ведь теорему можно применить и к случаю «плохих» поворотов на угол, очень хорошо приближающийся рациональными числами. Оказывается, что вся «неконструктивная» часть помещается в неявные константы $O(\cdot)$.

Первым шагом в доказательстве теоремы является построение ε-сети на группе $\mathbf{SU}(M)$ при достаточно малом ε. Причем эта сеть должна быть достаточно разреженной и содержать $O(\varepsilon^{-M^2})$ элементов. (Расстояние на группе определяется как операторная норма разности операторов.) Порождение такой сети может занять очень большое время, оценка которого как раз зависит от базиса.

После того как требуемая ε-сеть построена, можно уже довольно быстро строить хорошие приближения. Здесь существенна некоммутативность группы (поэтому для двумерной группы поворотов доказательство не проходит).

Конструкция основана на трех идеях.

1. *Иерархическое приближение.* Строится последовательность ε_k-сетей, после чего приближение оператора строится последовательно: оператор U приближается в самой грубой ε_0-сети оператором V_1, затем $V_1^{-1}U$ приближается в следующей по мелкости ε_1-сети и т. д.

2. *Коммутаторы для построения очень мелких сетей.* По сетям Γ_1, Γ_2 строится коммутатор

$$[\Gamma_1, \Gamma_2] = \{W\colon W = UVU^{-1}V^{-1},\ U \in \Gamma_1,\ V \in \Gamma_2\},$$

   который дает сеть очень высокого разрешения в очень малой окрестности единичного оператора.

3. *«Телескопирование».* Из подходящих сетей окрестностей единичного оператора взятием произведения сетей $\Gamma_1\Gamma_2$ конструируется сеть одновременно и достаточно мелкая, и покрывающая достаточно большую окрестность единицы.

7.4. Окончательное определение квантового алгоритма

После подробного изучения свойств квантовых схем можно уточнить определение квантового алгоритма. На рисунке 3 изображен один цикл обращения к квантовому ресурсу.

Здесь U_{b_k} — операторы из конечного универсального базиса; S_k — множество кубитов, на которые действует k-й оператор; x — результат измерения,

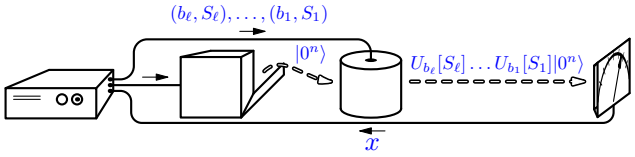


Рис. 3. Цикл обращения к квантовому ресурсу

вероятность наблюдения x

$$\mathbf{Pr}(U_{b_\ell}[S_\ell]\dots U_{b_1}[S_1]|0^n\rangle, x) = \left|\langle x|U_{b_\ell}[S_\ell]\dots U_{b_1}[S_1]|0^n\rangle\right|^2.$$

Время выполнения отмеченного на рисунке цикла действий: $O(\ell)$.

На самом деле такая схема использования квантового ресурса переусложнена. Достаточно одного такого цикла обращения к квантовому ресурсу.

Действительно, как мы уже знаем, все классические вычисления моделируются подходящими квантовыми схемами (через построение схем, а потом обратимых схем, которые реализуются квантовыми схемами).

Все промежуточные измерения можно моделировать подходящими квантовыми схемами. Для этого нужно после измерения некоторого кубита применять лишь операторы вида

$$U\colon |b\rangle \otimes |\psi\rangle \mapsto |b\rangle \otimes U_b|\psi\rangle.$$

(при необходимости такие операторы приближаются в используемом базисе).

Основываясь на этом наблюдении дадим следующее определение квантового алгоритма.

Определение. *Квантовый алгоритм* Q: это классический алгоритм A, который по входу x строит описание квантовой схемы C_x в универсальном конечном базисе, реализующей оператор U_x на n_x кубитах, и описание регистра результата S_x.

Время работы алгоритма на входе x: время работы A плюс размер схемы C_x.

Вероятность результата y *на входе* x:

$$\mathbf{Pr}(y \mid x) = \sum_{z:z[S_x]=y} \left|\langle x|U_x|0^{n_x}\rangle\right|^2.$$

Алгоритм *вычисляет функцию* $f(x)$ с вероятностью ошибки ε, если

$$\mathbf{Pr}(y \neq f(x) \mid x) < \varepsilon.$$

Сделаем несколько замечаний к этому определению. Во-первых, следует отметить, что более распространено определение, в котором алгоритм строит описание единой схемы для всех входов длины n и на вход квантовой схемы вместо $|0^n\rangle$ подается $|x\rangle$. Эти определения эквивалентны (проверьте!).

Во-вторых, как и в случае вероятностных вычислений, квантовые алгоритмы допускают быстрое понижение вероятности ошибки.

Задача 8. Пусть квантовый алгоритм Q вычисляет $f(x)$ с вероятностью ошибки $\varepsilon < 1/2$.

Алгоритм Q'_s работает следующим образом:

1. s раз независимо повторить алгоритм Q;
2. выдать результатом то значение y, которое встретилось чаще всего.

Докажите, что Q'_s вычисляет $f(x)$ с вероятностью ошибки $< (2\sqrt{\varepsilon(1-\varepsilon)})^s$.

Исходя из этого результата, мы будем дальше строить алгоритмы с вероятностью ошибки $< 1/3$, поскольку экспоненциальное увеличение точности потребует лишь полиномиального увеличения времени работы алгоритма.

Лекция 8. Факторизация чисел

Имея определение квантового алгоритма, нужно вернуться к вопросу о сравнении возможностей классической и квантовой модели. При этом, как обычно в теории вычислительной сложности, нас интересуют сверхполиномиальные разрывы. (Более слабые разрывы, — например, квадратичные — существуют и между различными вариантами классических моделей.)

Самым известным примером задачи, для которой построены полиномиальные квантовые алгоритмы, а все известные классические алгоритмы имеют (суб)экспоненциальную сложность, является задача разложения целого числа на множители (факторизации).

Наша основная цель сейчас — построить полиномиальный квантовый алгоритм факторизации. Есть два подхода к построению такого алгоритма. Мы рассмотрим тот, который основан на оценках собственных чисел унитарных операторов.

8.1. Алгоритмы оценки фазы (собственного числа)

Рассмотрим такую (не вполне формальную) задачу. Имеется унитарный оператор U и его собственный вектор $|\psi\rangle$ с собственным числом $\lambda = \exp(2\pi i\varphi)$:

$$U|\psi\rangle = \lambda|\psi\rangle.$$

Мы хотим определить собственное число λ или, что то же самое, его фазу (аргумент). Как это сделать? Рассмотрим следующую квантовую схему:

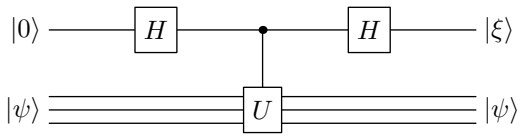


Рис. 1. Схема для косинуса

Как указано на рисунке, если на вход такой схемы подается вектор $|0\rangle \otimes |\psi\rangle$, то в результате применения схемы второй регистр не изменяется (поскольку умножение на собственное число можно учесть в первом регистре). Найдем изменение первого регистра:

$$|\xi\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & \lambda\end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}|0\rangle = \\ = \frac{1}{2}\begin{pmatrix}1+\lambda & 1-\lambda\\ 1-\lambda & 1+\lambda\end{pmatrix}|0\rangle = \frac{1}{2}\begin{pmatrix}1+\lambda\\ 1-\lambda\end{pmatrix}.$$

Предположим, что после применения схемы для косинуса мы измерили значение первого кубита. По определению, вероятность наблюдения 0 равна

$$\mathbf{Pr}(|\xi\rangle, 0) = \left|\frac{1+\lambda}{2}\right|^2 = \frac{1}{4}\left((1+\cos(2\pi\varphi))^2 + \sin(2\pi\varphi)^2\right) = \frac{1+\cos(2\pi\varphi)}{2}\,. \quad (1)$$

Мы видим из (1), что интересующая нас фаза, а точнее — ее косинус, связана с вероятностью наблюдения 0. Для оценки $\cos\varphi$ нужно найти оценку вероятности 0. Делается это обычным способом: повторим действие много раз и возьмем среднюю частоту нуля. Она и будет хорошим приближением к $\cos\varphi$.

Итак, применим основную схему к s различным управляющим кубитам, после чего измерим каждый из них.

Поскольку состояние управляющих кубитов после применения основной схемы $|\xi\rangle^{\otimes s}$, результаты измерений независимы, вероятность 0 в каждом кубите равна $p = (1 + \cos(2\pi\varphi))/2$. То есть мы получили схему испытаний Бернулли.

Насколько точно отношение числа нулей среди исходов измерения к s приближает p? Ответ дает известная теорема.[1)]

Теорема 1 (оценка Чернова). *Пусть проведена серия из s испытаний Бернулли с вероятностью успеха p. Вероятность отклонения частоты ν (отношения числа успехов к s) от вероятности оценивается как*

$$\mathbf{Pr}\left[|\nu - p| > \delta\right] < 2e^{-2\delta^2 s}.$$

Применение основной схемы позволяет оценить величину $\cos\varphi$. Чтобы оценить саму фазу, нам нужна еще оценка синуса. Для этого нужно поменять местами действительную и мнимую части λ:

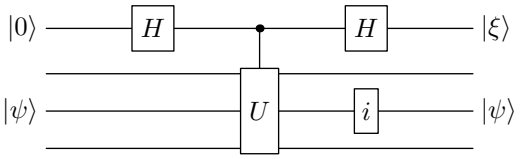


Рис. 2. Схема для синуса

Аналогично предыдущему получим

$$\mathbf{Pr}(|\xi\rangle, 0) = \left|\frac{1 + i\lambda}{2}\right|^2 = \frac{1 - \sin(2\pi\varphi)}{2}.$$

В итоге получаем следующее.

Утверждение 1. *Фаза φ собственного числа λ оценивается с точностью δ и вероятностью ошибки $< \varepsilon$ за $2s$ применений схем косинуса и синуса, где $s = O(\delta^{-2}\log(1/\varepsilon))$.*

Заметим, что вероятность ошибки уменьшается очень быстро. Но точность приближения уменьшается медленно: оценка с точностью 2^{-n} потребует экспоненциального времени (размера схемы).

Чтобы получить очень точное приближение фазы за разумное время, нужны дополнительные средства. Предположим, что нам доступен не только сам оператор U, но и все его степени.

Собственный вектор $|\psi\rangle$ оператора U остается собственным вектором и для всех степеней: если $U|\psi\rangle = \lambda|\psi\rangle$, то $U^k|\psi\rangle = \lambda^k|\psi\rangle$. Собственное число возводится в ту же степень, что и оператор. Поэтому фаза умножается на показатель степени k.

Оценим с точностью $\delta < \pi/8$ фазы $\varphi_k = 2^k\varphi$ степеней U^{2^k} при $k = 0, \ldots, n$.

Утверждение 2. *По этим данным можно оценить фазу $\varphi = \varphi_1$ с точностью $\pi/2^{n+3}$.*

Для доказательства утверждения 2 используем следующее наблюдение.

Лемма 1. *Если $|y - 2\varphi| < \delta < \pi$, то либо $|y' - \varphi| < \delta/2$, либо $|y'' - \varphi| < \delta/2$, где y', y'' — решения уравнения $2x \equiv y \pmod{2\pi}$.*

Доказательство леммы очевидно из рисунка 3.

[1)] Оригинальная статья [38]. Результат стал уже классическим. Посмотреть доказательство можно, например, в книге Алона и Спенсера [1, с. 287].

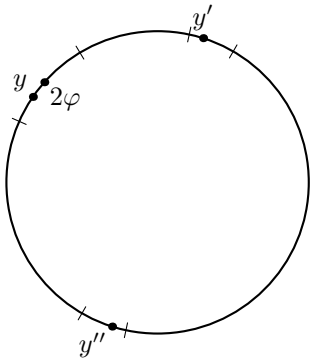


Рис. 3.

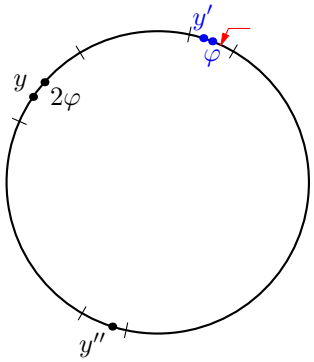


Рис. 4.

Доказательство утверждения 2. Применяем лемму, начиная с приближения $\varphi_n = 2^n\varphi$. Каждый раз мы должны выбирать между двумя возможными решениями. На k-м шаге делаем этот выбор, исходя исходя из δ-приближения φ_{n-k} (см. рис. 4). Однозначность такого выбора обеспечивает точность приближения $< \pi/4$. □

Мы построили два алгоритма для оценки фазы. В обоих случаях анализ проводился в предположении, что на вход оператору U или его степени подается собственный вектор. А что произойдет на другом входе?

Пусть $|\psi\rangle = \sum_{k=1}^{N} c_k|\psi_k\rangle$, где $|\psi_k\rangle$ — собственные вектора оператора U единичной длины: $U|\psi_k\rangle = \exp(2\pi i\varphi_k)|\psi_k\rangle$.

Утверждение 3. *Алгоритм оценки фазы при работе на векторе $|\psi\rangle$ с вероятностью $|c_k|^2$ выдает оценку фазы собственного числа λ_k.*

Доказательство. Напомним, что собственные векторы унитарного оператора ортогональны. После применения основной схемы вектор $|0\rangle \otimes |\psi\rangle = |0\rangle \otimes \sum_{k=1}^{N} c_k|\psi_k\rangle$ перейдет в вектор

$$\sum_{k=1}^{N} c_k|\xi_k\rangle \otimes |\psi_k\rangle, \tag{2}$$

слагаемые в (2) также ортогональны. Поэтому вероятность наблюдения нуля после применения схемы косинуса равна

$$\mathbf{Pr}(|\psi\rangle, 0) = \sum_k |c_k|^2 \frac{1 + \cos(2\pi\varphi_k)}{2} .$$

Что происходит при серии измерений? Раскладывая по ортогональной системе собственных векторов, убеждаемся, что амплитуда для набора исходов $x_1, \ldots, x_s$ равна

$$\sum_k c_k \prod_{j=1}^{s} (1 + (-1)^{x_j}\lambda_k)/2$$

(для каждого собственного вектора в управляющих кубитах получаем разложимое состояние $|\xi_k\rangle^{\otimes s}$).

