Классические и квантовые вычисления

А. Китаев; написал М. Вялый

13 лекция (14/05/98)

1. О физической реализации квантового компьютера

Физическая реализация квантового компьютера — чрезвычайно интересная, но сложная задача. Еще несколько лет назад высказывались сомнения в ее принципиальной разрешимости. Дело в том, что любое унитарное преобразование можно реализовать лишь с некоторой точностью. Кроме того, систему спинов или аналогичную квантовую систему нельзя полностью защитить от возмущений со стороны окружающей среды. Все это должно приводить к погрешностям, которые будут накапливаться в процессе вычисления. Через $L \sim \delta^{-1}$ шагов (где δ — точность каждого унитарного преобразования) вероятность ошибки станет порядка единицы. К счастью, эту трудность можно преодолеть, используя *квантовые коды, исправляющие ошибки*. В 1996 году П. Шор предложил схему коррекции ошибок в процессе квантового вычисления (fault-tolerant quantum computation), которая была вскоре усовершенствована. Окончательный результат состоит в следующем. Существует некоторое пороговое значение точности δ_0, такое что при $\delta < \delta_0$ возможно сколь угодно длинное квантовое вычисление. Однако при $\delta > \delta_0$ ошибки накапливаются быстрее, чем их удается исправлять. По разным оценкам, δ_0 лежит в интервале от 10^{-3} до 10^{-6} (точное значение зависит от характера возмущений и используемой схемы коррекции ошибок).

Таким образом, *принципиальных* препятствий для реализации квантового компьютера не существует. Однако задача столь трудна, что ее можно сравнить с задачей об управляемом термоядерном синтезе. В самом деле, необходимо удовлетворить нескольким почти несовместимым требованиям.

1) Элементы квантового компьютера — квантовые биты (спины или что-то подобное) — должны быть изолированы друг от друга и от окружающей среды.

2) Необходимо иметь возможность избирательного воздействия на пару квантовых бит. (Возможно, потребуется несколько типов воздействия на одну и ту же пару, описываемых различными унитарными операторами).

3) Каждый из унитарных операторов должен быть реализован с точностью $\delta < \delta_0$ (см. выше).

4) Реализуемые операторы должны быть достаточно нетривиальны. Точнее говоря, они должны образовывать *полный базис*, чтобы любой другой оператор в определенном смысле выражался через них.

В настоящее время существует несколько подходов к проблеме реализации квантового компьютера.

1. Ионы в ловушке. Это первая и наиболее хорошо разработанная идея. Существует экспериментальная техника, позволяющая удерживать отдельный ион в ловушке из переменного электрического поля в течение длительного времени (порядка 1 часа). Ион можно "охладить" (т.е. погасить колебательное движение) при помощи лазерного луча. Подбирая длительность и частоту лазерных импульсов, можно приготовить произвольную суперпозицию основного и возбужденного состояний. Таким образом, управлять отдельным ионом достаточно легко. В ловушку можно также поместить два или большее число ионов на расстоянии несколько микрон друг от друга и управлять каждым из них в отдельности. Однако организовать взаимодействие между ионами достаточно трудно. Для этой цели предложено использовать коллективные колебательные моды ионов (обычные механические колебания с частотой в несколько мегагерц).

2. **Ядерный магнитный резонанс.** В молекуле с несколькими *различными* ядерными спинами произвольное унитарное преобразование можно реализовать при помощи последовательности импульсов магнитного поля. Это было проверено экспериментально при комнатной температуре. Однако для приготовления начального состояния необходима температура $< 10^{-3}$K. Помимо трудностей с охлаждением, при такой температуре возрастают нежелательные взаимодействия молекул друг с другом. Кроме того, непонятно, как избирательно воздействовать на данный спин, если в молекуле есть несколько одинаковых спинов.

