Квантовые вычисления
Глава 5

Сысоев Сергей Сергеевич

Санкт-Петербургский государственный университет
Математико-механический факультет
http://se.math.spbu.ru/

1 Введение

Материал этой главы, последней во всем курсе, посвящен двум важным результатам, позволяющим сформировать некоторое представление о возможностях и границах квантовых вычислений.

В предыдущей главе мы познакомились с алгоритмом Шора, позволяющим при решении задачи разложения числа на множители на квантовом устройстве получить экспоненциальное ускорение по сравнению с классическим компьютером. Для всех ли задач возможно подобное ускорение в квантовых вычислениях? Какое ускорение можно в принципе всегда гарантированно получить при переходе к квантовой модели вычислений?

Эти, и многие другие вопросы пока остаются открытыми, и, может быть, кто-то из читателей найдет для них исчерпывающие ответы. Пока же наша цель – познакомиться с тем, что к сегодняшнему дню известно на этот счет.

Мы начнем с очень важного и общего результата, полученного в 1996 году Ловом Гровером. Алгоритм Гровера предназначен для поиска интересующей нас записи в неотсортированной базе данных. Это, например, как если бы нам потребовалось найти фамилию абонента в телефонном справочнике, если мы знаем телефон. Телефонные справочники отсортированы по фамилиям, и в них нет сортировки по телефонам. Поэтому поиск фамилии по номеру телефона – весьма трудоемкая задача. В самом неудачном случае нам придется просмотреть весь справочник, прежде чем мы найдем интересующую нас запись.

Поиск в неотсортированной базе – это просто еще одна формулировка задачи о поиске прообраза легко вычислимой функции в некоторой точке. Умение эффективно решать подобные задачи естественным образом влечет умение решать любую задачу из класса NP. Как мы уже упоминали в главе 1, класс NP – это такие задачи, для которых мы можем легко (эффективно) проверить решение. Эта проверка – всегда вычисление некоторой функции f, а поиск решения – это очень часто поиск обратной функции f^{-1}. Если кто-нибудь нам скажет, что знает фамилию абонента, которого мы ищем, и назовет ее, мы легко можем проверить эту догадку, найдя указанную фамилию в справочнике и сравнив телефон напротив нее с имеющимся у нас эталоном.

Или, например, в известной всем задаче коммивояжера необходимо найти в некотором графе гамильтонов путь, длина которого меньше заданного значения B. Имея конкретный путь – кандидат в решение задачи, мы можем легко вычислить его длину и проверить, подходит ли нам этот путь. Однако, имея лишь длину, мы обычно не можем легко подобрать путь, имеющий эту длину (то есть, вычислить обратную функцию).

Существование эффективного решения задачи коммивояжера – вопрос открытый, в том числе и для квантового компьютера. Но если мы представим функцию f в виде черного ящика, оракула, про внутреннее устройство которого мы ничего не знаем, то и в классическом, и в квантовом случае мы можем предъявить оптимальный алгоритм решения и проанализировать его сложность.

В классическом случае нам подходит любая стратегия перебора. Если мы ищем фамилию абонента в телефонном справочнике, мы можем просто просматривать всех абонентов и сравнивать их телефонные номера с эталонным. При емкости телефонного справочника N абонентов, мы получаем в среднем $O(N)$ операций сравнения.

Удивительно, но для квантовых вычислений существует более эффективный алгоритм. Это – алгоритм Гровера.

2 Алгоритм Гровера

Поскольку алгоритм Гровера представляет собой обобщенный, не зависящий от конкретной задачи поиск, функция f представляется в нем в виде черного ящика.

$$f : \{0,1\}^n \to \{0,1\}^n$$

В области определения f присутствует одна особая точка, на которой f принимает нужное нам значение:

$$\exists!\, \omega : f(\omega) = a$$

В приведенном выше примере с телефонным справочником ω – фамилия абонента, в то время как a – известный нам телефонный номер. Необходимо найти ω. Как видите, это – обобщенный вариант любой задачи из NP. Зная ответ – число ω, мы можем легко его проверить, вызвав оракул и посмотрев, получим ли мы значение a.

Определим функцию f_ω следующим образом .

$$f_\omega(x) = \delta_{x=\omega}$$

f_ω принимает значение "0" во всех точках, кроме ω, от которой она равна "1". Подобную функцию легко реализовать, имея оракул f.

Оракул

Вспомним, как действует определенный нами ранее квантовый оракул на состояние вида $\frac{1}{\sqrt{2}} |x\rangle (|0\rangle - |1\rangle)$:

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

Мы можем рассмотреть проекцию U_f на подпространство аргумента ($|x\rangle$):

$$U_\omega |x\rangle = (-1)^{f(x)} |x\rangle$$

Оператор U_ω действует как тождественный оператор на любой базисный вектор, кроме ω, поэтому мы можем записать его следующим образом:

$$U_\omega = I - 2 |\omega\rangle \langle\omega|$$

Действие этого оператора на любой вектор $|x\rangle$ определяется следующим образом:

$$U_\omega |x\rangle = |x\rangle - 2 |\omega\rangle \langle\omega|x\rangle$$

Для любого базисного вектора кроме ω скалярное произведение $\langle\omega|x\rangle$ дает 0, и мы имеем тождественный оператор:

$$\langle\omega|x\rangle = 0$$

$$U_\omega |x\rangle = |x\rangle - 2 |\omega\rangle \langle\omega|x\rangle = |x\rangle$$