Поэтому вероятность наблюдения набора исходов $x_1, \ldots, x_s$

$$\mathbf{Pr}(|\psi\rangle, (x_1, \ldots, x_s)) = \sum_k |c_k|^2 \prod_{j=1}^{s} \frac{1 + (-1)^{x_j}\cos(2\pi\varphi_k)}{2}.$$

Но такую же вероятность дает классический процесс: выбрать с вероятностью $|c_k|^2$ параметр $p_k = (1+\cos(2\pi\varphi_k))/2$ и провести s испытаний Бернулли с этим параметром. Отсюда следует утверждение, поскольку дальнейшие (классические) действия оценивают параметр испытаний Бернулли. □

Суммируем основные свойства алгоритмов оценки фазы:

- Используя только оператор U можно оценить фазу собственного числа с точностью δ и вероятностью ошибки ε за время $O(\delta^{-2}\log(1/\varepsilon)$.
- Используя операторы U^{2^k}, $k = 0,\ldots,n = O(\log(1/\delta))$, можно оценить фазу собственного числа с точностью δ и вероятностью ошибки ε за время $O(\log(1/\delta)\log\log(1/\delta)\log(1/\varepsilon))$ (повторный логарифм возникает из-за необходимости оценивать фазу для каждого U^{2^k} с вероятностью ошибки $< \varepsilon/n$).
- Если применять алгоритм оценки фазы к линейной комбинации собственных векторов $|\psi\rangle = \sum_{k=1}^{N} c_k|\psi_k\rangle$, то в результате работы алгоритма с вероятностью $|c_k|^2$ получается оценка фазы φ_k.

8.2. Алгоритм нахождения периода

Чтобы применять алгоритмы оценки фазы, нужно строить операторы, собственные числа которых содержат полезную для решения задачи информацию. Примером такого алгоритма является алгоритм нахождения периода.

Задача нахождения периода формулируется следующим образом.

Даны: двоичные записи чисел q, a, где $a < q$, $(a,q) = 1$ ((a,q) обозначает наибольший общий делитель).

Найти: *период* a относительно q, т. е. такое наименьшее неотрицательное число t, что $a^t \equiv 1 \pmod q$.

Другими словами, период — это порядок числа a в мультипликативной группе вычетов $(\mathbb{Z}/q\mathbb{Z})^*$.

Будем обозначать период числа a относительно q как $\mathrm{per}_q(a)$.

Нас интересует время работы алгоритмов, решающих задачу нахождения периода в зависимости от длины записи входа. Вместо длины записи входа будем использовать величину $n = 1 + \lceil \log q \rceil$, которая отличается от длины входа на множитель $O(1)$.

При построении квантового алгоритма нахождения периода нам потребуются некоторые стандартные теоретико-числовые алгоритмы (классические).[2)] А именно, мы будем использовать эффективные (работающие за полиномиальное время) классические алгоритмы для решения следующих задач (числа заданы двоичными записями):

- Найти значения арифметических операций над целыми числами.
- По x, q найти $x \bmod q$.
- По x, n, q найти $x^n \bmod q$.
- Проверить, является ли x точной степенью (т. е., что существует y и $k \geqslant 2$ такие, что $x = y^k$).
- По x, y найти наибольший общий делитель (x, y).

[2)] Эти алгоритмы нетрудно построить самостоятельно, они хорошо известны и описаны во многих книгах. Прочитать о них стоит в книгах [8, 29] (последняя является стандартным источником по алгоритмической теории чисел, но она, к сожалению, не переведена на русский язык).

«Квантовую» часть алгоритма нахождения периода будет представлять алгоритм точной оценки фазы для оператора циклического сдвига.

Более точно, рассмотрим оператор на n кубитах

$$U_a\colon |x\rangle \mapsto |ax \bmod q\rangle. \tag{3}$$

Как было сказано, для U_a существует схема полиномиального размера. Более того, существует схема полиномиального от m, n размера для $U_a^{2^m}$, так как $U_a^{2^m}\colon |x\rangle \mapsto |a^{2^m} x \bmod q\rangle$.

Поэтому в квантовых схемах для оценки фазы операторы U^k можно заменять на схемы полиномиального размера.

Нас будет интересовать действие оператора (3) не на всех векторах, а только на пространстве, натянутом на векторы $|1\rangle$, $|a\rangle$, $|a^2\rangle$, …, $|a^{t-1}\rangle$. Легко видеть, что это пространство является инвариантным пространством U_a, который циклически переставляет указанные векторы.

Утверждение 4. *Собственные числа циклической перестановки*

$$C_t\colon |j\rangle \mapsto |j+1 \bmod t\rangle$$

равны $\exp(2\pi ik/t)$.

Собственные векторы, отвечающие этим собственным числам

$$|\xi_k\rangle = \frac{1}{\sqrt{t}} \sum_{j=0}^{t-1} \exp(-2\pi ikj/t)|j\rangle.$$

Доказательство. Прямым вычислением проверим, что каждый из указанных векторов — собственный с указанным собственным числом:

$$\begin{aligned} C_t|\xi_k\rangle &= \frac{1}{\sqrt{t}} \sum_{j=0}^{t-1} \exp(-2\pi ikj/t) C_t|j\rangle = \\ &= \frac{1}{\sqrt{t}} \sum_{j=0}^{t-1} \exp(-2\pi ikj/t)|j+1 \bmod t\rangle = \frac{1}{\sqrt{t}} \sum_{j=0}^{t-1} \exp(-2\pi ik(j-1)/t)|j\rangle = \\ &= \exp(2\pi ik/t) \cdot \frac{1}{\sqrt{t}} \sum_{j=0}^{t-1} \exp(-2\pi ikj/t)|j\rangle = \exp(2\pi ik/t)|\xi_k\rangle \end{aligned}$$

Поскольку найдены t собственных чисел (вида $\exp(2\pi ik/t)$), то других собственных чисел нет. □

Алгоритм нахождения периода.

1°. $\ell = 5$ раз применим алгоритм оценки фазы с точностью 2^{-2n-2} и вероятностью ошибки $< 1/32$ к оператору U_a и вектору $|1\rangle$. В ответе получим некоторые дроби p_j/q_j, $j = 1, \ldots, \ell$.

2°. Для каждой дроби p_j/q_j найдем ближайшую дробь k'_j/t'_j со знаменателем $< 2^n$ (используя разложение в цепную дробь).

3°. Выдадим в качестве ответа наименьшее общее кратное чисел t'_j.

Проанализируем работу этого алгоритма по шагам.

В силу анализа алгоритма оценки фазы и возможности вычислять U^{2^m} за полиномиальное от m, n время первый шаг алгоритма выполняется за полиномиальное от длины входа время.

Упражнение 1. Проверьте, что для циклической перестановки C_t

$$|0\rangle = \frac{1}{\sqrt{t}} \sum_{k=0}^{t-1} |\xi_k\rangle \ . \tag{4}$$

Для оператора U_a вектор $|1\rangle$ играет такую же роль, как вектор $|0\rangle$ для циклической перестановки (по циклу переставляются показатели).

Из (4) следует, что результаты измерений в п. 1 алгоритма с вероятностью ошибки $< 5/32$ дают приближения с точностью 2^{-2n+2} к числам k_j/t, где каждое k_j распределено равномерно на множестве $\{0, \ldots, t-1\}$.

Для анализа второго шага алгоритма напомним свойства цепных дробей.

Утверждение 5. *Каждое действительное число α раскладывается в* цепную дробь

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ldots}} \ ,$$

подходящие дроби *имеют вид*

$$\frac{p_k}{q_k} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots + \cfrac{1}{a_k}}}} \ .$$

Теорема 2. *Для подходящих дробей выполняются неравенства*

$$\frac{p_0}{q_0} < \frac{p_2}{q_2} < \cdots < \alpha < \cdots < \frac{p_3}{q_3} < \frac{p_1}{q_1} \quad \textit{и} \quad \left| \frac{p_k}{q_k} - \frac{p_{k-1}}{q_{k-1}} \right| = \frac{1}{q_k q_{k-1}} \ .$$

Следствие.

$$\left| \alpha - \frac{p_{k-1}}{q_{k-1}} \right| < \frac{1}{q_{k-1}^2} \ .$$

Теорема 3. *Если $|\alpha - p/q| < 1/(2q^2)$, то p/q — подходящая дробь для α.*

Говоря неформально, приведенные факты означают, что подходящие дроби дают очень точные приближения к α и, наоборот: любое достаточно точное приближение к α обязательно встретится среди подходящих дробей.[3)]

Заметим также, что подходящие дроби для рационального α вычисляются за полиномиальное от длины записи α время (как в алгоритме Евклида).

Вернемся к анализу шага 2 алгоритма нахождения периода. Поскольку число p_j/q_j отличается от фазы k_j/t не более, чем на 2^{-2n+2}, а $t < 2^n$, то разложение в цепную дробь даст несократимую дробь $k'_j/t'_j = k_j/t$, где k_j — случайный (равномерно распределенный числитель).

Теперь перейдем к анализу шага 3. Осталась ровно одна трудность: случайный числитель k_j может иметь общий делитель со знаменателем t. Тогда t'_j будет каким-то собственным делителем t. Ясно, что НОК t'_j также будет делителем t. Оказывается, с большой вероятностью НОК будет просто равно t.

Лемма 2. *Пусть по равномерному распределению независимо выбраны числа $0 \leqslant k_j < t$, $1 \leqslant j \leqslant \ell$, $\ell \geqslant 2$.*

Тогда вероятность того, что наименьшее общее кратное знаменателей несократимых дробей, равных k_j/t, отлично от t, меньше $\frac{\pi^2}{3} \cdot 2^{-\ell}$.

[3)] Свойства цепных дробей хорошо известны. На всякий случай укажем книги [2, 16], в которой можно прочитать доказательства этих фактов.

Доказательство. Вероятность того, что $k_1, \ldots, k_\ell$ имеют общий простой делитель p, не больше, чем $1/p^\ell$.

Поэтому вероятность того, что $k_1, \ldots, k_\ell$ не взаимно просты в совокупности (равносильно тому, что НОК не равен t), не выше

$$\sum_{k=2}^{\infty} \frac{1}{k^\ell} < \frac{\pi^2}{3} \cdot 2^{-\ell} ,$$

что и требовалось доказать. □

Заметим также, что в силу соотношения

$$(x, y) \cdot [x, y] = xy$$

НОК находится за полиномиальное от длины входа время с помощью алгоритма Евклида.

Поэтому общее время работы алгоритма нахождения периода ограничено полиномом от длины входа.

Оценим вероятность ошибки. Она не превосходит

$$\left(\frac{\pi^2}{3} + 5\right) \cdot \frac{1}{32} < \frac{1}{3} .$$

Здесь первое слагаемое оценивает ошибку на шаге 3 с помощью леммы 2, а второе — ошибку на первом шаге (приближение к фазе).

Теорема 4. *Существует полиномиальный квантовый алгоритм нахождения периода, вероятность ошибки которого* $< 1/3$.

Классического полиномиального алгоритма нахождения периода на данный момент не найдено. Таким образом, мы получили первый пример задачи, в которой квантовый ресурс дает сверхполиномиальное улучшение по сравнению с известными классическими алгоритмами.

8.3. Сводимость задачи факторизации к задаче нахождения периода

Напомним формулировку задачи разложения числа на простые множители (факторизации).

Дано: натуральное число y в двоичной записи.

Найти: разложение y на простые множители

$$y = p_1^{\alpha_1} p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k}.$$

Для задачи факторизации неизвестны эффективные классические алгоритмы. И есть довольно веское прагматическое свидетельство в пользу того, что таких алгоритмов вообще нет.

Дело в том, что на предположении о трудности задачи факторизации основаны практические алгоритмы криптографии (шифрование с открытым ключом, например).

Если бы существовал нетрудоемкий алгоритм ее решения, кто-нибудь уже взломал бы RSA.

Теорема 5. *Существует полиномиальная вероятностная сводимость задачи факторизации к задаче нахождения периода.*

Следствие. *Существует полиномиальный квантовый алгоритм факторизации числа.*

Итак, квантовый алгоритм факторизации числа использует сводимость из теоремы 5 и описанную выше процедуру нахождения периода.

Замечание 1. Исходный квантовый алгоритм Шора [59] также использовал эту сводимость. Квантовая часть (нахождение периода) была устроена иначе и основывалась на преобразовании Фурье (переход к базису собственных векторов для циклического сдвига).

Алгоритм Шора также включает использование цепных дробей, которое вносит основной вклад в общее время вычисления.

Опишем сводимость из теоремы 5. Мы построим процедуру, которая использует подпрограмму вычисления периода и находит некоторый нетривиальный делитель числа y с вероятностью $\geqslant 1/2$ или сообщает, что y простое. Из этой процедуры уже легко получить полиномиальный алгоритм факторизации:

1. Повторяя процедуру нахождения делителя s раз, уменьшаем вероятность ошибки до 2^{-s}.

2. Применяем такую усиленную процедуру $O(\log y)$ раз (оценка количества множителей в разложении на простые), вероятность ошибки ухудшается до $O(\log y/2^s)$, но этого достаточно.

Процедура нахождения делителя.

1°. Проверяем четность y. Если y — четное, то выдаем ответ «2».

2°. Проверяем, извлекается ли из y нацело корень k-й степени при $2 \leqslant k \leqslant \log_2 y$. Если $y = m^k$, то ответ «m».

3°. Выбираем случайно и равновероятно a среди чисел от 1 до y, вычисляем $r = \mathrm{per}_y(a)$ (используя имеющийся по предположению алгоритм нахождения периода) и, если r — нечетное, то ответ «y — простое».

4°. Если же r четное, то находим $d = (a^{r/2} - 1, y)$ алгоритмом Евклида и если $d > 1$, то ответ «d», иначе ответ «y — простое».

Из построения ясно, что эта процедура работает за полиномиальное время. Ясно также, что неправильным может быть лишь ответ «y — простое». Осталось оценить вероятность ошибки.

Лемма 3. *Процедура нахождения делителя выдает собственный делитель y с вероятностью не меньше $1 - 1/2^{k-1}$, где k — число различных простых делителей y. (Для простого y эта вероятность равна 0.)*

В доказательстве леммы 3 нам потребуются факты из теории чисел.

Теорема 6 (китайская теорема об остатках). *Если $y = \prod_{j=1}^{k} p_j^{\alpha_j}$ — разложение на простые, то*

$$(\mathbb{Z}/y\mathbb{Z})^* \cong \prod_{j=1}^{k} (\mathbb{Z}/p_j^{\alpha_j})^*.$$

Теорема 7 (цикличность мультипликативной группы вычетов по модулю p^α). *Группа $(\mathbb{Z}/p^\alpha\mathbb{Z})^*$ — циклическая, т. е. существует такой вычет g, что*

$$(\mathbb{Z}/p^\alpha\mathbb{Z})^* = \{g^s : 0 \leqslant s < p^\alpha - p^{\alpha-1}\}.$$

Эти теоремы хорошо известны, их доказательства можно найти в книгах по теории чисел, например, в [4] или в уже упоминавшейся книге Баха и Шаллита [29].