3. **Системы сверхпроводящих гранул.** При сверхнизких температурах единственной степенью свободы микроскопической сверхпроводящей гранулы (диаметром в несколько сотен ангстрем) является ее заряд. Он может изменяться на величину, кратную двум зарядам электрона (поскольку электроны в сверхпроводнике связаны в пары). Меняя внешний электрический потенциал, можно добиться такой ситуации, когда два зарядовых состояния будут иметь почти одинаковую энергию. Эти два состояния можно использовать в качестве базисных состояний квантового бита. Гранулы взаимодействуют между собой посредством джозефсоновских контактов и взаимной электрической емкости. Этим взаимодействием можно управлять. Основная трудность состоит в том, что нужно управлять каждой гранулой в отдельности, причем с высокой точностью. По-видимому, этот подход перспективен, но для его реализации потребуется создание новой технологии.

4. **Анионы.** Анионы — это особые возбуждения в двумерных квантовых системах, в частности, в двумерной электронной жидкости в магнитном поле. Этот подход мне представляется наиболее интересным (поскольку я являюсь его автором), поэтому я опишу его более подробно.

   Основной проблемой при создании квантового компьютера является необходимость реализации унитарных преобразований с точностью $\delta < \delta_0 \sim 10^{-3} - 10^{-6}$. Для этого, как правило, требуется контролировать параметры системы с еще большей точностью. Можно, однако, представить себе ситуацию, когда высокая точность достигается автоматически, т.е. исправление ошибок происходит на физическом уровне. Примером являются двумерные системы с анионными возбуждениями. (Более строго пример такой системы описан в следующем разделе.)

   Все частицы в трехмерном пространстве являются либо бозонами, либо фермионами. Волновая функция бозонов не меняется при перестановке двух частиц, а волновая функция фермионов умножается на -1. В любом случае, при возвращении каждой из частиц в прежнее место состояние системы не меняется. В двумерных системах возможно более сложное поведение. Прежде всего, заметим, что речь пойдет не об элементарных частицах типа электрона, а о возбуждениях, или дефектах в двумерной электронной жидкости. Такие возбуждения похожи на "настоящие" (т.е. элементарные) частицы, но обладают некоторыми необычными свойствами. Возбуждение может иметь дробный электрический заряд

(1/3 или 1/4 от заряда электрона). При движении одного возбуждения *вокруг другого* состояние окружающей их электронной жидкости меняется строго определенным образом, зависящим от типа возбуждений и от *топологии* пути, но не от конкретной траектории. Например, волновая функция может домножиться на число типа $e^{\frac{2\pi}{3}i}$. Возбуждения с таким свойством называются *абелевыми анионами.*

Более интересны *неабелевы анионы*, которые пока не наблюдались экспериментально. (Теория предсказывает существование неабелевых анионов в двумерной электронной жидкости в магнитном поле при факторе заполнения $\frac{5}{2}$). При наличии нескольких неабелевых анионов состояние электронной жидкости является вырожденным, причем кратность вырождения экспоненциально зависит от числа анионов. Другими словами, существует не одно, а много состояний, которые могут образовывать произвольные квантовые суперпозиции. На такую суперпозиция нельзя никак воздействовать, не перемещая анионы, поэтому она идеально защищена от возмущений. Если обвести один анион вокруг другого, суперпозиция подвергнется определенному унитарному преобразованию. Это преобразование является *абсолютно точным.* (Ошибка может возникнуть только если анион "вырвется у нас из рук" вследствие квантового туннелирования).

На первый взгляд, проект с использованием анионов выглядит наименее реалистично. Прежде всего, абелевы анионы не годятся для квантовых вычислений, а неабелевы еще только предстоит найти в эксперименте. Для реализации квантового компьютера нужно контролировать *каждую* частицу с зарядом 1/4 заряда электрона, которые будут двигаться на расстояниях порядка долей микрона друг от друга. Это чрезвычайно сложная техническая задача. Однако, с учетом высокого требования к точности, осуществить любой из перечисленных выше подходов ничуть не легче. Кроме того, идея *топологического квантового вычисления*, лежащая в основе подхода с анионами, может воплотиться каким-либо другим способом. Например, защищенная от возмущений квантовая степень свободы может возникнуть в точке соприкосновения трех фаз двумерной электронной жидкости (пока это только гипотеза).

2. Анионы (иллюстративный пример на основе торического кода)

На примере уже рассмотренного нами торического кода можно дать более точное представление об анионных системах.