Для вектора ω мы имеем

$$U_\omega |\omega\rangle = |\omega\rangle - 2 |\omega\rangle \langle\omega|\omega\rangle = |\omega\rangle - 2 |\omega\rangle = - |\omega\rangle$$

Начальное состояние – то, с которого начинается работа алгоритма Гровера, строится следующим образом:

$$|s\rangle = H |0\rangle^n = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle$$

$|s\rangle$ – это уже знакомая нам равновзвешенная сумма всех базисных векторов подпространства аргумента. Такие начальные состояния мы уже много раз готовили в предыдущих задачах. Определим еще вспомогательный оператор U_s:

$$U_s = 2 |s\rangle \langle s| - I$$

Итерация Гровера.
Алгоритм Гровера итеративен. Каждая его итерация определяется, как применение операторов U_ω и U_s к текущему состоянию системы:

$$R_{grov} = U_s U_\omega$$

Действие итерации Гровера на начальное состояние s изображено на (fig. 1).

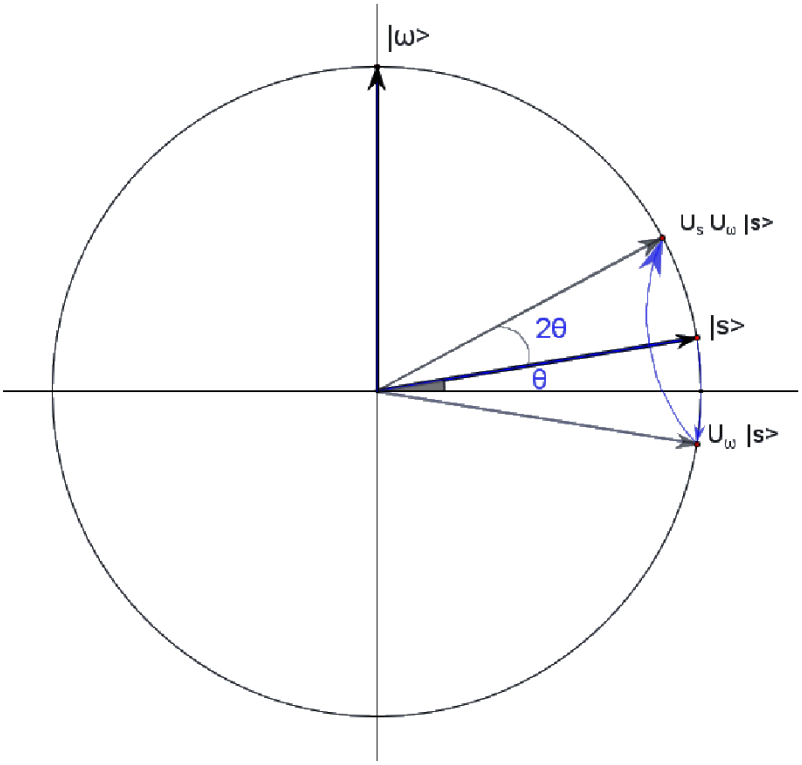

Fig. 1. Первая итерация Гровера.

На (fig. 1) вертикальная ось – это искомый вектор ω, а горизонтальная ось – это гиперпространство, образованное всеми остальными базисными векторами. Вектор $|s\rangle$ – сумма всех базисных векторов – нависает над горизонтальной гиперплоскостью под углом θ.

$$\sin\theta = |\langle s|\omega\rangle| = \frac{1}{2^{n/2}} \tag{1}$$

Чем больше размерность пространства, тем меньше угол θ, тем ниже находится вектор $|s\rangle$.

Оператор U_ω является отражением вектора системы относительно гиперплоскости, ортогональной ω. Применив его, мы получаем вектор $U_\omega|s\rangle$, в котором все базисные составляющие остались неизменны, и только составляющая $|\omega\rangle$ заменилась на $-|\omega\rangle$. Оператор U_s – это отражение вектора системы относительно вектора $|s\rangle$. После него мы получаем вектор $U_s U_\omega|s\rangle$.

Получается, что первая итерация Гровера повернула исходный вектор к искомому вектору $|\omega\rangle$ на угол 2θ. Теперь угол вектора системы с горизонтальной гиперплоскостью составляет 3θ.

На (fig. 2) изображено действие второй итерации Гровера, которая также поворачивает вектор системы на угол 2θ в направлении $|\omega\rangle$.