Доказательство леммы 3. В каких случаях процедура нахождения делителя выдает ответ «y — простое»?

Введем обозначения $r_j = \mathrm{per}_{(p_j^{\alpha_j})}\, a = 2^{s_j} r'_j$, где r'_j — нечетное. В этих обозначениях удобно формулируется ответ: процедура сообщает, что y простое тогда и только тогда, когда

$$s_1 = s_2 = \cdots = s_k.$$

Действительно, заметим, что период r является НОК r_j по китайской теореме.

Поэтому если $r = \mathrm{per}_y(a)$ — нечетное, то r_j нечетное для каждого j. Это случай $s_1 = s_2 = \cdots = s_k = 0$.

Если же $r = \mathrm{per}_y(a)$ — четное, то $(a^{r/2}+1)(a^{r/2}-1) \equiv 0 \pmod{y}$. Так как $a^{r/2} \not\equiv 1 \pmod{y}$ (по определению периода), в этом случае процедура выдаст ответ «y — простое» только тогда, когда $a^{r/2} \equiv -1 \pmod{y}$.

Период r равен НОК всех периодов r_j. Поэтому показатель степени 2, которая входит в разложение r на простые множители, равен $\max_j s_j$.

Заметим, что $a^{r_j/2} \equiv -1 \pmod{p_j^{\alpha_j}}$ (используем цикличность $(\mathbb{Z}/p_j^{\alpha_j}\mathbb{Z})^*$). Если $s_1 = s_2 = \ldots = s_k = \max_j s_j \geqslant 1$, то $a^{r/2} \equiv a^{r_j/2} \equiv -1 \pmod{p_j^{\alpha_j}}$ (используем китайскую теорему об остатках).

С другой стороны, если не все s_j равны, то при некотором m (для которого s_m меньше $\max_j s_j$) получим $a^{r/2} \equiv 1 \pmod{p_m^{\alpha_m}}$, т. е. $a^{r/2} \not\equiv -1 \pmod{y}$.

Теперь осталось оценить вероятность события $s_1 = s_2 = \ldots = s_k$ при случайном выборе a. По китайской теореме об остатках случайный равномерный выбор a есть то же самое, что независимый случайный равномерный выбор всех $a_j = a \pmod{p_j^{\alpha_j}}$.

Оценим вероятность события $s_1 = s$ при независимом выборе a_1. Пусть $p_1^{\alpha_1} - 1 = 2^t q$, q — нечетное, g — образующая циклической группы $(\mathbb{Z}/p_1^{\alpha_1}\mathbb{Z})^*$. Тогда

$$|\{a_1 : s_1 = s\}| = |\{g^{2^{t-s}m} : m \text{ — нечетное}\}| =$$
$$= \begin{cases} q, & \text{если } s = 0, \\ (2^s - 2^{s-1})q = \dfrac{1}{2}2^{s-1}q < \dfrac{1}{2}2^t q, & \text{если } s > 0, \end{cases}$$

поэтому вероятность $s_1 = s$ не больше $1/2$.

Отсюда получаем, что вероятность события $s_1 = s_2 = \ldots = s_k$ не выше $1/2^{k-1}$. □

Из доказанной леммы следует, что ошибки процедуры нахождения делителя $< 1/2$ (ошибка возможна только на составных числах, для которых $k \geqslant 2$).

Лекция 9. Класс BQP

9.1. Другие примеры эффективных квантовых алгоритмов

Алгоритм нахождения периода допускает обобщение на действия абелевых групп.

Напомним, что *действием группы* G на множестве X называется гомоморфизм $G \to S(X)$ в группу биективных преобразований множества X. Другими словами, это функция $\alpha\colon G \times X \to X$, для которой выполняются условия

- $\alpha(g, \cdot)$ — взаимно однозначное отображение X на X;
- $\alpha(g_1 g_2, x) = \alpha(g_1, \alpha(g_2, x))$.

Часто обозначение функции опускается: $\alpha(g, x)$ обозначается как gx.

Стабилизатором действия называется множество $S(x_0) = \{g \in G : gx_0 = = x_0\}$. Легко проверить, что это подгруппа группы G.

Далее мы будем рассматривать действия группы $\mathbb{Z}^n$ и стабилизаторы этих действий. Для конструктивного задания подгрупп $\mathbb{Z}^n$ полезно следующее утверждение.

Утверждение 1. *Любая подгруппа $G < \mathbb{Z}^n$ изоморфна $\mathbb{Z}^k$. Поэтому для G существует базис образующих g_k:*

$$G = \left\{ g : g = \sum_k z_k g_k \right\},\ z_k \text{ определены однозначно.}$$

Задача нахождения стабилизатора действия $\mathbb{Z}^n$ формулируется следующим образом.

Дано: эффективное действие группы $\mathbb{Z}^n$ на множестве $X = \{0,1\}^m$. Это означает, что задан алгоритм вычисления $\alpha(z, x)$, работающий за полиномиальное время. Кроме того, указан элемент $x_0 \in X$.

Найти: базисную систему образующих для $S(x_0)$.

Задача нахождения периода числа по сути является частным случаем задачи нахождения стабилизатора. Рассмотрим действие $\mathbb{Z}$

$$(n, x) \mapsto (a^n x \bmod q).$$

Стабилизатор 1 состоит из чисел, кратных $\mathrm{per}_q(a)$.

Теорема 1. *Существует полиномиальный квантовый алгоритм нахождения стабилизатора действия $\mathbb{Z}^n$.*

Алгоритм нахождения стабилизатора аналогично аналогичен алгоритму нахождения периода. Мы не будем приводить полную конструкцию? ограничившись лишь несколькими замечаниями.

Во-первых, алгоритм ищет двойственную систему образующих для группы характеров стабилизатора, т. е. гомоморфизмов $S \to \mathbf{U}(1)$. Характер имеет вид

$$(z_1, \ldots, z_n) \overset{\varphi}{\longmapsto} \exp\left(2\pi i \sum_j \varphi_j z_j\right),$$

где $\varphi_j \leqslant 1$ — рациональные числа.

Чтобы сгенерировать случайный характер, используется оценка собственных чисел операторов сдвига по каждой компоненте. Эта «квантовая» часть алгоритма устроена точно так же, как в случае нахождения периода.

Аналогично процедуре вычисления периода проверяется, что достаточно большое количество случайных характеров порождает всю группу.

После того как построены образующие группы характеров, образующие стабилизатора находятся из решения системы линейных диофантовых уравнений.

Приведем пример еще одной задачи, которая является частным случаем задачи нахождения стабилизатора.

Определение. Пусть g — образующая $(\mathbb{Z}/p\mathbb{Z})^*$, p — простое. *Дискретный логарифм* $\log_g(a)$ — это наименьшее положительное k такое, что $g^k = a$.

Теорема 2. *Существует полиномиальный квантовый алгоритм вычисления дискретного логарифма.*

Вычисление дискретного логарифма сводится к нахождению стабилизатора. Рассмотрим действие $\mathbb{Z}^2$

$$(z_1, z_2)(x) = g^{z_1} a^{z_2} x \pmod{p}.$$

Найдем стабилизатор этого действия, а в нем — элементы вида $(k, -1)$ (для этого нужно решить систему линейных диофантовых уравнений). Числа k образуют арифметическую прогрессию, дискретный логарифм является наименьшим положительным числом в этой прогрессии.

Из этих примеров видно, что квантовые алгоритмы позволяют использовать скрытые симметрии в условиях задачи. Однако незначительные усложнения формулировок уже приводят к задачам, для которых эффективные квантовые алгоритмы неизвестны (и само их существование сомнительно).

Приведем два примера.

Задача обращения перестановки.

Дано: перестановка $\sigma\colon \{0,1\}^n \to \{0,1\}^n$ задана схемой вычисления.

Найти: обратную перестановку σ^{-1}.

Заметим, что если дана схема вычисления любой степени перестановки, то задача сводится к нахождению стабилизатора действия $\mathbb{Z}$. (Если известен порядок перестановки, то обратная перестановка вычисляется эффективно аналогично тому, как вычисляются остатки от больших степеней по модулю заданного числа.)

Если ослабить наши возможности и отказаться от вычисления произвольных степеней, квантовый алгоритм, основанный на оценке фазы собственного числа перестает работать (так как мы уже не может оценивать фазу с большой точностью). Никаких других идей решения это задачи квантовыми алгоритмами в настоящее время неизвестно и вообще сомнительно, что такие алгоритмы существуют.

Второй пример связан с действиями симметрических групп.

Дано: действие симметрической группы $S_n \times \{0,1\}^n \to \{0,1\}^n$ задано схемой вычисления, элемент $x_0 \in \{0,1\}^n$.

Найти: $S(x_0)$.

К этой задаче сводится задача изоморфизма графов. Настойчивые поиски эффективного квантового алгоритма для этой задачи не увенчались успехом.

Даже если рассмотреть простейшую, в некотором смысле, неабелеву группу D_n (группу симметрий правильного n-угольника), задача о нахождении стабилизатора действия остается очень трудной. Наилучший квантовый алгоритм для D_n (Куперберг [51], 2003) требует времени (и памяти) $2^{O(\sqrt{\log n})}$.

9.2. Классы сложности

Теперь мы посмотрим на квантовые алгоритмы с точки зрения структурной теории вычислительной сложности. Идея, на которой основывается подход

структурной теории сложности, состоит в переходе от анализа конкретных вычислительных задач к анализу классов задач, которые решаются при использовании ресурсов указанного вида в указанном количестве.

Надежда здесь на то, что классов окажется меньше, чем задач, и их соотношения будет легче анализировать. Отметим сразу, что эта надежда оправдалась лишь частично.

Технически проще иметь дело не с вычислительными задачами, а с *языками*: подмножествами слов в некотором алфавите. С языком L связана естественная алгоритмическая задача: дано слово x, проверить, что $x \in L$.

Подробно излагать теорию сложности мы не будем из-за недостатка времени. Поэтому ограничимся несколькими примерами.[1)]

Пример. Класс P**.** Язык L принадлежит классу P тогда и только тогда, когда существует алгоритм проверки $x \in L$, работающий за полиномиальное время.

Класс P — наиболее популярная формализация понятия эффективного вычисления.

Пример. Класс BPP**.** $L \in \mathrm{P}$ равносильно тому, что существует *вероятностный* алгоритм проверки $x \in L$, работающий за полиномиальное время с вероятностью ошибки $< 1/3$.

На данный момент класс BPP — наиболее разумная формализация понятия эффективного вычисления.

Из определения следует, что $\mathrm{P} \subseteq \mathrm{BPP}$. Является ли это включение строгим? На данный момент это один из самых важных вопросов в теории сложности. Разработаны мощные методы *дерандомизации* — перевода алгоритмов, использующих случайные биты, в детерминированные алгоритмы. Однако равенство $\mathrm{P} = \mathrm{BPP}$ доказано лишь по модулю некоторого предположения о другом сложностном классе E — классе языков, которые решаются за *простое экспоненциальное время*, т. е. за время $2^{O(\log n)}$, где n — длина входа.

Предположение состоит в том, что в классе E существуют языки с большой схемной сложностью. Более точно, построим по языку L семейство булевых функций $L_n \colon \{0,1\}^n \to \{0,1\}$, где $L_n(x) = 1$ равносильно $x \in L$. Язык имеет большую схемную сложность, если схемная сложность L_n экспоненциальна $2^{\Omega(n)}$.

Это предположение выглядит естественно, но доказать его известными на данный момент методами не удается.

Самый известный из классов сложности — класс NP.

Пример. Класс NP**.** Язык L принадлежит классу NP тогда и только тогда, когда существует вычислимый за полиномиальное время предикат (функция в $\{0,1\}$) от двух переменных $R(x,y)$ и полином $p(\cdot)$ такие, что

если $x \in L$, то существует такой y, что $|y| \leqslant p(|x|)$ и $R(x,y)$;

если $x \notin L$, то для любого y из $|y| \leqslant p(|x|)$ следует $\neg R(x,y)$.

Неформально принадлежность задаче классу NP можно понимать как существование короткого *доказательства* принадлежности слова языку.

Заметим, что в отличие от классов P и BPP класс NP определяется несимметрично. Формально эта несимметричность выражается следующим образом. По любому классу сложности определим класс сложности со-$\mathcal{C}$, состоящий из языков вида $\bar{L}$, $L \in \mathcal{C}$.

[1)] Стандартный источник знаний по теории сложности на русском языке — книга Гэри и Джонсона [5] — уже не покрывает всех интересных тем теории сложности. Более новые учебники, скажем, книга Сипсера [60] или книга Арори и Барака [27], на русский язык не переводились.

Нетрудно видеть, что P = co-P, BPP = co-BPP. Что касается классов NP и co-NP, то их точное соотношение неизвестно, однако популярной является гипотеза, что NP $\neq$ co-NP.

Для нашего обсуждение особенно важен следующий класс.

Пример. Класс суммирования PP**.** Язык L принадлежит классу PP тогда и только тогда, когда существует вероятностный алгоритм проверки $x \in L$, работающий за полиномиальное время и такой, что

если $x \in L$, то вероятность ответа «да» $> 1/2$;

если $x \notin L$, то вероятность ответа «да» $\leqslant 1/2$.

Для этого класса можно дать равносильное определение.

Утверждение 2. *$L \in$ PP тогда и только тогда, когда существует вычислимый за полиномиальное время предикат (функция в $\{0,1\}$) от двух переменных $R(x,y)$ и полином $p(\cdot)$ такие, что*

если $x \in L$, то количество таких y, что $|y| = p(|x|)$ и $R(x,y)$, больше $2^{p(|x|)}/2$;

если $x \notin L$, то количество таких y, что $|y| = p(|x|)$ и $R(x,y)$, не больше $2^{p(|x|)}/2$.

Чтобы понять, что условия в утверждении 2 равносильны определению, нужно рассмотреть биты слова y как случайные биты для вероятностного алгоритма. Аналогично можно охарактеризовать и класс BPP.

Утверждение 3. BPP $\subseteq$ PP*;* NP $\subseteq$ PP*.*

Доказательство. Первое включение очевидно из определения.

Для доказательства второго рассмотрим язык $L \in$ NP. Пусть R, p – предикат и полином из определения NP, примененного к L.