Итак, вновь рассмотрим квадратную сетку на торе (а можно и на плоскости — сейчас нас будет интересовать только ее центральная часть). Как и раньше для каждой грани p и звезды каждой вершины s рассмотрим проверочные операторы

$$A_p^{(z)} = \prod_{j\in\,\textit{граница}(p)} \sigma_j^z, \quad A_s^{(x)} = \prod_{j\in\,\textit{звезда}(s)} \sigma_j^x.$$

Состояние кодового подпространства задается условиями $A_p^{(z)}|\xi\rangle = |\xi\rangle$, $A_s^{(x)}|\xi\rangle = |\xi\rangle$. Их можно переписать другим способом. Рассмотрим следующий *гамильтониан* — эрмитов оператор

$$\mathcal{H} = \sum_p (A_r^{(p)} + 1) + \sum_s (A_s^{(x)} + 1).$$

Этот оператор неотрицательный, причем его нулевое подпространство в точности совпадает с кодовым подпространством торического кода. Таким образом, вектора из кодового подпространства являются собственными и обладают наименьшей энергией (т. е. собственным значением гамильтониана). Такие вектора называются *основными состояниями*, а вектора из ортогонального дополнения — *возбужденными состояниями*.

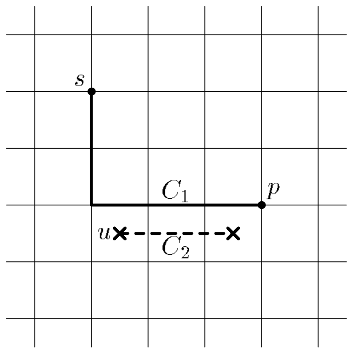
Рассмотрим возбужденные состояния с наименьшей ненулевой энергией, когда нарушено ровно два условия (например, вершинных). (Число нарушенных условий каждого типа четное, поскольку $\prod_s A_s^{(x)} = \prod_u A_u^{(z)} = 1$). Тогда для двух вершин, в которых кодовые условия нарушаются, выполнено

$$A_s^{(x)}|\eta\rangle = -|\eta\rangle, \quad A_p^{(x)}|\eta\rangle = -|\eta\rangle.$$

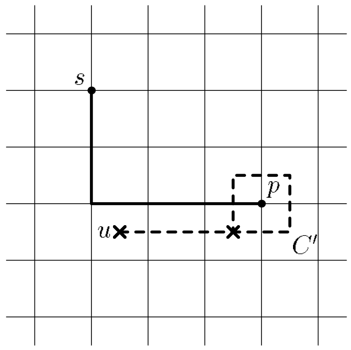
Как можно получить состояние $|\eta\rangle$ из кодового состояния $|\xi\rangle$? Соединим p и s решеточным путем C_1 и подействуем на $|\xi\rangle$ оператором $B = \prod_{j\in C_1} \sigma_j^z$. Этот оператор коммутирует с проверочными вершинными операторами для всех промежуточных вершин пути C, а в концах — антикоммутирует: $BA_s^{(x)} = -A_s^{(x)}B$. Положим $|\eta\rangle = B|\xi\rangle$ и покажем, что $|\eta\rangle$ удовлетворяет требуемым свойствам. Для вершины s (аналогично и для p) имеем

$$|\eta\rangle = B|\xi\rangle = BA_s^{(x)}|\xi\rangle = -A_s^{(x)}B|\xi\rangle = -A_s^{(x)}|\eta\rangle.$$

($A_s^{(x)}|\xi\rangle = |\xi\rangle$ так как состояние $|\xi\rangle$ — кодовое).



Любое состояние системы можно построить из элементарных возбуждений двух типов, одни из которых “живут” на вершинах, другие — на гранях. Элементарное возбуждение — это просто нарушенное кодовое условие, но теперь мы думаем о нем как о частице. Частицы-возбуждения можно двигать, создавать и уничтожать. Пара возбуждений первого типа получается из основного (кодового) состояния действием оператора B, приведенного выше; пара возбуждений второго типа — действием оператора $D = \prod_{j\in C_2} \sigma_j^x$, где C_2 — путь, соединяющий две грани, как показано на рисунке. Как и раньше проверяется, что $A_u^{(z)}(D|\xi\rangle) = -D|\xi\rangle$.