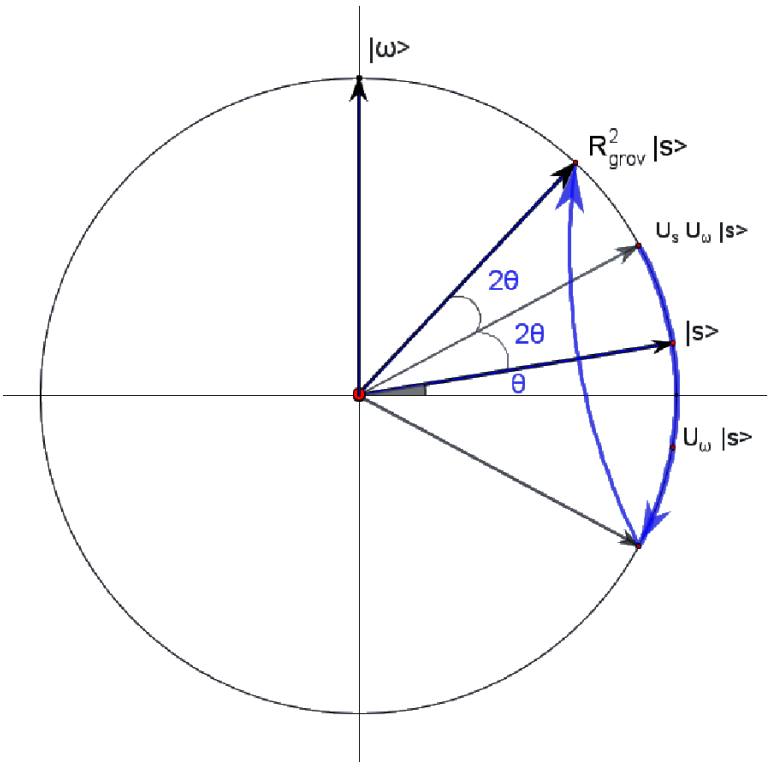

Fig. 2. Вторая итерация Гровера.

Нетрудно убедиться, что каждая следующая итерация Гровера будет делать с вектором системы тоже самое – приближать его к $|\omega\rangle$ на угол 2θ. После T итераций угол между вектором системы и гиперплоскостью будет равен $(\theta+2T\theta)$, и нам нужно подобрать такое количество итераций T, чтобы он стал близок к $\pi/2$. Это позволит при измерении состояния получить вектор $|\omega\rangle$ с вероятностью, близкой к единице.

$$\theta + 2T\theta = \pi/2 \iff T = \frac{\pi}{4\theta} - \frac{1}{2}$$

При больших значениях n угол θ становится достаточно мал и может быть приближенно заменен своим синусом (1):

$$T = \frac{\pi\sqrt{2^n}}{4} - \frac{1}{2} \approx \frac{\pi\sqrt{2^n}}{4}$$

Как видите, вместо классических 2^n итераций квантовому компьютеру требуется только $2^{n/2}$ обращений к оракулу. Это, конечно, не экспоненциальное ускорение, как в алгоритме Шора. Но и квадратичного ускорения во многих задачах может оказаться вполне достаточно.

Например, при $n = 2$ вместо 4-х обращений к оракулу нам необходимо только одно. А при $n = 8$ алгоритму Гровера потребуется около 12-ти итераций (вместо классических 256-ти).

Пример реализации алгоритма Гровера для двух кубитов на симуляторе квантового компьютера представлен на (fig. 3), (fig. 4), (fig. 5).

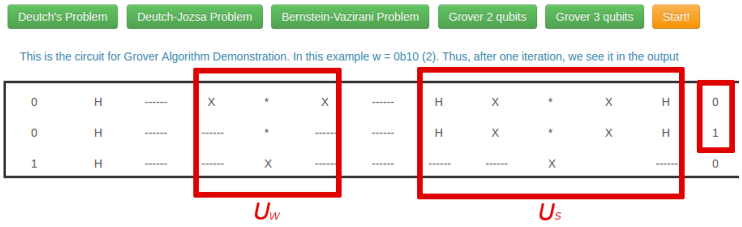

Fig. 3. Алгоритм Гровера на двух кубитах. $\omega = 0\text{x}10 = 2$.

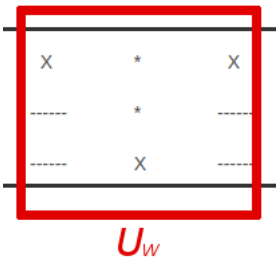

Fig. 4. Оператор U_ω. $\omega = 0\text{x}10 = 2$.

Оператор U_ω (fig. 4) действует (выполняется контролируемый двумя кубитами аргумента оператор CNOT), только если верхний (младший) бит аргумента равен 0, а средний (старший) бит аргумента равен 1, что соответствует двоичному представлению числа 2. Гейт X на первом (младшем) кубите аргумента повторяется дважды, первый раз для того, чтобы выполнилось условие действия CNOT для значения 0x10, а второй – чтобы вернуть младший кубит аргумента к его первоначальному значению (нейтрализовать действие первого X).

Оператор U_s (fig. 5) действует, как отражение относительно вектора $|s\rangle$, что соответствует отражению относительно вектора $|+\rangle^n$ в базисе Адамара.

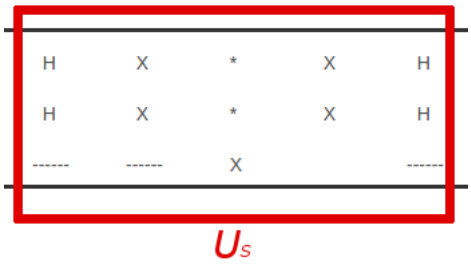

Fig. 5. Оператор U_s.

Применив преобразование Адамара (слева и справа), мы получаем возможность реализовать U_s как отражение относительно вектора $|0\rangle^n$.