Тогда L удовлетворяет определению класса PP с полиномом $\tilde{p}(n) = p(n) + 1$ и предикатом

$$\tilde{R}(x, yb) = \begin{cases} 1, & \text{если } b = 0; \\ R(x,y), & \text{если } b = 1. \end{cases}$$

В силу такого определения по крайней мере при половине y предикат $\tilde{R}$ истинен. Поэтому достаточно хотя бы одного y из другой половины, чтобы предикат был истинен более чем в половине случаев. □

Теперь объясним, почему класс PP был назван классом суммирования. Точнее было бы сказать «классом вычисления экспоненциальных сумм», а совсем точная формулировка дается в следующей теореме.

Теорема 3. *Пусть $f_n \colon \{0,1\}^n \mapsto \mathbb{Z}$ — семейство функций, вычислимых за полиномиальное от n время.*

Задача определения знака суммы

$$S_n = \sum_{x \in \{0,1\}^n} f_n(x)$$

лежит в классе PP *(вход — описание алгоритма вычисления f_n и n в унарной системе).*

Замечание 1. Равносильная формулировка получается, если рассмотреть два семейства $f_n\colon \{0,1\}^n \mapsto \mathbb{N}$, $g_n\colon \{0,1\}^n \mapsto \mathbb{N}$, и спрашивать о знаке разности

$$\sum_{x\in\{0,1\}^n} f_n(x) - \sum_{x\in\{0,1\}^n} g_n(x).$$

Доказательство. Поскольку f_n, g_n вычислимы за полиномиальное время, длина L_n их записи ограничена полиномом от входа. Мы полагаем без ограничения общности, что $L_n > 0$.

По f_n, g_n строим предикат $R(x,u,v,b)$, где $x \in \{0,1\}^n$, $0 \leqslant u,v < 2^{L_n}$, $b \in \{0,1\}$. Предикат $R(x,u,v,b)$ истинен тогда и только тогда, когда

$$\begin{aligned} v \geqslant g_n(x) \text{ и } b = 0,\ u = 0, \qquad & \text{или } v \equiv 0 \pmod 2 \text{ и } b = 0,\ u > 0, \\ \text{или } u < f_n(x) \text{ и } b = 1,\ v = 0, \qquad & \text{или } u \equiv 0 \pmod 2 \text{ и } b = 1,\ v > 0. \end{aligned}$$

Вклад каждого x в множество истинных значений предиката R состоит из четырех слагаемых

$$\begin{aligned} &2^{L_n} - g_n(x) && (\text{вклад } b = 0, u = 0), \\ &(2^{L_n} - 1)2^{L_n - 1} && (\text{вклад } b = 0, u > 0), \\ &f_n(x) && (\text{вклад } b = 1, v = 0), \\ &(2^{L_n} - 1)2^{L_n - 1} && (\text{вклад } b = 1, v > 0). \end{aligned}$$

В сумме это дает $f_n(x)+(2^{L_n}-g_n(x))+(2^{L_n}-1)2^{L_n}$. Аналогично подсчитывается вклад x в множество ложных значений предиката R. Он равен $(2^{L_n} - f_n(x)) + + g_n(x) + (2^{L_n} - 1)2^{L_n}$.

Поэтому вклад x в разность $2\big(f_n(x) - g_n(x)\big)$. Суммируя по x получаем искомое. □

Включения между рассмотренными классами изображены на диаграмме 1.

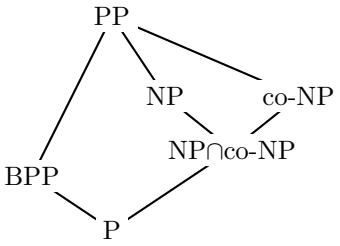


Рис. 1.

9.3. Определение класса BQP. Примеры полных задач

Определение. $L \in$ BQP равносильно тому, что существует *квантовый* алгоритм проверки $x \in L$, работающий за полиномиальное время с вероятностью ошибки $< 1/3$.

Этот класс формализует понятие задачи эффективно разрешимой с использованием квантового ресурса. Поскольку «подбрасывание монетки» моделируется квантовыми алгоритмами, как это объяснялось выше, справедливо следующее включение.

Утверждение 4. BPP $\subseteq$ BQP.

Прежде чем обсудить «верхние границы» сложности для класса BQP, напомним примеры входящих в него задач, предположительно не принадлежащих классу BPP:

- нахождение периода;
- факторизация;
- дискретный логарифм.

В какие классы сложности попадают эти задачи?

Заметим, что определения классов сложности были даны для языков, а приведенные примеры относятся к алгоритмическим задачам вычисления функций.

Будем использовать следующую конструкцию.

Задаче вычисления функции $f\colon \{0,1\}^* \to \{0,1\}^*$ сопоставим язык $L_f = \{\langle x, j\rangle : j\text{-й бит } f(x) \text{ равен 1 и } |f(x)| \geqslant j\}$.

Если $L_f \in \mathcal{C}$ для некоторого класса сложности $\mathcal{C}$, то будем также говорить, что *задача вычисления f лежит в классе $\mathcal{C}$*.

Утверждение 5. *Факторизация лежит в классе* NP ∩ co-NP.

Идея доказательства этого утверждения состоит в том, что за полиномиальное время можно проверить корректность предлагаемого разложения данного числа на простые.

После этого легко определить как равенство нулю некоторого бита (co-NP), так и равенство единице (NP).

Замечание 2. В намеченном выше рассуждении мы опирались на тот факт, что проверка простоты числа принадлежит P. Однако без этого предположения можно обойтись и доказать более простыми рассуждениями, что проверка простоты лежит в классе NP ∩ co-NP.

Утверждение 6. *Вычисление периода лежит в классе* NP ∩ co-NP.

Идея доказательства аналогична предыдущему случаю (задаче факторизации). За полиномиальное время можно проверить, что число, представленное в виде разложения на простые множители, является периодом: $t = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ — период тогда и только тогда, когда $a^t \equiv 1 \pmod q$ и $a^{t/p_j} \not\equiv 1 \pmod q$.

Утверждение 7. *Вычисление дискретного логарифма лежит в классе* NP ∩ co-NP.

Применяем ту же идею. $\log_g(a)$ — единственное положительное число, меньшее q и такое, что $g^s \equiv a \pmod q$. Здесь мы используем, что g является образующей мультипликативной группы вычетов по модулю q, что обещано в определении задачи дискретного логарифмирования. Впрочем, без этого предположения можно обойтись, так как уже показана принадлежность вычисления периода классу NP ∩ co-NP.

А какие задачи «самые трудные» в классе BQP?

Определение. Задачу A будем называть *трудной* для класса $\mathcal{C}$, если любую задачу из этого класса можно свести к A, и *полной* для класса $\mathcal{C}$, если дополнительно к тому A лежит в $\mathcal{C}$.

Замечание 3. В этом определении намеренно вместо слова «язык» использовано менее определенное слово «задача» (алгоритмическая).

Дело в том, что ограничиваясь языками, труднее говорить о полных задачах. Для многих классов: NP ∩ co-NP, BPP и т. д., включая и класс BQP, неизвестно существование полных языков.

Однако ситуация улучшается, если рассматривать более общее понятие задачи с априорными условиями (promise problem). Такая задача задается двумя непересекающимися языками: L_1 (ответ положительный) и L_0 (ответ отрицательный). Для остальных слов ответ не определен и решающий задачу алгоритм может вести себя на этих словах произвольным образом.

Именно в таком смысле мы и понимаем далее слово «задача».

Приведем пример полной задачи для класса NP ∩ co-NP. Даны две КНФ C_0 и C_1. Известно (априорная информация), что ровно одна из них выполнима. Определить, какая именно.

Здесь мы используем тот хорошо известный факт, что задача выполнимости КНФ полна в классе NP.

После этой подготовки мы готовы привести пример BQP-полной задачи (с априорной информацией). Как и следовало ожидать, она формулируется в терминах квантовых схем.

Дано: описание квантовой схемы в стандартном базисе $\{\text{c-NOT}, H, K(\pi/4)\}$.

Известно: вероятность наблюдения 1 в первом кубите либо больше $2/3$, либо меньше $1/3$.

Выяснить: какой из случаев имеет место.

BQP-полнота этой задачи следует непосредственно из определений.

Следующий пример тоже связан с квантовыми схемами, но чуть менее очевиден. Это задача оценки матричного элемента.

Дано: описание квантовой схемы в стандартном базисе $\{\text{c-NOT}, H, K(\pi/4)\}$, реализующей оператор U.

Известно: действительная часть матричного элемента $\langle 0^n|U|0^n\rangle$ больше $1/3$, либо меньше $-1/3$.

Выяснить: какой из случаев имеет место.

Теорема 4. *Задача оценки матричного элемента* BQP*-полна.*

Доказательство. Начнем с доказательства BQP-трудности этой задачи.

Пусть U — оператор, реализуемый схемой, для которой нужно оценить вероятность p наблюдения 1 в первом кубите.

Докажем, что для $V = U^\dagger \sigma_z[1] U$ выполняется равенство

$$\langle 0^n|V|0^n\rangle = 1 - 2p.$$

Пусть $U|0^n\rangle = |0\rangle \otimes |\psi_0\rangle + |1\rangle \otimes |\psi_1\rangle$.

Тогда

$$\langle 0^n|V|0^n\rangle = \langle 0^n|U^\dagger \sigma_z[1] U|0^n\rangle = \langle \psi_0|\psi_0\rangle - \langle \psi_1|\psi_1\rangle = 1 - 2p.$$

Осталось заметить, что $p < 1/3$ влечет $\langle 0^n|V|0^n\rangle > 1/3$, а $p > 2/3$ влечет $\langle 0^n|V|0^n\rangle < -1/3$.

Теперь докажем принадлежность задачи оценки матричного элемента классу BQP.

Итак, мы хотим оценить действительную часть матричного элемента оператора U. Построим ту же схему, которая использовалась для оценки фазы собственного числа.

Вероятность наблюдения 1 в первом кубите, если на вход схемы подан вектор $|0\rangle|0^n\rangle$, равна

$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \sum_k |c_k|^2 \frac{1 - \operatorname{Re}\lambda_k}{2}\ , \tag{1}$$

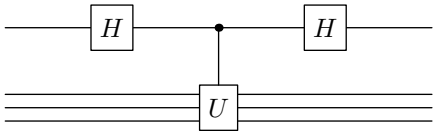


Рис. 2.

где λ_k — собственные числа оператора U, а c_k — коэффициенты разложения $|0^n\rangle$ по собственным векторам $|\psi_k\rangle$.

Выразим матричный элемент через те же величины:

$$\langle 0^n|U|0^n\rangle = \sum_{j,k} c_j^* c_k \lambda_k \langle\psi_j|\psi_k\rangle = \sum_k |c_k|^2 \lambda_k. \tag{2}$$

Здесь мы использовали, что собственные вектора унитарного оператора ортогональны.

Сравнивая (1) и (2), получаем

$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \frac{1}{2}(1 - \mathrm{Re}(\langle 0^n|U|0^n\rangle)).$$

Поскольку в задаче оценки матричного элемента заранее объявлено, что модуль матричного элемента больше $1/3$, мы видим, что вероятность 1 при одном знаке матричного элемента больше $2/3$, а при другом — меньше $1/3$. □

Приведем без доказательства еще два примера BQP-полных задач, формулировки которых не содержат явных отсылок к квантовым схемам.

Для формулировки задачи Нилла – Лафламма нам потребуется понятие *знаковой весовой функции*:

$$S(A, B, x, y) = \sum_{b: Ab=0} (-1)^{b^T Bb} x^{|b|} y^{n-|b|},$$

где A, B — матрицы над полем $\mathbb{F}_2$, b — вектор, $|b|$ — количество единиц (вес Хэмминга).

Задача Нилла – Лафламма формулируется так.

Дано: A — матрица с единицами на диагонали, k, ℓ — числа.

Известно: $|S(A, lwtr(A), k, \ell)| \geqslant (k^2 + \ell^2)^{n/2}/2$. Здесь $lwtr(A)$ получается из A заменой всех элементов на и над главной диагональю на нули.

Найти: знак $S(A, lwtr(A), k, \ell)$.

Теорема 5 (Knill, Laflamme [49], 1999). *Задача Нилла – Лафламма полна в классе* BQP.

В доказательстве этой теоремы используется подходящий универсальный базис для квантовых схем. В этом базисе задача об оценке матричного элемента пересказывается в виде оценки знаковой весовой функции указанного вида.

Для формулировки следующей задачи нам будет нужна некоторая версия понятия разреженной матрицы. А именно, под разреженной матрицей мы будем понимать матрицу, элементы которой заданы схемой вычисления, при этом известно, что количество ненулевых элементов в каждой строке и каждом столбце ограничено полиномом от логарифма размера матрицы.

В задаче Воцяна – Янцинга речь идет об оценке диагонального элемента степени разреженной матрицы. Точная ее формулировка такова.

Дано: A — матрица размера N, заданная схемой вычисления, числа b, $m = \mathrm{polylog}(N)$, j, $\varepsilon = 1/\mathrm{polylog}(N)$, g.

Известно: A — разреженная, симметричная; $\|A\| \leqslant b$; $|(A^m)_{jj} - g| \geqslant \varepsilon b^m$.
Найти: знак разности $(A^m)_{jj} - g$.

Теорема 6 (Janzing, Wocjan [45], 2007). *Задача оценки диагонального элемента полна в классе* BQP.

Доказательство BQP-трудности этой задачи основано на идее, которая нами не обсуждалась: связь между схемами и собственными числами локальных эрмитовых операторов (гамильтонианов).

9.4. Другие модели квантового вычисления

Помимо стандартной модели вычисления, которая подробно обсуждалась выше, есть еще несколько моделей. Те из них, которые не противоречат известным физическим законам, оказываются эквивалетными стандартной модели.

Приведем несколько примеров.

Частичные измерения. Эта модель предложена Нильсеном [52] в 2001 году. Феннер и Жанг ее усовершенствовали [40]. Получилась такая модель универсального квантового вычисления.

- Приготовление кубитов в состоянии $|0\rangle$.
- Конечный набор измерительных приборов.
- Измерения производятся над не более чем тремя кубитами с двумя возможными исходами, т. е.
    1. Пространство (не более чем трех) кубитов разбивается в ортогональную сумму двух подпространств (определяется выбором прибора измерения).
    2. Измерение состоит в том, что выбирается проекция на одно из подпространств с вероятностью, равной квадрату длине проекции вектора состояния на это подпространство.
    3. Проекция нормируется.
- Квантовая память (хранение многокубитных состояний, возникающих в процессе измерений).