Что случится (см. рис. на врезке), если двигать возбуждение одного типа (крестик) вокруг возбуждения второго типа (кружочка)? Мы считаем, что каждое из возбуждений входит в какую-нибудь пару; где-то вдалеке могут быть еще возбуждения, которые ни на что не влияют. (Разбиение возбуждений на пары произвольно, и ответ от него не зависит). Движение возбуждения описывается оператором $\prod_{j\in C'} \sigma_j^x$, зависящим от контура обхода C'. В результате мы получим

$$|\psi\rangle \;\mapsto\; \prod_{j\in C'} \sigma_j^x|\psi\rangle = A_p^{(x)}|\psi\rangle = -|\psi\rangle.$$

То есть вектор состояния домножился на -1. Это и означает, что рассматриваемые возбуждения являются (абелевыми) анионами.

На торе можно двигать частицы по двум различным циклам, образующим базис в группе гомологий. Например, можно создать из основного состояния пару возбуждений одного типа, обнести одно из возбуждений по циклу и проаннигилировать со вторым возбуждением. Этот процесс описывается некоторым оператором, действующим на кодовом подпространстве, — произведением σ_j^z вдоль пути на решетке, либо σ_j^x вдоль пути на двойственной решетке. Поскольку существует два типа возбуждений, мы имеем 4 таких оператора: $Y_1^{(z)}$ и $Y_2^{(x)}$ соответствуют одному базисному циклу, а $Y_2^{(z)}$ и $Y_1^{(x)}$ — другому. Эти операторы действуют на два закодированных q-бита как σ_i^z, σ_i^x $(i=1,2)$, потому что они обладают такими же коммутационными соотношениями: $Y_i^{(x)}Y_i^{(z)} = -Y_i^{(z)}Y_i^{(x)}$ (остальные пары коммутируют).

3. Как производить вычисления, устойчивые к ошибкам

Общий случай вычислений, устойчивых к ошибкам, довольно сложен. Трудность состоит в том, что нужно производить действия с закодированными кубитами, не раскодирую их ни на каком этапе вычислений (иначе они окажутся не защищенными от возмущений). Опишем один из простейших способов таких вычислений, позволяющий реализовать оператор $\Lambda(\sigma^x) : |a,b\rangle \mapsto |a, a\oplus b\rangle$. В данном случае достаточно ограничиться побитными операциями над кодовыми q-битами.

Выберем некоторый квантовый код для q-битов a и b. Будем рассматривать сиплектические коды специального вида, когда в проверочные операторы входят либо только σ^x, либо только σ^z, причем без знака "-".

$$m \text{ штук} \begin{cases} A_l^{(x)} = \sigma(f_l^{(x)}); \\ A_r^{(z)} = \sigma(f_r^{(z)}); \end{cases} \quad f_l^{(x)}, f_r^{(z)} \in (\mathbb{F}_2)^n.$$

Пусть $m = n-1$, тогда размерность кодового подпространства $\mathcal{L} \subset \mathbb{B}^{\otimes n}$ равна $2^{n-m} = 2$. Например, можно использовать код Шора или торический код с дополнительным проверочным оператором $Y_2^{(z)}$. Чтобы описывать действие на закодированном бите, мы должны отождествить некоторым образом $\mathcal{L}$ и $\mathbb{B}$ (пространство состояний одного q-бита). Для этого введем *информационные операторы* $Y^{(x)} = \sigma(g^{(x)})$ и $Y^{(z)} = \sigma(g^{(z)})$ (соответствующие операторам σ^x, σ^z в пространстве состояний одного q-бита). Операторы $Y^{(x)}, Y^{(z)}$ должны быть произведениями σ_j^x и σ_j^z, соответственно; они должны коммутировать со всеми проверочными операторами и антикоммутировать друг с другом:

$$Y^{(x)}Y^{(z)} = -Y^{(z)}Y^{(x)}; \qquad Y^{(x,z)}A_l^{(x,z)} = A_l^{(x,z)}Y^{(x,z)}.$$

(Для торического кода информационные операторы совпадают с введенными ранее операторами $Y_1^{(x)}$ и $Y_1^{(z)}$).