Давайте еще раз посмотрим на схему нашего алгоритма.
1. Почему мы именно таким образом выбрали наше начальное состояние – вектор $|s\rangle$? Дело в том, что каждая итерация Гровера приближает вектор системы к искомому вектору на угол 2θ. Чем больше угол θ, тем быстрее мы решим задачу. Мы исходим из предположения, что нам ничего неизвестно про вектор $|\omega\rangle$, и, соответственно, единственный вектор, угол которого с $|\omega\rangle$ мы знаем – это вектор, равноудаленный от всех других векторов.
2. Мы предположили, что искомая точка только одна. Если их несколько, скажем, k, то по вертикальной оси тоже возникает гиперплоскость из всех этих векторов, а $\sin\theta$ становится равен $\frac{\sqrt{k}}{\sqrt{2^n}}$. Это потребует перерасчета количества итераций. Таким образом, нам необходимо знать точно, сколько решений у поставленной нам задачи. Если мы этого не знаем, мы рискуем неверно рассчитать количество итераций и "проскочить" искомые вектора. В алгоритме Гровера, добавляя итерации сверх оптимального количества, мы отдаляемся от решения!
3. И, наконец, почему мы именно так определили оператор U_s? Нам в пространстве аргумента необходима точка, от которой мы могли бы отражаться в направлении $|\omega\rangle$. Чем ближе эта точка к $|\omega\rangle$, тем меньше нам потребуется отражений. Если бы эта точка лежала ровно на середине пути, то мы попали бы в $|\omega\rangle$ всего за одну итерацию! Но где же взять эту точку? Если мы ничего не знаем об ω, то опять же, наилучшей точкой для таких отражений для нас будет точка, равноудаленная от всех векторов пространства.

Гипотетический квантовый компьютер с архитектурой фон Неймана.
Давайте немного пофантазируем. Когда мы только запускаем алгоритм Гровера, мы ничего не знаем о состоянии $|\omega\rangle$. Представим, что мы выполнили ровно половину итераций Гровера (fig. 6 справа). Вектор системы – $|s_6\rangle$ "знает" о состоянии $|\omega\rangle$ намного больше, чем мы. Он находится ровно на

половине пути. Если бы мы могли отразиться от этого вектора, мы бы пришли в состояние $|\omega\rangle$ всего за одну итерацию!

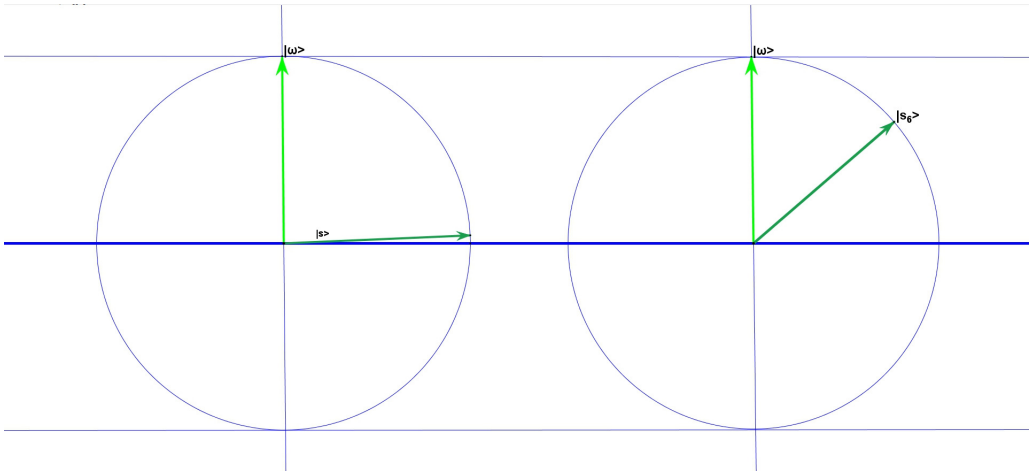

Fig. 6. Алгоритм Гровера на двух квантовых системах.

Проблема заключается в том, что нам нечего отразить относительно этого вектора, поскольку он сам является вектором системы. Вот если бы мы могли использовать его как оператор в некоторой другой системе, в которой итерации Гровера еще не начались... Вот тогда бы мы действительно пришли в $|\omega\rangle$ за одну итерацию (fig. 7), (fig. 8). Более того, само состояние $|s_6\rangle$ могло бы быть получено из состояния $|s_3\rangle$ всего за одну итерацию совершенно таким же способом (fig. 9), (fig. 10). И, вместо $\sqrt{2^n}$ итераций мы получили бы только n.

Для подобного усовершенствования нам потребуется научиться использовать состояния (квантовые данные) как операторы – квантовые гейты. В классических вычислениях это называется архитектурой фон Неймана. К сожалению, в настоящий момент физическая осуществимость этой архитектуры для квантовых вычислений находится под большим вопросом.

Подобная операция не входит в математическую модель, которую мы с вами разобрали во второй главе. Поэтому прекращаем фантазировать и переходим к доказательству оптимальности алгоритма Гровера в рамках той модели, которая у нас есть.

3 Оптимальность алгоритма Гровера.