Адиабатическое квантовое вычисление. Эта модель предложена Farhi, Goldstone, Gutmann, Sipser, 2000 [39] как алгоритм решения SAT. При точных оценках оказалось, что алгоритм требует экспоненциального времени.

Эквивалентность стандартной модели доказана Aharonov, van Dam, Kempe, Landau, Lloyd, Regev, 2005 [22].

- Гамильтониан — эрмитов оператор. k-Локальный гамильтониан — сумма операторов, действующих на k кубитах, $k = O(1)$.
- Основное состояние гамильтониана — собственный вектор, отвечающий наименьшему собственному числу.
- Вычисление задается двумя локальными гамильтонианами
    1. начальным гамильтонианом H_0, основное состояние которого предполагается разложимым (легко изготовить);
    2. конечным гамильтонианом H_1, измерение основного состояния которого дает ответ.

- Время работы пропорционально полиному от обратной точности $1/\varepsilon$, $\|H_1 - H_0\|$ и $\min \Delta(s)$, где $\Delta(s)$ — зазор между наименьшим и следующим по величине собственным числом гамильтониана $(1-s)H_0 + sH_1$.

Такая оценка времени работы связана с основной идеей: начиная с основного состояния исходного гамильтониана, менять его медленно, чтобы сохранять основное состояние, и прийти к конечному гемальтониану. В некотором смысле это квантовая версия метода отжига (simulated annealing) — известной эвристики, используемой при решении трудных оптимизационных задач.

Топологическая квантовая теория поля. В некоторых квантовых системах нет естественной структуры кубитов, но пространство состояний имеет большую размерность. Таково положение дел в топологической квантовой теории поля. Ограничимся кратким перечнем результатов в этом направлении и ссылками.

Унитарный топологический модулярный функтор: функтор из категории (поверхности, диффеоморфизмы) в категорию (унитарные пространства, унитарные операторы).

Freedman, Kitaev, Wang, 2000 [41] показали, что значения функтора можно эффективно приближать квантовыми схемами (вводя подходящим образом кубитовую структуру).

Freedman, Larsen, Wang, 2000 [42] нашли функтор, который универсален для квантового вычисления. Это означает, что квантовые схемы могут быть представлены как значения функтора от некоторого (явно заданного в некотором смысле) диффеоморфизма.

Aharonov, Jones, Landau, 2005 [23] построили алгоритм приближенного (в аддитивном смысле) вычисления полинома Джонса (NP-трудная задача для точного вычисления).

9.5. О соотношении класса BQP и классических классов сложности

Прежде всего дадим «верхнюю оценку» на класс BQP.

Теорема 7. $\mathrm{BQP} \subseteq \mathrm{PP}$.

Опишем план доказательства этой теоремы, опуская технические детали.

Матричный элемент оператора U, задаваемого схемой, выражается в виде суммы слагаемых, каждое из которых эффективно вычисляется. Поэтому можно сослаться на теорему 3 о принадлежности классу PP вычислении знака экспоненциальной суммы.

Действительно, если $U = U_\ell \dots U_1$, то

$$\langle 0^n|U|0^n\rangle = \sum_{x_0,\dots,x_\ell} \prod_{j=1}^{\ell} \langle x_{j-1}|U_j|x_j\rangle,$$

где $x_0 = x_\ell = 0^n$, $x_k \in \{0,1\}^n$.

Как видим, достаточно эффективной вычислимости матричных элементов базисных операторов.

Наиболее прост случай базиса Ши $\{\text{c-NOT}, F\}$. Все матричные элементы в этом случае можно представить рациональными числами со знаменателем 5.

Задача 1. Завершите доказательство включения $\mathrm{BQP} \subseteq \mathrm{PP}$, используя схемы в стандартном базисе $\{\text{c-NOT}, H, K(\pi/4)\}$.

Указание: докажите, что за полиномиальное от n время можно построить рациональное число $q = r/10^k$, r, k — целые, для которого $|q - \sqrt{2}| < 2^{-n}$.

Теперь можно уточнить диаграмму включений классов, добавив в нее класс эффективных квантовых алгоритмов BQP.

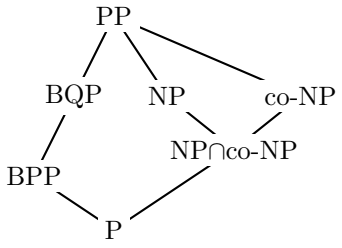


Рис. 3. Предполагается, что других включений в диаграмме нет (и тем самым, все включения строгие)

Доказательство того, что любое из указанных на диаграмме 3 включений строгое, стало бы самым выдающимся достижением в теории сложности за все время ее существования.

Пока удается доказывать лишь варианты таких включений, относящиеся к *релятивизованным* классам.

Если A — задача (язык, класс), а $\mathcal{C}$ — класс сложности, то $\mathcal{C}^A$ — класс сложности, состоящий из тех задач, которые решаются алгоритмами из класса $\mathcal{C}$, которые имеют доступ к оракулу (подпрограмме), решающему задачу A.

Здесь мы отчасти возвращаемся к нашей первой теме — алгоритмам «черного ящика», которые идейно близки к оракульным вычислениям.

Заметим прежде всего, что непосредственно из определений следует, что для любого оракула диаграмма включений сохраняется:

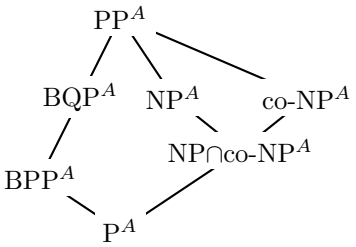


Рис. 4.

Однако теперь можно утверждать и строгость включений для некоторых оракулов.

Теорема 8 (Benett, Bernstein, Brassard, Vazirani [31], 1996). *Существует такой оракул A, что* $\mathrm{BQP}^A \not\supseteq (\mathrm{NP} \cap \text{co-NP})^A$.

Доказательство этой теоремы о разделении основано на нижних оценках для задачи поиска Гровера. Оракул не строится явно, а неравенство доказывается для «почти всех» оракулов некоторого специального вида.

Еще раньше была доказана другая теорема о разделении релятивизованных классов.

Теорема 9 (Bernstein, Vazirani [32], 1993). *Существует такой оракул A, что* $\mathrm{BQP}^A \supset \mathrm{BPP}^A$.

Лекция 10. Моделирование квантовых схем классическими средствами. О реализации квантового компьютера

10.1. Моделирование квантового ресурса классическими средствами

Естественный вопрос, который возникает в связи с квантовыми алгоритмами, — возможность моделировать их работу классическими средствами.

Для вероятностных алгоритмов, скажем, существуют мощные методы *дерандомизации*, которые основаны на том, что источник случайных битов заменяется на программу, которая порождает *псевдослучайные биты*.

С квантовым ресурсом ситуация сложнее. Вероятностные распределения, порождаемые квантовыми схемами, устроены сложнее, чем последовательность независимых случайных битов. Не вполне ясно, каким мог бы быть общий механизм *псевдоквантового вычисления*. Наиболее прямолинейное обобщение идей дерандомизации состояло бы в построении в пространстве полиномиальной размерности экспоненциально большого семейства векторов, имитирующего вычислительный базис на n кубитах, и замены линейных унитарных преобразований на подходящие преобразования пространства малой размерности. Первая часть такой программы легко выполняется, используя хорошие коды. Однако никаких общих идей по моделированию унитарных операторов, реализуемых квантовыми схемами, пока не придумано.

Мы рассмотрим два примера, когда моделирование квантовых схем классическими средствами возможно за полиномиальное (от размера схемы) время. Оба случая не дают универсального квантового вычисления, хотя в некотором смысле очень близки к этому.

Первый пример — схемы в *симплектическом базисе* $\{\text{c-NOT}, H, K(\pi/2)\}$. От универсального базиса он отличается тем, что фазовый сдвиг «грубее» (в универсальном базисе используется $\pi/4$).

Симплектический базис не является универсальным. Как мы увидим, операторы из этого базиса образуют конечную подгруппу $\mathbf{U}((\mathbb{C}^2)^{\otimes n})$.

Теорема 1 (Готтесман – Нилл, 1997). *Вероятность наблюдения* 1 *в первом кубите при измерении состояния, которое получено применением схемы в симплектическом базисе, вычисляется классическим алгоритмом за полиномиальное время.*

В доказательстве теоремы 9.4 о BQP-полноте задачи об оценке матричного элемента помимо схемы, реализующей оператор U, использовались только операторы из симплектического базиса. Поэтому получаем такое следствие.

Следствие. *Матричные элементы оператора, заданного схемой в симплектическом базисе, можно вычислять классическим алгоритмом за полиномиальное время.*

Замечание 1 (о сцепленности как квантовом ресурсе). *Сцепленность* как явление означает существование в тензорном произведении пространств неразложимых в вычислительном базисе состояний, например ЭПР пара

$$|\text{ЭПР}\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle.$$

В *квантовой теории информации* вводятся разнообразные меры сцепленности и количество сцепленности (скажем, ЭПР-пар) является важным информационным ресурсом.

Теорема Готтесмана – Нилла показывает сомнительность пользы от понятия сцепленности для квантовых вычислений. Сцепленность порождается операторами из симплектического базиса

$$|\text{ЭПР}\rangle = \text{c-NOT}[1,2]H[1]|00\rangle,$$

а тот моделируется классически.

Перед доказательством теоремы Готтесмана – Нилла нужно вернуться к матрицам Паули. Введем для них другую систему обозначений, включающую также и единичный оператор:

$$\sigma_{00} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}; \qquad \sigma_{01} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \sigma_z;$$
$$\sigma_{10} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \sigma_x; \qquad \sigma_{11} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \sigma_y. \tag{1}$$

Напомним, что в пространстве операторов определено произведение Фробениуса по формуле

$$\langle A, B\rangle = \mathrm{Tr}(A^\dagger B).$$

Для (вещественного) пространства эрмитовых операторов это скалярное произведение.

Несложными вычислениями проверяется следующее утверждение.

Утверждение 1. *Матрицы Паули $\frac{1}{\sqrt{2}}\sigma_{\alpha\beta}$ образуют ортонормированный базис в пространстве эрмитовых операторов на $\mathbb{C}^2$.*

Учитывая мультипликативность следа относительно тензорного произведения, получаем отсюда

Следствие. *Тензорные произведения*

$$\sigma(f) = \sigma(\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_n, \beta_n) \stackrel{def}{=} \sigma_{\alpha_1,\beta_1} \otimes \sigma_{\alpha_2,\beta_2} \otimes \cdots \otimes \sigma_{\alpha_n,\beta_n} \tag{2}$$

образуют ортогональный базис в пространстве эрмитовых операторов на $(\mathbb{C}^2)^{\otimes n}$. Здесь $f \in \mathbb{F}_2^{2n}$.

Будем называть этот базис *симплектическим*, а операторы вида (2) — σ-операторами.

Удобство обозначений (1) проявляется при записи произведений σ-операторов. Для произведений матриц Паули имеем следующие формулы

$$\sigma(\alpha,\beta)^2 = I; \quad \sigma(\alpha_1,\beta_1)\sigma(\alpha_2,\beta_2) = i^{\tilde\omega(\alpha_1,\beta_1;\alpha_2,\beta_2)}\sigma(\alpha_1\oplus\alpha_2, \beta_1\oplus\beta_2).$$

С точностью до фазового множителя индексы складываются по модулю 2. То же самое, разумеется, справедливо и для общих σ-операторов. Фазовый множитель устроен довольно хитро. Можно проверить, что он задается формулой

$$\tilde\omega(\alpha_1,\beta_1;\alpha_2,\beta_2) = \alpha_1^2\beta_1^2 + (\alpha_2)^2(\beta_2)^2 - (\alpha_1+\alpha_2)^2(\beta_1+\beta_2)^2 + 2\alpha_2\beta_1 \bmod 4.$$

для произведения матриц Паули, а для общих σ-операторов нужно просуммировать фазовые множители из каждого тензорного множителя по модулю 4.

Лемма 1. *Действие симплектических операторов $X \mapsto UXU^\dagger$ на пространстве эрмитовых операторов сохраняет базис σ-операторов (с точностью до фазового множителя i^a).*

Доказательство. Операторы из симплектического базиса действуют на одном или двух кубитах. Достаточно проверить утверждение леммы для действия одного оператора.

Для одного кубита выполняются следующие соотношения (повороты сфер Блоха, которые уже нам встречались)

$$H\sigma_x H^\dagger = \sigma_z, \qquad H\sigma_y H^\dagger = -\sigma_y, \qquad H\sigma_z H^\dagger = \sigma_x;$$
$$K\sigma_x K^\dagger = \sigma_y, \qquad K\sigma_y K^\dagger = -\sigma_x, \qquad K\sigma_z K^\dagger = \sigma_z.$$

Для c-NOT[1, 2] (без ограничения общности фиксируем кубиты, на которые действует c-NOT) имеют место соотношения (проверьте!)

$$U\sigma_z[1]U^\dagger = \sigma_z[1], \qquad U\sigma_x[1]U^\dagger = \sigma_x[1]\sigma_x[2],$$
$$U\sigma_z[2]U^\dagger = \sigma_z[1]\sigma_z[2], \qquad U\sigma_x[2]U^\dagger = \sigma_x[2].$$

Поскольку σ-операторы замкнуты относительно умножения с точностью до фазового множителя i^a, получаем отсюда утверждение леммы. □

Следствие. *Симплектические схемы порождают конечную группу унитарных матриц.*

Действительно, симплектические операторы имеют определители вида i^a. Ядром действия группы $\mathbf{U}$ на пространстве эрмитовых операторов $X \mapsto UXU^\dagger$ являются скалярные операторы λI. Так как имеется не более $4^N N!$ (здесь $N = 2^{2n}$) операторов, которые переставляют σ-операторы и, быть может, подкручивают их на множитель i^a, то порядок группы симплектических операторов не больше $4^{N+1}N!$.

Лемма 2. *Если оператор U задается симплектической схемой, то $U^\dagger\sigma(f)U = i^{U_a(f)}\sigma(\tilde{U}f)$, причем U_a и $\tilde{U}f$ вычисляются за полиномиальное время классическим алгоритмом.*

Доказательство. Из формул умножения операторов Паули и формул действия для образующих симплектической группы следует, что $\tilde{U}$ — линейное преобразование на $\mathbb{F}^{2n}$.