Будем применять $\Lambda(\sigma^x)$ побитно к кодовым битам и покажем, что в результате $\Lambda(\sigma^x)$ будет применен к закодированному вектору.

Пусть $U = \Lambda(\sigma^x)$. Оператор U действует на σ-операторах так:

$$U\sigma_1^x U^\dagger = \sigma_1^x\sigma_2^x; \quad U\sigma_2^x U^\dagger = \sigma_2^x; \quad U\sigma_1^z U^\dagger = \sigma_1^z; \quad U\sigma_2^z U^\dagger = \sigma_1^z\sigma_2^z.$$

При побитовом действии оператора U на кодовые биты будут выполняться соотношения

$$U^{\otimes n}\sigma_{1,j}^x (U^\dagger)^{\otimes n} = \sigma_{1,j}^x\sigma_{2,j}^x; \; U^{\otimes n}\sigma_{2,j}^x (U^\dagger)^{\otimes n} = \sigma_{2,j}^x; \; U^{\otimes n}\sigma_{1,j}^z (U^\dagger)^{\otimes n} = \sigma_{1,j}^z; \; U^{\otimes n}\sigma_{2,j}^z (U^\dagger)^{\otimes n} = \sigma_{1,j}^z\sigma_{2,j}^z.$$

Поэтому для проверочных операторов будут выполняться соотношения:

$$\begin{aligned} U^{\otimes n}A_{1,j}^{(x)}(U^\dagger)^{\otimes n} &= A_{1,j}^{(x)}A_{2,j}^{(x)}; & U^{\otimes n}A_{2,j}^{(x)}(U^\dagger)^{\otimes n} &= A_{2,j}^{(x)}; \\ U^{\otimes n}A_{1,j}^{(z)}(U^\dagger)^{\otimes n} &= A_{1,j}^{(z)}; & U^{\otimes n}A_{2,j}^{(z)}(U^\dagger)^{\otimes n} &= A_{1,j}^{(z)}A_{2,j}^{(z)}, \end{aligned}$$

Следовательно, если $|\xi\rangle \in \mathbb{B}^{\otimes 2n}$ — кодовый вектор (т. е. $A_{1,j}^{(x;z)}|\xi\rangle = |\xi\rangle$ и $A_{2,j}^{(x;z)}|\xi\rangle = |\xi\rangle$), то вектор $U^{\otimes n}|\xi\rangle$ также является кодовым. Информационные операторы преобразуются под действием

$U^{\otimes n}$ аналогичным образом:

$$\begin{aligned} U^{\otimes n} Y_1^{(x)} (U^\dagger)^{\otimes n} &= Y_1^{(x)} Y_2^{(x)}; \qquad & U^{\otimes n} Y_2^{(x)} (U^\dagger)^{\otimes n} &= Y_2^{(x)}; \\ U^{\otimes n} Y_1^{(z)} (U^\dagger)^{\otimes n} &= Y_1^{(z)}; \qquad & U^{\otimes n} Y_2^{(z)} (U^\dagger)^{\otimes n} &= Y_1^{(z)} Y_2^{(z)}, \end{aligned}$$

Это и означает, что на закодированные биты действует оператор U.

4. О вычислениях в симплектическом базисе

Опишем смешанную, квантово-классическую, модель вычислений. Например, помимо уже рассматривавшегося квантового управления, можно реализовывать и классическое управление:

Квантовое управление	Классическое управление
$\Lambda(W)\vert 0\rangle \otimes \vert\xi\rangle = \vert 0\rangle \otimes \vert\xi\rangle$	$\Lambda_{\text{кл}}(W)(\vert 0\rangle\langle 0\vert \otimes \rho) = \vert 0\rangle\langle 0\vert \otimes \rho$
$\Lambda(W)\vert 1\rangle \otimes \vert\xi\rangle = \vert 1\rangle \otimes W\vert\xi\rangle$	$\Lambda_{\text{кл}}(W)(\vert 1\rangle\langle 1\vert \otimes \rho) = \vert 1\rangle\langle 1\vert \otimes (W\rho W^\dagger)$

Перечислим, что можно делать в смешанной модели вычислений.