Мы выяснили, что для квантовых вычислений существует алгоритм, позволяющий находить прообраз неизвестной функции f за время $O(2^{n/2})$, где n – разрядность аргумента f. Это впечатляющее открытие может вселить

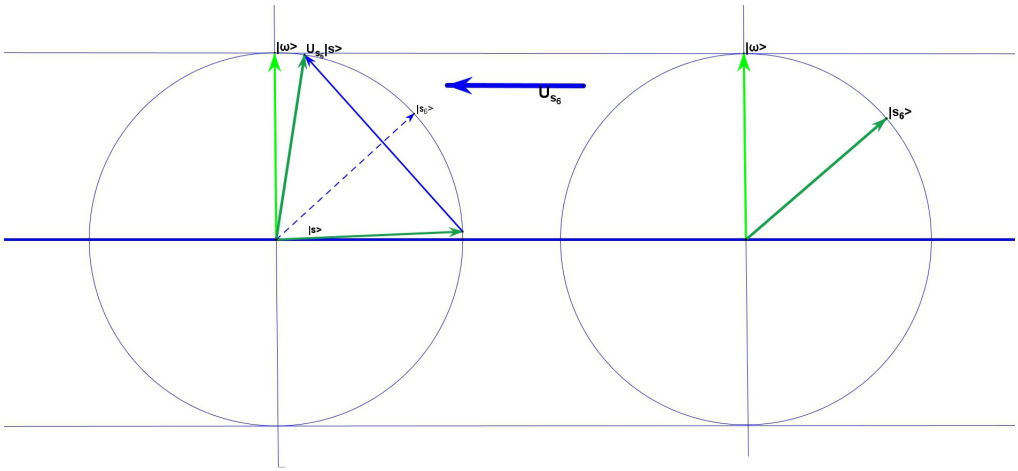

Fig. 7. Получение оператора U_{s_6}.

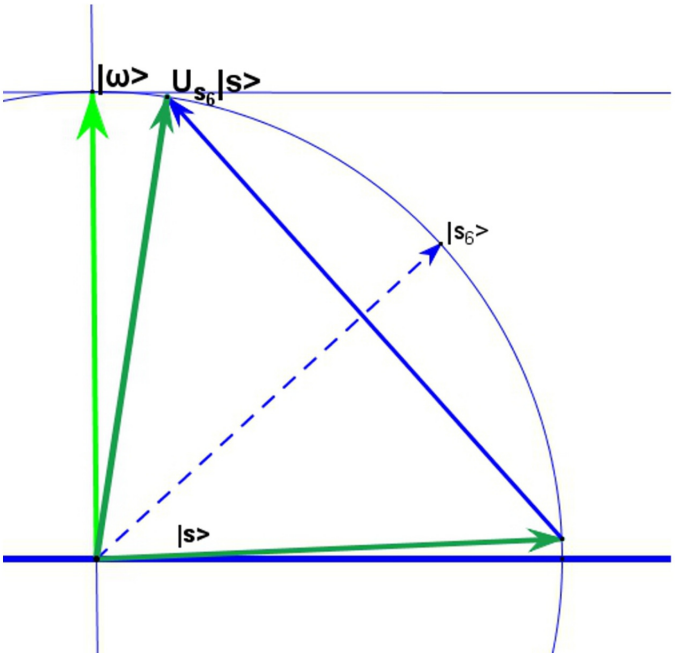

Fig. 8. Действие оператора U_{s_6}.

в нас надежду, подогреваемую успехом Питера Шора, на то, что можно придумать алгоритм еще лучше, который давал бы нам результат еще более быстрый, может быть даже экспоненциально. Надежде этой, тем не менее,

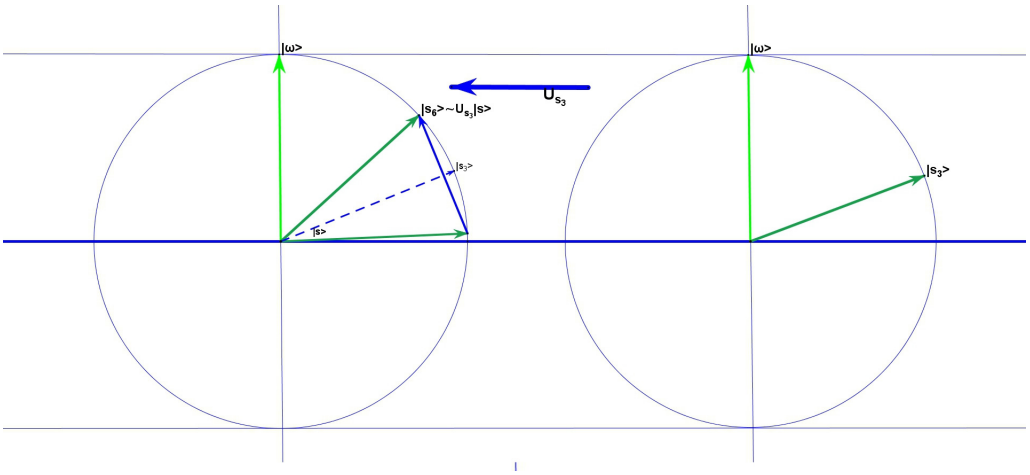

Fig. 9. Получение оператора U_{s_3}.