Фазовый множитель вычисляется эффективно по тем же формулам. □

Теперь перепишем удобным для доказательства теоремы Готтесмана – Нилла образом формулу для вероятности наблюдения 1 в первом кубите при измерении состояния $|\psi\rangle = \sum_x c_x|x\rangle$

$$\mathbf{Pr}(|\psi\rangle, 1) = \sum_{x:x_1=1} |c_x|^2 = \langle\psi|\Pi_1[1]|\psi\rangle = \langle\psi|\frac{1}{2}(\sigma_{00} - \sigma_{11}) \otimes I^{\otimes n-1}|\psi\rangle =$$
$$= \frac{1}{2}\left(\langle\psi|\sigma(f_0)|\psi\rangle - \langle\psi|\sigma(f_1)|\psi\rangle\right) = \frac{1}{2}\big(1 - \langle\psi|\sigma(f_1)|\psi\rangle\big) \quad (3)$$

Здесь $\sigma(f_1) = \sigma_z[1]$ в обозначениях, которые мы использовали ранее.

Доказательство теоремы Готтесмана – Нилла. Пусть оператор U задается симплектической схемой. Из формулы (3) для вероятности наблюдения получаем

$$\mathbf{Pr}(U|0^n\rangle, 1) = \frac{1}{2}\left(1 - \langle 0^n|U^\dagger\sigma(f_1)U|0^n\rangle\right) = \frac{1}{2}\big(1 - i^{U_a(f_1)}\$bra0^n\sigma(\tilde{U}f_1)|0^n\rangle\big).$$

Оператор $\sigma(\tilde{U}f_1) = \bigotimes_k \sigma(\alpha_k, \beta_k)$ — разложимый и его компоненты, а также фазовый множитель $U_a(f_1)$ вычисляются за полиномиальное время в силу леммы 2. Поэтому

$$\mathbf{Pr}(U|0^n\rangle, 1) = \frac{1}{2} - \frac{i^{U_a(f_1)}}{2}\prod_k \langle 0|\sigma(\alpha_k, \beta_k)|0\rangle$$

вычисляется за полиномиальное время. □

Это доказательство теоремы Готтесмана – Нилла предложено Джоза [46], который заметил также, что групповая структура в этом рассуждении не существенна. Нужны два ингредиента:

- множество $\mathcal{S}_n$ эрмитовых операторов такое, что $\langle 0^n|S|0^n\rangle$ вычисляется эффективно для $S \in \mathcal{S}_n$, нужно также, чтобы это множество содержало $\sigma(f_1)$;
- множество унитарных операторов $\mathcal{K}_n$, сохраняющих $\mathcal{S}_n$, т. е. $K^\dagger S K \in \mathcal{S}_n$ для $K \in \mathcal{K}_n$, $S \in \mathcal{S}_n$.

В случае теоремы Готтесмана – Нилла $\mathcal{S}_n$ — это группа σ-операторов с фазовыми подкрутками, $\mathcal{K}_n$ — группа симплектических операторов.

Вторым примером квантовых схем, эффективно моделируемых классическими алгоритмами, являются так называемые *плоские схемы Вэлианта*.

Определение. Двухкубитовый элемент Вэлианта (matchgate) задается матрицей

$$G(A,B) = \begin{pmatrix} p & 0 & 0 & q \\ 0 & w & x & 0 \\ 0 & y & z & 0 \\ r & 0 & 0 & s \end{pmatrix}, \quad \text{где } A = \begin{pmatrix} p & q \\ r & s \end{pmatrix}, \quad B = \begin{pmatrix} w & x \\ y & z \end{pmatrix}$$

две унитарные матрицы с одинаковым детерминантом.

Определение. *Плоская схема Вэлианта* составлена из элементов Вэлианта. Кубиты схемы упорядочены и элементы Вэлианта применяются только к соседним кубитам. Предполагается также, что матричные элементы в схеме Вэлианта эффективно вычислимы.

Теорема 2 (Valiant [64], 2001). *Существует алгоритм, который за полиномиальное время вычисляет вероятности исходов для состояния, которое получается из $|0^n\rangle$ применением плоской схемы Вэлианта.*

Теорема 3 (Josza, Miyake [47], 2008). *Схемы, в которых элементы Вэлианта применяются к кубитам на расстоянии не больше 2, универсальны для квантового вычисления.*

Мы не приводим доказательств этих теорем, ограничимся лишь несколькими замечаниями о теореме Вэлианта. На данный момент известно три совершенно разных доказательства этой теоремы.

Сам Вэлиант доказывал ее сводимостью к вычислению пфаффиана плоского графа. Для него есть полиномиальный алгоритм вычисления, который, в частности, позволяет находить за полиномиальное время количество совершенных паросочетаний в плоском графе. На данный момент существует большое количество статей, в которых изучаются схемы из элементов Вэлианта.

Второе доказательство получено Джоза [46]. Оно исходит из сделанного выше замечания. В случае плоских схем Вэлианта возможен выбор выбор подходящего семейства $\mathcal{S}_n$ (семейство $\mathcal{K}_n$ — это, конечно, плоские схемы Вэлианта):

$$\mathcal{S}_n = \mathbb{R}(\sigma_z^{\otimes k} \otimes \sigma_x \otimes I^{\otimes n-k-1}, \sigma_z^{\otimes k} \otimes \sigma_y \otimes I^{\otimes n-k-1}), \quad 1 \leqslant k \leqslant n.$$

Наконец, есть третье доказательство, придуманное Терхал и Дивинченцо [63]. Оно к тому же показывает физический смысл плоских схем Вэлианта. Оказывается, эти схемы описывают физические системы «невзаимодействующих фермионов». Более подробное обсуждение этой модели лежит за пределами данных лекций, заинтересованный читатель может найти подробности в литературе [48, 63].

10.2. Квантовые вычисления, устойчивые к ошибкам

Если говорить о возможности создания устройств, реализующих квантовые алгоритмы (для краткости, квантовые компьютеры), то начинать нужно с обсуждения проблемы шума.

На заре развития вычислительной техники фон Нейман дал убедительное объяснение, почему дискретные системы более пригодны к длинным и сложным вычислениям, чем аналоговые. В последних из-за неустранимого шума постепенно накапливаются неточности. В дискретной системе ошибка (скажем, изменение бита) происходит довольно редко и влияние этой ошибки может быть очень сильно снижено применением корректирующих кодов.

Квантовые системы, описываемые стандартной моделью, — аналоговые. Их нетривиальные возможности связаны с тонкой игрой амплитудами. Конечно, как мы видели, накопление ошибок при применении неточной реализации унитарных операторов происходит линейно.

Но это означает, что для реализации схемы с приемлемой точностью необходима реализация операторов с точностью, обратно пропорциональная размеру схемы. Уже это плохо. Любая реалистическая модель будет давать конечную величину ошибки.

Однако есть и еще одна проблема помимо неточностей в реализации элементов схем, в реальных системах имеется *шум*: случайные сбои в работе. Оказывается, в квантовом случае произвольные сбои устроены сложнее, чем просто искажение параметров унитарного преобразования.

Поэтому квантовые алгоритмы подвержены шуму в большей степени. Настолько большей, что многие физики в начале 90-х годов прошлого столетия сомневались в возможности реализации квантовых схем ненадежными элементами, подверженными шуму.

Проблема квантовых вычислений, устойчивых к ошибкам, тщательно изучалась. Результаты этого изучения скорее положительные. Ниже я попробую дать представление об этих результатах.[1)]

Анализ квантовых вычислений, подверженных ошибкам, производится при следующем предположении, оправданном сегодняшним уровнем физических знаний. Считаем, что реализации элемента квантовой схемы возникает ошибка, причем вероятность ошибки мала (но не сколь угодно мала, скажем, 1%) и каждая ошибка влияет только на небольшое количество кубитов.

Вопрос тогда ставится следующим образом: можно ли, используя такие элементы строить сколь угодно большие схемы, которые дают ответ с небольшой вероятностью ошибки?

Прежде всего нужно понять, какова должна быть модель квантовой ошибки. Выше было сказано, что шевеления унитарного оператора недостаточно, чтобы описать все возможные ошибки.

Оказывается, весь наш формализм стандартной квантовой модели непригоден доля описания ошибок общего вида. Дело в том, что унитарная эволюция возможна лишь для замкнутых систем. Эволюция открытых систем, взаимодействующих со своим окружением, описывается более сложной моделью. А шум, разумеется, и есть (неконтролируемое) взаимодействие с окружающей средой.

[1)] Большая глава с популярным рассказом о вычислениях, устойчивых к ошибкам, есть в книге Нильсена, Чанга [11]. Сравнительно короткий обзор Прескилла [54] может также оказаться полезным введением в тему. Подробное изложение исправления ошибок и симплектических кодов (stabilizer codes) можно найти в диссертации Готтесмана [44], в работе Аароновой и Бен-Ора [21] и в диссертации Алифериса [24].

Пример. Возьмем ЭПР пару

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle.$$

Предположим, что ошибка состоит в том, что потерялся второй кубит. В каком состоянии оказывается первый кубит?

Ответ: Первый кубит с вероятностью $1/2$ находится в состоянии $|0\rangle$ и с вероятностью $1/2$ в состоянии $|1\rangle$.

Упражнение 1. Докажите, что не существует *чистого* (т. е. элемента унитарного пространства, как это определялось выше) квантового состояния, которое давало бы такие вероятности наблюдения исходов.

Для описания состояний, возникающих после физических преобразований общего вида, нужны более общие, *смешанные*, состояния. Они представляют собой вероятностные распределения на множестве чистых состояний. Поскольку это множество бесконечно, то и вероятностных распределений на нем много. Однако природа устроена в этом месте несколько проще, чем может показаться на первый взгляд.

Действительно, два вероятностных распределения на чистых состояниях *неразличимы*, если они дают одинаковые распределения исходов при одинаковом наблюдении. В этом случае нет никаких оснований считать, что они задают разные состояния системы. Приходим к такому определению.

Определение. Класс неразличимых распределений на чистых состояниях называется *смешанным состоянием*.

Теорема 4. *Смешанное состояние однозначно описывается* оператором плотности ρ, *удовлетворящим условиям:*

$$\rho^\dagger = \rho; \quad \langle\psi|\rho|\psi\rangle \geqslant 0; \quad \operatorname{Tr}\rho = 1.$$

Вероятность наблюдения исхода x в состоянии ρ равна

$$\mathbf{Pr}(\rho, x) = \operatorname{Tr}(\rho\Pi_x) = \langle x|\rho|x\rangle. \tag{4}$$

Доказательство. Сопоставим чистым состояниям операторы плотности ранга 1, т. е. проекторы на одномерное подпространство $\Pi_\psi = |\psi\rangle\langle\psi|$.

Тогда вероятностному распределению будет соответствовать оператор

$$\rho = \sum_j p_j \Pi_{\psi_j}, \quad p_j \geqslant 0, \ \sum_j p_j = 1 \tag{5}$$

(для простоты ограничиваемся распределениями с конечными носителями)

Вероятности исходов в чистом состоянии запишем как

$$\mathbf{Pr}(|\psi\rangle, x) = |c_x|^2 = \langle x|\psi\rangle\langle\psi|x\rangle = \langle x|\Pi_\psi|x\rangle = \operatorname{Tr}(\Pi_\psi\Pi_x).$$

В последнем равенстве использовано циклическое свойство следа

$$\operatorname{Tr}(ABC) = \sum_{j,k,l} a_{jk}b_{kl}c_{lj} = \operatorname{Tr}(BCA).$$

Для смешанного состояния получим

$$\mathbf{Pr}(\sum_j p_j\Pi_{\psi_j}, x) = \sum_j p_j \operatorname{Tr}(\Pi_\psi\Pi_x) = \operatorname{Tr}(\rho\Pi_x).$$

Таким образом, вероятности зависят только от ρ, а не от выбора вероятностной смеси (5). □

Смешанные состояния, в частности, описывают состояние части квантовой системы, находящейся в неразложимом состоянии.

Пусть мы имеем ρ — смешанное в общем случае состояние составной системы AB, где возможные результаты наблюдения системы A — это $\{a_1, \ldots, a_n\}$, а системы B — это $\{b_1, \ldots, b_m\}$.

Как определить состояние подсистемы A?

Для этого нужно посчитать вероятность наблюдения результата a_i в составной системе и выразить ее с помощью подходящего оператора плотности:

$$\mathbf{Pr}(\rho, a_j) = \sum_k \mathbf{Pr}(\rho, a_j b_k) = \mathbf{Pr}(\operatorname{Tr}_B \rho, a_j). \tag{6}$$

Здесь $\operatorname{Tr}_B \rho$ обозначает искомый ответ, который называется *частичным следом оператора*.

Определение. Частичный след — это линейное отображение операторов на пространстве $\mathcal{A} \otimes \mathcal{B}$ в операторы на пространстве $\mathcal{A}$, которое на *разложимых* операторах задается формулой

$$\operatorname{Tr}_B(X \otimes Y) = X \operatorname{Tr} Y, \tag{7}$$

а на остальные операторы продолжается по линейности.

Упражнение 2. Докажите, что частичный след определен корректно, т. е. его значение не зависит от представления оператора в виде суммы разложимых).

Корректность формулы (6) вытекает из следующего наблюдения.

Упражнение 3. Проверьте, что для разложимых чистых состояний $\rho = \rho_1 \otimes \rho_2$

$$\mathbf{Pr}(\rho, a_j) = \mathbf{Pr}(\operatorname{Tr}_B \rho, a_j).$$

Правило из физики: возможные преобразования квантовой системы. Все физически реализуемые преобразования операторов плотности являются композициями преобразований следующих трех видов:

- Действие унитарного оператора на операторах плотности:

  $$\rho \mapsto U \rho U^\dagger$$

  (Для чистых состояний это согласовано с предыдущим определением унитарного преобразования, так как $|\psi\rangle\langle\psi| \mapsto U|\psi\rangle\langle\psi|U^\dagger = |U\psi\rangle\langle U\psi|$.)

- Добавление в систему новой части в известном состоянии γ:

  $$\rho \mapsto \rho \otimes \gamma;$$

- Отбрасывание части составной системы (взятие частичного следа):

  $$\rho \mapsto \operatorname{Tr}_B \rho.$$

Теперь можно описать модель квантовой ошибки.

Определение. *Элемент U с ошибкой ε на r кубитах* реализует преобразование операторов плотности

$$\rho \mapsto (1 - \varepsilon) U \rho U^\dagger + \varepsilon E(\rho), \tag{8}$$

где E — преобразование матриц плотности, которое является суммой преобразований, нетривиально действующих на r кубитах.

Почему преобразование (8) удовлетворяет данному выше описанию? Конечно, можно явно найти требуемую композицию основных преобразований. Однако есть и более общее рассуждение.