1) Можно делать все классические операции (есть полный классический базис).

2) Можно превратить квантовый бит в классический и наоборот.

3) Есть набор некоторый набор квантовых операторов; можно применять их с классическим управлением.

Далее будем рассматривать такую модель вычислений, где в качестве квантовых операций используется ***симплектический*** базис, т.е. набор операторов $\{S, K, \Lambda(\sigma^x)\}$.

Симплектических операторов достаточно, чтобы кодировать и декодировать симплектические коды, и исправлять ошибки в таких кодах. Но симплектический базис не является полным.

Если добавить к симплектическому базису оператор $R = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{\mathrm{i}} \end{pmatrix}$, где $\sqrt{\mathrm{i}} = \dfrac{1+\mathrm{i}}{\sqrt{2}}$, то получим уже полный базис (любой оператор может быть сколь угодно точно приближен композициями R и симплектических).

Оказывается, что в смешанной модели вычислений можно применение оператора R заменить использованием дополнительных q-битов, приготовленных в состоянии $|\eta\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{\sqrt{\mathrm{i}}}{\sqrt{2}}|1\rangle$.

Покажем как реализовать применение оператора R к произвольному состоянию ρ. Достаточно (по линейности) ограничиться применением R к чистому состоянию $|\xi\rangle = c_0|0\rangle + c_1|1\rangle$. Запишем $|\xi\rangle \otimes |\eta\rangle$ в виде матрицы, строки которой соответствуют первому q-биту, а столбцы — второму. Применим оператор $\Lambda(\sigma^x)$:

$$|\xi\rangle \otimes |\eta\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} c_0 & \sqrt{\mathrm{i}}\, c_0 \\ c_1 & \sqrt{\mathrm{i}}\, c_1 \end{pmatrix} \xrightarrow{\Lambda(\sigma^x)} \begin{pmatrix} c_0 & \sqrt{\mathrm{i}}\, c_0 \\ \sqrt{\mathrm{i}}\, c_1 & c_1 \end{pmatrix}.$$

Теперь измерим значение второго q-бита. Если измеряемое значение равно 0, то получим в первом бите состояние, задаваемое первым столбцом матрицы, если 1 — то получим состояние, задаваемое вторым столбцом. Поэтому после измерения имеем смешанное квантово-классическое состояние

$$\frac{1}{2}\left(\left(c_0|0\rangle + \sqrt{\mathrm{i}}\, c_1|1\rangle\right)\left(c_0^*\langle 0| + (\sqrt{\mathrm{i}}\, c_1)^*\langle 1|\right),\, 0\right) + \frac{1}{2}\left(\left(\sqrt{\mathrm{i}}\, c_0|0\rangle + c_1|1\rangle\right)\left((\sqrt{\mathrm{i}}\, c_0)^*\langle 0| + c_1^*\langle 1|\right),\, 1\right).$$

Теперь мы будем использовать второй бит как классический управляющий бит для применения оператора K к первому биту. Если значение второго бита равно 0, то в первом бите уже находится $R|\xi\rangle$. Если же значение управляющего бита равно 1, то после применения оператора K получим

$$K(\sqrt{\mathrm{i}}\,c_0|0\rangle + c_1|1\rangle) = \sqrt{\mathrm{i}}\,(c_0|0\rangle + \sqrt{\mathrm{i}}\,c_1|1\rangle),$$

то есть $\sqrt{\mathrm{i}}\,R|\xi\rangle$. Итак, в любом случае состояние первого бита будет задаваться матрицей плотности

$$(c_0|0\rangle + \sqrt{\mathrm{i}}\,c_1|1\rangle)(c_0^*\langle 0| + (\sqrt{\mathrm{i}}\,c_1)^*\langle 1|) \;=\; R|\xi\rangle\langle\xi|R^\dagger.$$