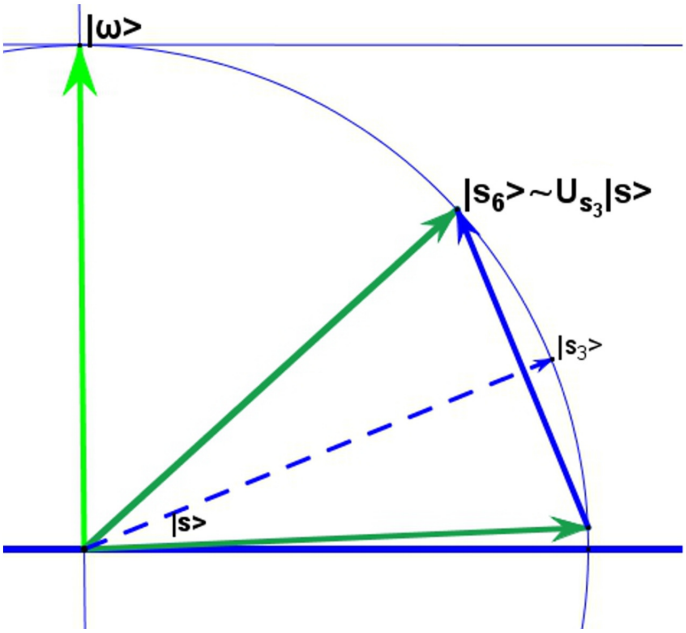

Fig. 10. Действие оператора U_{s_3}.

не суждено сбыться, по крайней мере в рамках принятой нами модели вычислений. В случае, если про функцию f_ω ничего неизвестно, кроме коли-

чества ее особых точек, на которых она равна 1, алгоритм Гровера является оптимальным для поиска этих точек.

Для того, чтобы доказать это, введем некоторые обозначения.

1. $|\omega\rangle$ – искомое "помеченное" состояние.
2. $U(\omega, t) = U_t U_\omega U_{t-1} U_\omega \cdots U_1 U_\omega$ – унитарный оператор, включающий в себя t обращений к оракулу U_ω. Этот оператор имеет смысл обобщенного алгоритма поиска $|\omega\rangle$ посредством вызовов оракула. В алгоритме Гровера, например, все операторы U_i равны оператору U_s. Каждый вызов оракула U_ω мы будем считать за отдельную итерацию алгоритма.
3. $|\psi_0\rangle$ – начальное состояние системы.
4. $|\psi_t\rangle = U(\omega, t)\,|\psi_0\rangle$ – состояние системы после итерации t.
5. $T : U(\omega, T)\,|\psi_0\rangle = |\psi_T\rangle \approx |\omega\rangle$ – количество итераций, после выполнения которых вектор системы оказывается достаточно близок к искомому вектору $|\omega\rangle$.
6. $|\psi_\omega\rangle = |\psi_T\rangle$.
7. $N = 2^n$.

Мы докажем, что существует такой единичный вектор $|\phi\rangle$, для которого выполняется следующее двойное неравенство:

$$4T \geq \sum_{\omega=0}^{N-1} \| \, |\psi_\omega\rangle - |\phi\rangle \, \|^2 \geq 2N - 2\sqrt{N} \tag{2}$$

Поскольку левая и правая части неравенства не зависят от вектора $|\phi\rangle$, неравенство (3) верно всегда,

$$4T \geq 2N - 2\sqrt{N} \tag{3}$$

и мы будем вправе сделать вывод, что для любого алгоритма $U(\omega, t)$ количество итераций T, необходимых для нахождения $|\omega\rangle$ будет иметь порядок $O(\sqrt{N})$, что соответствует сложности алгоритма Гровера:

$$T \geq \sqrt{\frac{N}{2} - \frac{\sqrt{N}}{2}} \sim O(\sqrt{N})$$

Доказательство правой части (2).
Во-первых, заметим, что для разных значений ω, векторы $|\psi_\omega\rangle$ формируют почти ортонормированный базис.

$$\langle \psi_{\omega_i} | \psi_{\omega_j} \rangle =: \Delta_{i,j}$$

$$|\Delta_{i,j}| \approx 0$$

При значении $|\Delta_{i,j}|$ значительно отличающемся от 0 мы можем получить, например, результат $|\omega_i\rangle$ вместо $|\omega_j\rangle$ со значительной вероятностью, что противоречит нашему предположению об эффективности алгоритма $U(\omega, T)$.

Поэтому далее мы будем считать $|\Delta_{i,j}|$ равным 0, а набор векторов $|\psi_\omega\rangle$ – ортонормированным.

Распишем подробнее среднюю часть неравенства (2):

$$\sum_{\omega=0}^{N-1} \| \, |\psi_\omega\rangle - |\phi\rangle \, \|^2 = \sum_{\omega=0}^{N-1} \|\psi_\omega\|^2 + \sum_{\omega=0}^{N-1} \|\phi\|^2 - \sum_{\omega=0}^{N-1} \langle\psi_\omega|\phi\rangle - \sum_{\omega=0}^{N-1} \langle\phi|\psi_\omega\rangle \quad (4)$$

Запишем разложение вектора $|\phi\rangle$ в базисе $|\psi_\omega\rangle$:

$$|\phi\rangle = (x_0, x_1, x_2, \cdots x_{n-1}) \quad (5)$$

$$x_j = a_j + b_j i$$

Подставив (5) в (4) и заменив нормы единичных векторов единицами, получаем:

$$(4) = 2N - \sum_{j=0}^{N-1} (x_j + x_j^*) = 2N - 2\sum_{j=0}^{N-1} Re x_j = 2N - 2\sum_{j=0}^{N-1} a_j \quad (6)$$

Лемма 1.
Для любого набора вещественных чисел c_j, $j = 1 \cdots K$, выполняется неравенство:

$$\left(\sum_{j=1}^{K} c_j\right)^2 \leq K \sum_{j=1}^{K} c_j^2$$

Доказательство.