Теорема 5 (представление операторной суммой). *Любое физически реализуемое преобразование матриц плотности представляется в виде*

$$\rho \mapsto \sum_m A_m \rho A_m^\dagger, \quad \textit{где } \sum_m A_m^\dagger A_m = I.$$

Следствие. *Множество физически реализуемых преобразований операторов плотности выпукло.*

Имея модель квантовой ошибки, можно точно сформулировать вопрос о построении схем, устойчивых к ошибкам. За недостатком времени мы пропустим эти определения и ограничимся несколькими иллюстрациями. Они относятся к частному случаю вычисления: реализации тождественного оператора.

Это обобщение проблемы корректирующих кодов на квантовый случай. Пусть произошла ошибка. Мы хотим ее исправить (ослабленный вариант: если не исправить, то хотя бы заметить, что случилась ошибка).

Задача исправления ошибок в квантовом случае труднее, и это можно показать на очень простом примере. В классическом случае есть код повторений, когда значение бита повторяется несколько раз. Тогда можно обнаружить и исправить чуть меньше половины ошибок.

В квантовом случае это не так. Уже ошибка на одном кубите, причем унитарная, приводит к совпадению кодовых векторов. Рассмотрим два вектора из кодового пространства:

$$|\psi_1\rangle = |0^n\rangle + |1^n\rangle; \quad |\psi_2\rangle = |0^n\rangle - |1^n\rangle.$$

Пусть ошибка имеет вид $E = \sigma_z[j]$. Тогда $E|\psi_2\rangle = |\psi_1\rangle$. После такой ошибки у нас не остается шансов различить эти два кодовых вектора.

Необходимым (и, как выясняется, достаточным) условием исправления ошибок является ортогональность образов ортогональных кодовых векторов после воздействия ошибки. Действительно, любую пару ортогональных векторов (единичной длины) можно унитарным преобразованием перевести в любую другую пару. Эти соображения лежат в основе определения квантового корректирующего кода.

Определение. *Квантовый код*, исправляющий ошибки из множества $\mathcal{E}$ — это подпространство V пространства $(\mathbb{C}^2)^{\otimes n}$ такое, что

$$\forall\, |\xi_1\rangle, |\xi_2\rangle \in V \ \forall\, X, Y \in \mathcal{E} \ (\langle\xi_2|\xi_1\rangle = 0) \Rightarrow \left(\langle\xi_2|Y^\dagger X|\xi_1\rangle = 0\right). \tag{9}$$

Определение. В этом случае $\mathcal{E}$ состоит из линейных отображений, которые являются суммами r-локальных (действующих только на r кубитах).

В этом случае условие (9) упрощается до

$$\forall\, |\xi_1\rangle, |\xi_2\rangle \in V \ \forall\, f \in \mathbb{F}_2^{2n} (\|f\| \leqslant 2r) \ (\langle\xi_2|\xi_1\rangle = 0) \Rightarrow (\langle\xi_2|\sigma(f)|\xi_1\rangle = 0). \tag{10}$$

Здесь $\|f\|$ — количество ненулевых пар (α_k, β_k) в наборе f. Параметр $2r$ называется *кодовым расстоянием*.

Несмотря на обескураживающую неудачу с кодом повторений, оказывается, что квантовые корректирующие коды можно строить весьма похожим на классические линейные коды образом. При это возникают так называемые симплектические коды (stabilizer codes).

Идея конструкции симплектического кода состоит в том, чтобы использовать в качестве кода выбираются собственные подпространства систем попарно коммутирующих σ-операторов.

Задача 4. (Правила коммутирования σ-операторов) Проверьте, что

$$\sigma(f)\sigma(g) = (-1)^{\omega(f,g)}\sigma(g)\sigma(f),$$

где

$$\omega(f,g) = \sum_k \Big(\alpha_k(f)\beta_k(g) - \alpha_k(g)\beta_k(f)\Big) \bmod 2.$$

Таким образом, симплектический код задается изотропным подпространством F пространства $\mathbb{F}_2^n$, снабженного симплектической формой ω.

В этих линейно-алгебраических терминах можно также выразить кодовое расстояние симплектического кода.

Теорема 6. *Кодовое расстояние кода, задаваемого подпространством F, равно*

$$\min(\|f\| : f \in F^\perp \setminus F), \quad \text{где } F^\perp = \{g : \forall f \in F\ \omega(f,g) = 0\}.$$

Эта теорема позволяет применять конструкции классических корректирующих кодов (с дополнительными условиями) для построения квантовых кодов.

Конструкции симплектических кодов позволяют перенести на случай квантовых кодов все основные результаты классической теории корректирующих кодов: коды, исправляющие любое заданное количество ошибок; асимптотически хорошие квантовые коды; каскадные конструкции; процедуры декодирования. Особо отметим, что последние реализуются в симплектическом базисе, что играет важную роль в общей теории квантовых вычислений, устойчивых к ошибкам.

Опишем общие идеи, лежащие в основе такого вычисления.

- Кубиты кодируются корректирующим кодом, после чего применяется каскадная конструкция для уменьшения вероятности ошибки (глубина каскада — логарифм от размера схемы, которую нужно реализовать).
- Действия над закодированными кубитами выполняются без раскодирования: строятся преобразования кодовых слов, отвечающие действию унитарного оператора на исходных кубитах.

Реализация этих идей весьма непроста, а сложность анализа полученных конструкций при наиболее реалистических предположениях (как сделано выше) очень велика. Тем не менее, удалось получить следующий принципиальный результат.

Теорема 7 (пороговая теорема). *Существует такое число p_0, что любая квантовая схема размера ℓ может быть реализована с вероятностью ошибки не более ε схемой размера $O(\mathrm{poly}(\log \ell/\varepsilon)\ell)$ из неточных элементов, вероятность ошибки каждого из которых не превосходит p_0.*

Пороговая теорема позволяет надеяться на возможность реализации квантового компьютера. Если бы она была неверна, то квантовые алгоритмы вошли бы в коллекцию моделей, интересных и важных с теоретической точки зрения, но не имеющих шансов на практическую реализацию.

К этой теореме необходимо сделать несколько замечаний.

- Пороговая теорема доказана лишь при очень малых значениях порога: $p_0 \sim 10^{-5}$. Один из критических вопросов в области квантовых вычислений: определить точную величину порога.
- Неизбежность высокого параллелизма. Схемы из неточных элементов, в которых одновременно преобразуются лишь $O(1)$ кубитов, при любой ненулевой величине порога ошибки моделируются вероятностными алгоритмами [20].

Таким образом, пороговая теорема не только показывает принципиальную возможность создания квантового компьютера, но и недвусмысленно говорит о значительной трудности этой задачи

10.3. О возможности создания квантового компьютера

На данный момент исследования в области создания квантовых компьютеров не достигли решающего успеха. Поэтому существуют (и выражаются разными людьми) различные точки зрения на саму возможность создания квантового компьютера. Встречаются взгляды в очень широком диапазоне. Мы приведем их в виде «шкалы оптимизма».

Квантовые компьютеры

1. никогда не будут созданы, потому что их вычислительные возможности не стоят усилий по их созданию.
2. никогда не будут созданы из-за непреодолимых технологических трудностей.
3. невозможны, поскольку противоречат (еще не открытым) законам природы.
4. будут построены, но будут реализовывать более слабую, чем стандартная, модель квантового вычисления.
5. на основе известных законов квантовой механики возможны и будут построены в обозримом будущем.
6. могут быть гораздо мощнее, чем нам сейчас кажется из-за неполноты знаний законов квантового мира. Когда они будут созданы, квантовые компьютеры смогут решать алгоритмически неразрешимые задачи.

Последняя, на наш взгляд, самая оптимистическая точка зрения принадлежит Р. Пенроузу, который активно ее пропагандирует в своих популярных книгах (см., например, [12]).

Возможность 3 наиболее интригующая для физиков. Им, конечно, очень хочется строить квантовые компьютеры прежде всего для того, чтобы проверить законы квантовой механики в совершенно экзотических условиях, которые сами по себе не возникают в природе.

Для построения квантовых компьютеров потенциально пригодны любые квантовые системы. Перечислим наиболее популярные подходы:[2)]

- Фотоны.
- Ионные ловушки.
- ЯМР.
- Ядерные спины в полупроводниках.

Кроме того, физики обсуждают возможность построения квантовых компьютеров на основе и более экзотических физических систем. Например, на основе топологических квазичастиц — анионов. В случае анионов проблема шума значительно снижается, так как локальное воздействие на них тривиально. К сожалению, анионов нужного сорта пока открыть не удалось.

Под конец обсудим возможность 4 из шкалы оптимизма. Проиллюстрируем ее на примере оптики (т. е. реализации квантового вычисления фотонами).

[2)] Подробнее см. книгу Нильсена и Чанга [11].

Чтобы реализовать универсальное квантовое вычисление, нужны нелинейные оптические элементы. Это плохо из-за большого затухания в таких элементах. Если ограничиться только линейными элементами (светоделители и фазовращатели — почти то же самое, что полупрозрачные пластинки и поляризационные фильтры), то получается интересная ограниченная модель квантового вычисления, которую еще можно называть моделью невзаимодействующих бозонов.

В недавней работе Ааронсона и Архипова возможности линейной оптики описаны следующей моделью, не требующей знания физики.

Состояния вычислительной системы — многочлены из $\mathbb{C}[x_1, \ldots, x_m]$ степени n, здесь $m \geqslant n$.

Начальное состояние — многочлен $f_0 = x_1 x_2 \ldots x_n$.

Вычисление: применение унитарного преобразования U к переменным:

$$f_{\text{final}}(x) = f_0(Ux).$$

Измерение в состоянии

$$\sum_{S=(s_1,\ldots,s_n)} \alpha_S x_1^{s_1} \ldots x_m^{s_m}$$

дает S с вероятностью $\mathbf{Pr}[S] = |\alpha_S|^2 s_1! \ldots s_m!$.

Ааронсон и Архипов связали возможности линейной оптики с ограниченным вариантом задачи вычисления перманента (эта задача полна в некотором классе сложности, который мы не вводили, и уж во всяком случае NP-трудна). Если бы квантовые вычисления позволяли считать перманент, интерес к созданию квантовых компьютеров увеличился бы на два порядка (как минимум).

Однако линейной оптике отвечает более слабая задача оценки квадрата перманента (ОКП). В этой задаче нужно оценить квадрат модуля перманента $|\operatorname{perm}(X)|^2$ с аддитивной точностью $\pm\varepsilon n!$ и вероятностью ошибки δ за время $\operatorname{poly}(n, 1/\varepsilon, 1/\delta)$ в предположении, что входом является матрица, выбранная из вероятностного распределения $N(0,1)^{n\times n}$, при котором значения всех матричных элементов независимы и каждый распределен по Гауссу (плотность вероятности $\exp(-t^2/2)$).

Теорема 8 (Aaronson, Arkhipov [19], 2010). *Если существует классический алгоритм, который приближает распределение, порождаемое схемой в модели невзаимодействующих бозонов, то задача оценки квадрата перманента принадлежит классу* $\mathrm{BPP}^{\mathrm{NP}}$ *(вероятностные вычисления с оракулами из* NP*).*

Чем замечательна эта теорема? Авторы рассчитывают, что из принадлежности ОКП классу $\mathrm{BPP}^{\mathrm{NP}}$ удастся получить коллапс полиномиальной иерархии (что почти столь же сомнительно как $\mathrm{P} = \mathrm{NP}$).

Успех этой программы даст первое по-настоящему сильное свидетельство в пользу трудности задач, которые решаются квантовыми устройствами.

Поэтому оптимисты могут присоединится к реализации программы Ааронсона – Архипова. А пессимисты могут попробовать построить вероятностный алгоритм решения задачи оценки квадрата перманента. Нельзя сказать, что это совсем уж невероятно: для других ослаблений задачи вычисления перманента вероятностные алгоритмы были построены.

Предупреждение. Список литературы не претендует на полноту. Возможно, есть и более удачные ссылки вместо тех, которые здесь приводятся. Также весьма вероятно, что пропущены ссылки на некоторые важные и/или пионерские работы.

Литература

[1] Алон Н., Спенсер Дж. Вероятностный метод. М.: Бином, 2007.

[2] Бугаенко В.О. Уравнения Пелля. Библиотека «Математическое просвещение», вып. 13. М.: МЦНМО, 2001.

[3] Винберг Э.Б. Курс алгебры. М.: Факториал, 1999.

[4] Виноградов И. М. Основы теории чисел. Изд. 8-е, испр. М.: Наука, 1972.

[5] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

[6] Китаев А.Ю. Квантовые вычисления: алгоритмы и исправление ошибок. УМН, т. 52, №6, 1997. С. 53–112.

[7] Китаев А., Шень А., Вялый М. Классические и квантовые вычисления. М: МЦНМО, ЧеРо, 1999. Англ. пер. (испр. и доп.) A.Yu.Kitaev, A.H. Shen, M.N. Vyalyi. Classical and quantum computation. AMS, 2002.

[8] Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. М.: МЦНМО, 2000.

[9] Кострикин А. И., Манин Ю. И. Линейная алгебра и геометрия. М.: Наука, 1986.

[10] Марков А.А. Об одном вопросе Д.И. Менделеева. Изв. СПб Акад. наук, т. 62, 1889. С. 1–24.

[11] Нильсен М., Чанг И. Квантовые вычисления и квантовая информация. М.: Мир, 2006. Оригинал: Nielsen M.A., Chuang I.L. quantum computation and information. Cambridge Univ. Press, 2000.

[12] Пенроуз Р. Тени разума: в поисках науки о сознании. Москва–Ижевск: Институт компьютерных исследований, 2005.

[13] Полиа Г., Сеге Г. Задачи и теоремы из анализа. Т. 2. Изд. 3е. М.: Наука, 1978.

[14] Разборов А.А. О квантовой коммуникационной сложности симметричных предикатов. Изв. РАН (серия матем.), т. 67, №1, 2003. С. 159–176, arXiv:quant-ph/0204025.

[15] Р. Фейнман, Р. Лейтон, М. Сэндс. Фейнмановские лекции по физике. Тт. 8 – 9. Квантовая механика. М.: Мир, 1978.

[16] Хинчин А.Я. Цепные дроби. М.: Наука, 1978.

[17] Холево А.С. Квантовые системы, каналы и информация. М.: МЦНМО, 2010.

[18] S. Aaronson and A. Ambainis. Quantum search of spatial regions. Theory of Computing, 1(4):47–79, 2005, arXiv:quant-ph/0303041v3.