$$\left(\sum_{j=1}^{K} c_j\right)^2 + \frac{1}{2}\sum_{i,j=1}^{K} (c_i - c_j)^2 =$$

$$= \sum_{i,j=1}^{K} c_i c_j + \frac{1}{2}K\sum_{i=1}^{K} c_i^2 + \frac{1}{2}K\sum_{j=1}^{K} c_j^2 - \sum_{i,j=1}^{K} c_i c_j =$$

$$= K\sum_{j=1}^{K} c_j^2$$

Получается, что:

$$\left(\sum_{j=1}^{K} c_j\right)^2 + \frac{1}{2}\sum_{i,j=1}^{K} (c_i - c_j)^2 = K\sum_{j=1}^{K} c_j^2 \implies$$

$$\implies \left(\sum_{j=1}^{K} c_j\right)^2 \leq K\sum_{j=1}^{K} c_j^2$$

Лемма доказана.

$$\langle\phi|\phi\rangle = 1 \implies \sum_{j=0}^{N-1} x_j x_j^* = 1 = \sum_{j=1}^{N-1} (a_j^2 + b_j^2) \implies \sum_{j=1}^{N-1} a_j^2 \leq 1 \qquad (7)$$

Из (7) и **леммы 1** следует:

$$N \geq N \sum_{j=1}^{N-1} a_j^2 \geq \left(\sum_{j=1}^{N-1} a_j \right)^2 \implies \sqrt{N} \geq \sum_{j=1}^{N-1} a_j \qquad (8)$$

Подставляя (8) в (6), получаем:

$$2N - 2 \sum_{j=0}^{N-1} a_j \geq 2N - 2\sqrt{N},$$

и правая часть неравенства доказана для любого единичного вектора $|\phi\rangle$.

Левая часть неравенства.
Оператор $U(\omega, t)$ мы определили следующим образом:

$$U(\omega, t) = U_t U_\omega U_{t-1} U_\omega \cdots U_1 U_\omega$$

Определим вектор $|\phi_t\rangle$:

$$|\phi_t\rangle = U_t U_{t-1} U_{t-2} \cdots U_1 |\psi_0\rangle$$

Этот вектор получается из вектора начального состояния при действии алгоритма $U(\omega, t)$, из которого убраны (заменены на I) все вызовы оракула U_ω.

Рассмотрим следующую величину:

$$E(\omega, t) := \|(U_\omega - I)|\psi_t\rangle\| = \|(I - 2|\omega\rangle\langle\omega| - I)|\psi_t\rangle\| = 2|\langle\omega|\psi_t\rangle| \qquad (9)$$

Определим функцию $F(t)$:

$$F(t) := \| |\psi_t\rangle - |\phi_t\rangle \|$$

$$F(t) = \|U_t U_\omega |\psi_{t-1}\rangle - U_t |\phi_{t-1}\rangle\| =$$

добавим и вычтем внутри нормы значение $U_t |\psi_{t-1}\rangle$:

$$= \|U_t U_\omega |\psi_{t-1}\rangle - U_t |\psi_{t-1}\rangle + U_t |\psi_{t-1}\rangle - U_t |\phi_{t-1}\rangle\| \leq$$

$$\leq \|U_t(U_\omega - I)|\psi_{t-1}\rangle)\| + \|U_t(|\psi_{t-1}\rangle - |\phi_{t-1}\rangle)\| =$$

Поскольку оператор U_t унитарен, его можно убрать из нормы:

$$= \|(U_\omega - I)|\psi_{t-1}\rangle)\| + \| |\psi_{t-1}\rangle - |\phi_{t-1}\rangle \| = E(\omega, t-1) + F(t-1) \qquad (10)$$

Из (10) и (9) следует:

$$F(T) = \| \, |\psi_\omega\rangle - |\phi_T\rangle \, \| \leq 2 \sum_{t=1}^{T} | \, \langle \omega | \psi_{t-1} \rangle \, | \implies$$

$$\implies \| \, |\psi_\omega\rangle - |\phi_T\rangle \, \|^2 \leq 4 \Bigg(\sum_{t=1}^{T} | \, \langle \omega | \psi_{t-1} \rangle \, | \Bigg)^2 \leq$$

(применим **лемму 1**)

$$\leq 4T \sum_{t=1}^{T} | \, \langle \omega | \psi_{t-1} \rangle \, |^2$$

Теперь мы можем оценить сверху среднюю часть неравенства (2) для вектора $|\phi_T\rangle$:

$$\sum_{\omega=0}^{N-1} \| \, |\psi_\omega\rangle - |\phi_T\rangle \, \|^2 \leq 4T \sum_{\omega=0}^{N-1} \sum_{t=1}^{T} | \, \langle \omega | \psi_{t-1} \rangle \, |^2 =$$

$$= 4T \sum_{t=1}^{T} \sum_{\omega=0}^{N-1} | \, \langle \omega | \psi_{t-1} \rangle \, |^2 =$$

$\sum_{\omega=0}^{N-1} | \, \langle \omega | \psi_{t-1} \rangle \, |^2$ – это квадрат нормы вектора $|\psi_{t-1}\rangle$.

$$= 4T \sum_{t=1}^{T} 1 = 4T^2$$

и левая часть неравенства (2) доказана.

4 Всегда-ли квантовый компьютер имеет преимущество над классическим?

До настоящего момента мы рассматривали задачи, на которых квантовый компьютер показывал себя лучше, чем классический. У читателя может возникнуть ощущение, что это всегда так. К сожалению, нет.

Рассмотрим функции вида

$$f(x) : \{0,1\}^n \to \{0,1\}$$

и определим для каждой из них строку

$$x(f) = x_{N-1} x_{N-2} \cdots x_0$$

$$N = 2^n, \; x_i = f(i)$$

Оракул U_f мы определяли следующим образом:

$$U_f \, |x\rangle \, |y\rangle = |x\rangle \, |y \oplus f(x)\rangle$$

Следовательно

$$\forall i \in \{0, \cdots N-1\}$$

$$U_f |i\rangle |y\rangle = x_i |i\rangle |y \oplus 1\rangle + (1 - x_i) |i\rangle |y\rangle$$

Получается, что после T вызовов оракула U_f коэффициент при каждом базисном векторе будет полиномом степени T от различных x_i.

Обозначим $\tilde{x_i} = 1 - 2x_i$ и рассмотрим функцию "четность":

$$Parity(\tilde{x}) = \prod_{i=0}^{N-1} \tilde{x_i},$$

которая равна 1, если функция f, для которой определена строка $x(f)$, равна 1 для четного числа аргументов. В противном случае функция Parity равна -1.

Представим, что имея оператор U_f в виде квантового черного ящика, мы хотим определить четность функции f.

Классическая сложность этой задачи – N обращений к оракулу. По своему обыкновению, мы разработали некий эффективный с нашей точки зрения квантовый алгоритм, определяющий четность функции f за T обращений к квантовому оракулу U_f. Мы предполагаем при этом, что $T < N/2$, то есть наш алгоритм дает по крайней мере двукратное ускорение по сравнению с классическим перебором.

После запуска этого алгоритма при каждом из базисных векторов образовался полином от x_i степени T, и, следовательно, вероятность измерения любого из них – полином степени $2T < N$. Какие-то из этих векторов, в случае получения их в результате измерения, мы будем интерпретировать, как признак четности функции. Обозначим вероятность получения при измерении векторов этого типа – P^{2T}_{even}.

Рассмотрим сумму по всем функциям вероятностей ответа "четная" на нашем устройстве, умноженных на реальную четность функции:

$$\sum_{\tilde{x}} P^{2T}_{even}(\tilde{x}) \prod_{i=0}^{N-1} \tilde{x_i} \tag{11}$$

Мы можем выделить некоторый индекс j и разбить сумму (11) на две – первая по всем функциям, таким, что $f(j) = 0$, и вторая – по всем функциям, таким, что $f(j) = 1$:

$$(11) = \sum_{\tilde{x}:\tilde{x_j}=1} P^{2T}_{even}(\tilde{x}) \prod_{i \neq j} \tilde{x_i} - \sum_{\tilde{x}:\tilde{x_j}=-1} P^{2T}_{even}(\tilde{x}) \prod_{i \neq j} \tilde{x_i} \tag{12}$$

Поскольку $2T < N$, для каждого одночлена M из P^{2T}_{even} мы можем найти индекс j, такой, что x_j не входит в этот одночлен. Тогда для этого одночлена выражение (12) обращается в 0:

$$\sum_{\tilde{x}:\tilde{x_j}=1} M(\tilde{x}) \prod_{i \neq j} \tilde{x_i} - \sum_{\tilde{x}:\tilde{x_j}=-1} M(\tilde{x}) \prod_{i \neq j} \tilde{x_i} =$$

$$= \sum_{\tilde{x}} M(\tilde{x}) \prod_{i \neq j} \tilde{x_i} - \sum_{\tilde{x}} M(\tilde{x}) \prod_{i \neq j} \tilde{x_i} = 0$$

Поскольку это верно для каждого одночлена, то и вся сумма (11) оказывается равна 0!

Распишем это открытие вот так:

$$0 = \sum_{\tilde{x}} P^{2T}_{even}(\tilde{x}) \prod_{i=0}^{N-1} \tilde{x_i} =$$

$$= \sum_{\tilde{x}:\tilde{x}\ even} P^{2T}_{even}(\tilde{x}) - \sum_{\tilde{x}:\tilde{x}\ odd} P^{2T}_{even}(\tilde{x}) \implies$$

$$\implies \sum_{\tilde{x}:\tilde{x}\ even} P^{2T}_{even}(\tilde{x}) = \sum_{\tilde{x}:\tilde{x}\ odd} P^{2T}_{even}(\tilde{x})$$

Получается, что если мы вызывали оракул U_f меньше, чем $N/2$ раз, то вероятность получить ответ, что функция четная, одинакова для четных и нечетных функций. Значит, квантовые вычисления для этой задачи не дают ускорения даже в 2 раза.

На этой оптимистичной ноте мы заканчиваем наше краткое знакомство с квантовыми вычислениями. Надеюсь, что представленный здесь материал разбудил интерес читателя к дальнейшему изучению и собственным исследованиям в этой области. С моей стороны остается пожелать удачи в этом нелегком, но интересном занятии.

5 Дополнительные материалы

1. **John Preskill. Lecture notes for "Physics 219/Computer Science 219. Quantum Computation" (Formerly Physics 229)**
   http://www.theory.caltech.edu/people/preskill/ph229/index.html
2. **Симулятор квантового компьютера.** http://qc-sim.appspot.com