[19] Aaronson S., Arkhipov A. The Computational Complexity of Linear Optics, 2010. http://www.scottaaronson.com/papers/optics.pdf The Computational Complexity of Linear Optics

[20] Aharonov D., Ben-Or M. Polynomial simulations of decohered quantum computers, 1996. arXiv:quant-ph/9611029.

[21] Aharonov D., Ben-Or M. Fault-tolerant computation with constant error rate, 1999. arXiv:quant-ph/9906129.

[22] Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, Oded Regev. Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation, 2004. arXiv:quant-ph/0405098v2

[23] Dorit Aharonov, Vaughan Jones, Zeph Landau. A Polynomial Quantum Algorithm for Approximating the Jones Polynomial. Algorithmica, 2009. Vol. 55. P. 395–421.

[24] Aliferis P. Level reduction and quantum threshold theorem, 2007. arXiv:0703230v1.

[25] Andris Ambainis. Quantum lower bounds by quantum arguments. J. Comput. Syst. Sci., 64:750–767, 2002, arXiv:quant-ph/0002066.

[26] Andris Ambainis. Polynomial degree vs. quantum query complexity. J. Comput. Syst. Sci., 72(2):220–238, 2006, arXiv:quant-ph/0305028.

[27] Arora S., Barak B. Computational complexity: a modern approach. Cambridge Univ. Press, 2009.

[28] Babai L., Kimmel P.G. Randomized Simultaneous Messages: Solution of a Problem of Yao in Communication Complexity. IEEE Conference on Computational Complexity, 1997. P. 239–246, см. также http://www.cs.uchicago.edu/research/publications/techreports/TR-2001-07

[29] Bach E., Shallit J. Algorithmic number theory. MIT Press, 1997.

[30] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, Ronald de Wolf. Quantum Lower Bounds by Polynomials. FOCS 1998: 352–361, arXiv:quant-ph/9802049.

[31] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. SIAM Journal on Computing, 26(5):1510–1523, 1997, arXiv:quant-ph/9701001

[32] E. Bernstein, U. Vazirani. Quantum complexity theory. SIAM Journal on Computing, 26(5):1411-1473, 1997. http://www.cs.berkeley.edu/~vazirani/bv.ps

[33] G. Brassard, A. Broadbent, A. Tapp. Quantum Pseudo-Telepathy, 2004. arXiv:quant-ph/0407221v3.

[34] M. Boyer, G. Brassard, P. Hoyer, A. Tapp, Tight bounds on quantum searching, Proceedings, PhysComp 1996, arXiv:quant-ph/9605034.

[35] H. Buhrman, R. Cleve, J. Watrous, R. de Wolf. Quantum fingerprinting, 2001. arXiv:quant-ph/0102001v1.

[36] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In Proc. 30th ACM Symp. on Theory of Computing (STOC), pages 63–68, 1998.

[37] Buhrman H., Wolf, de, R. Complexity measures and decision tree complexity: a survey. J. Theor. Comput. Sci., vol. 288, 2002. P. 21–43.

[38] Chernoff H. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. Annals of Mathematical Statistics, vol 23 (4), 1952. P. 493–507.

[39] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, arXiv:quant-ph/0001106, 2000.

[40] Fenner S. A., Zhang Y. Universal quantum compuation with two- and three-qubit projective measurements, 2001. arXiv:quant-ph/0111077v2.

[41] Michael H. Freedman, Alexei Kitaev, Zhenghan Wang. Simulation of topological field theories by quantum computers, 2000. arXiv:quant-ph/0001071v3.

[42] Michael Freedman, Michael Larsen, Zhenghan Wang. A modular functor which is universal for quantum computation, 2000. arXiv:quant0ph/0001108v2.

[43] Dmitry Gavinsky, Julia Kempe, Ronald de Wolf. Quantum Communication Cannot Simulate a Public Coin, 2004. arXiv:quant-ph/0411051v1.

[44] Gottesman D. Stabilizer Codes and Quantum Error Correction, 1997. arXiv:quant-ph/9705052v1.

[45] Janzing D., Wocjan P. A Simple PromiseBQP-complete Matrix Problem. Theory of computing. Volume 3, 2007. P. 61–79. http://theoryofcomputing.org

[46] Josza R. Embedding classical into quantum computation, 2008. arXiv:quant-ph/0812.4511v1.

[47] Jozsa R. and Miyake A., Matchgates and classical simulation of quantum circuits, 2008, arXiv:quant-ph/0804.4050.

[48] Knill E. Fermionic linear optics and matchgates, 2001. arXiv:quant-ph/0108033v2.

[49] Knill E., Laflamme R. Quantum Computation and Quadratically Signed Weight Enumerators, 1999. arXiv:quant-ph/9909094v1.

[50] Knuth D. Combinatorial matrices, 1991.
http://www-cs-faculty.stanford.edu/~knuth/preprints.html#unpub

[51] Kuperberg G. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem, 2003, arXiv:quant-ph/0302112v2.

[52] Nielsen M.A. Universal quantum computation using only projective measurement, quantum memory, and preparation of the $|0\rangle$ state, 2001. arXiv:quant-ph/0108020v1

[53] Nisan N., Szegedy M. On the degree of Boolean functions as real polynomials. Computational Complexity, vol.4 (4), 1994. P. 301–313.

[54] Preskill J. Reliable quantum computers, 1997. arXiv:quant-pt/9705031v3.

[55] Razborov A.A. On the distributional complexity of disjointness. Theoretical Computer Science, vol. 106, 1992. P. 385–390.

[56] Reichardt B.W. Span programs and quantum query algorithms. ECCC, 2010. TR110.

[57] Sherstov A.A. Lower bounds in communication complexity and learning theory via analytic methods. Dissertation. Univ of Texas, 2009. http://www.cs.utexas.edu/~sherstov/publications/pdf/Sherstov_Dissertation.pdf

[58] Shi Y. Both Toffoli and Controlled-NOT need little help to do universal quantum computation, 2002. arXiv:quant-ph/0205115v2.

[59] Shor P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. vol. 26 (5), 1997. P. 1484–1509, arXiv:quant-ph/9508027v2.

[60] Sipser M. Introduction to the Theory of Computation. Boston: PWS Publishing Company, 1997.

[61] Solovay R. Unpublished manuscipt, 1995.

[62] Robert Špalek and Mario Szegedy. All quantum adversary methods are equivalent. Theory of Computing, 2(1):1–18, 2006, arXiv:quant-ph/0409116.

[63] Terhal B. M., DiVincenzo D. P. Classical simulation of noninteracting-fermion quantum circuits, 2001, arXiv:quant-ph/0108010.

[64] Valiant L. Quantum circuits that can be simulated classically in polynomial time. SIAM J. on Computing, vol 31, no 4, 2002. P. 1229–1254.

В этом разделе собраны задачи к курсу. Число в квадратных скобках в начале строки с номером задачи (или какого-то ее пункта) указывает количество баллов, которым оценивается полное решение данной задачи.

Для успешной сдачи экзамена нужно набрать не менее 33 баллов. Для хорошей оценки — не менее 50 баллов, для отличной — не менее 66 баллов.

[2] **Задача 1.** Докажите, что тензорное произведение унитарных операторов унитарно.

[1] **Задача 2.** Найдите такое унитарное преобразование U, которое различает $|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ и $|\xi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ в том смысле, что измерение состояний $U|\xi\rangle$ и $(U \otimes I)|\psi\rangle$ дает разные распределения вероятностей в первом бите.

[2] **Задача 3.** Постройте квантовый алгоритм с запросами к «черному ящику», вычисляющий дизъюнкцию за $O(1)$ запросов в предположении, что $|x| = n/4$ или $|x| = 0$.

[10] **Задача 4.** Рассмотрим алгоритмы решения задачи поиска Гровера (существует ровно один положительный ответ), которые не используют дополнительной памяти и работают в пространстве $\mathbb{C}^m$, ортонормированный базис которого образуют возможные ответы на задачу поиска.

Докажите, что если такой алгоритм решает задачу поиска с вероятностью ошибки, не превосходящей ε, то он делает не менее $\Omega(\sqrt{m})$ запросов.

Предупреждение: нижняя оценка для вычисления дизъюнкции тут не работает.

[1] **Задача 5.** Найдите степень многочлена p, точно представляющего дизъюнкцию n переменных.

[3] **Задача 6.** Пусть $f(x)$ — многочлен такой, что $|f(0)| < 1/3$, $|f(1) - 1| < 1/3$, а $-1/3 < f(k) < 4/3$ при $2 \leqslant k \leqslant n$. Докажите, что $\deg f \geqslant \sqrt{n/6}$.

Указание. Используйте неравенство Маркова: для многочлена f от одной переменной из $|f(x)| \leqslant 1$ на отрезке $[-1; 1]$ следует $|f'(x)| \leqslant (\deg f)^2$ на отрезке $[-1; 1]$.

[10] **Задача 7** (задача ван Дама). Докажите, что для любой булевой функции $Q_{1/3}(f) \leqslant n/2 + \sqrt{n}$, где n — количество переменных.

Указание: используйте преобразование, аналогичное решению задачи Дойча – Джоза для того, чтобы с небольшой вероятностью ошибки получить в результате измерения всю таблицу значений булевой функции.

[3] **Задача 8.** Докажите, что общий протокол квантовой коммуникации можно моделировать протоколом с передачей лишь по одному выделенному кубиту, причем длина этого протокола ограничена удвоенной длиной исходного протокола общего вида.

Указание: перестановка кубитов (тензорных сомножителей) — унитарный оператор.

[3] **Задача 9.** Докажите, используя хорошие коды, что $R_\varepsilon(\mathrm{EQ}) = O(\log n)$.

[3] **Задача 10.** Докажите, что $R_\varepsilon^{\mathrm{SMP,\ pub}}(\mathrm{EQ}) = O(\log n)$.

Задача 11.

[4] (i) Докажите, что не существует вероятностной стратегии в игре Мермина, которая гарантировала бы (с вероятностью 1) победу игроков, даже если они используют общий генератор случайности.

[2] (ii) Постройте стратегию с общим генератором случайности, при которой вероятность выигрыша $3/4$.

[1] **Задача 12.** Напишите в вычислительном базисе матрицу оператора c-NOT$[1,3]$, который действует на трех кубитах. Оператор c-NOT — это оператор обратимого копирования: c-NOT: $|x,y\rangle \mapsto |x, x\oplus y\rangle$.

[4] **Задача 13.** Докажите, что в базисе из перестановок двух битов не все отображения реализуемы в расширенном смысле.

[4] **Задача 14.** Докажите, что без использования вспомогательных битов невозможно реализовать отображение

$$\mathrm{c}^{(n)}\text{-NOT}\colon (x_1,\dots,x_n,y)\mapsto (x_1,\dots,x_n,y\oplus x_1x_2\dots x_n)$$

в базисе из перестановок n битов.

[3] **Задача 15.** Постройте схему полиномиального размера в базисе $\{\text{cc-NOT}, \text{c-NOT}, H, K(\pi/2)\}$, которая реализует оператор Гровера

$$R_\psi = 2|\psi\rangle\langle\psi| - I, \quad |\psi\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle.$$

Дайте как можно более точную оценку размера схемы.

[2] **Задача 16.** Докажите, что любой поворот трехмерного пространства является композицией двух поворотов на угол π.

[3] **Задача 17.** Докажите, что если оператор U_1 приближается в расширенном смысле оператором $\tilde U_1$ с точностью δ_1, а оператор U_2 приближается в расширенном смысле оператором $\tilde U_2$ с точностью δ_2, то U_1^{-1} приближается в расширенном смысле $\tilde U_1^{-1}$ с точностью δ_1, а U_1U_2 приближается в расширенном смысле $\tilde U_1\tilde U_2$ с точностью $\delta_1+\delta_2$.

[2] **Задача 18.** Угол между прямыми a_0, a_1, проходящими через центр сферы, равен ϑ. Докажите, что $O(1/\vartheta)$ поворотов вокруг осей a_0, a_1 достаточно, чтобы точку p сферы перевести в точку q.

[6] **Задача 19.** Докажите, что если $\cos\alpha = -\cos^2\frac{\pi}{8}$, то α несоизмерим с π.

Указание: проверьте по индукции, что $\cos(n\alpha)$ имеет вид $\frac{p_n}{2^{k_n}} + q_n\sqrt{2}$, где p_n — нечетные целые.

[3] **Задача 20.** Пусть квантовый алгоритм Q вычисляет $f(x)$ с вероятностью ошибки $\varepsilon < 1/2$.

Алгоритм Q'_s работает следующим образом:

1. s раз независимо повторить алгоритм Q;
2. выдать результатом то значение y, которое встретилось чаще всего.

Докажите, что Q'_s вычисляет $f(x)$ с вероятностью ошибки $< (2\sqrt{\varepsilon(1-\varepsilon)})^s$.

[10] **Задача 21.** Преобразованием Фурье назовем оператор

$$F_q|x\rangle = \frac{1}{\sqrt{q}}\sum_{y=0}^{q-1}\exp\left(2\pi i\frac{xy}{q}\right)|y\rangle,$$

Постройте квантовую схему размера $\mathrm{poly}(n\log(1/\delta))$, реализующую преобразование Фурье на группе $\mathbb{Z}_k$ при любом $k\le 2^n$ с точностью δ.

[2] **Задача 22.** Докажите, что PP = co-PP.

[4] **Задача 23.** Докажите PP-трудность задачи вычисления матричного элемента квантовой схемы в стандартном базисе
$\{\text{c-NOT}, H, K(\pi/4)\}$ с аддитивной точностью 2^{-m}. (Число m задается в унарной системе счисления.)

[6] **Задача 24.** Используя квантовые схемы в стандартном базисе
$\{\text{c-NOT}, H, K(\pi/4)\}$, Докажите PP-трудность задачи вычисления весовой функции линейного кода в точке $\exp(\pi i/4)$ с аддитивной точностью 2^{-m}. (Число m задается в унарной системе счисления.)

Кодом называется подпространство C пространства $\mathbb{F}_2^n$, весовая функция

$$w(t) = \sum_{x \in C} t^{\|x\|}.$$

[3] **Задача 25.** Представьте измерение в классическом базисе как композицию элементарных преобразований операторов плотности (унитарное действие, добавление системы в чистом состоянии, взятие частичного следа).

[3] **Задача 26.** Представьте преобразование операторов плотности

$$\rho \mapsto (1-\varepsilon) U \rho U^\dagger + \varepsilon E$$

как композицию элементарных преобразований операторов плотности. Здесь U — произвольный оператор на одном кубите, E — преобразование потери когерентности, которое заменяет все внедиагональные элементы матрицы на нули.