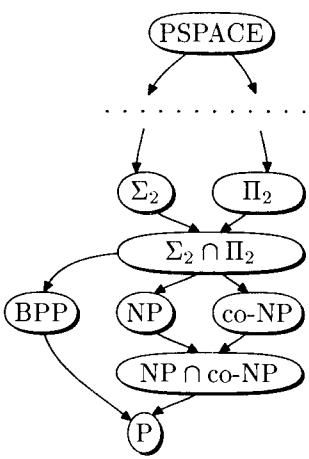
процессе игры были названы состояния M. Если состояние правого конца получается из состояния левого конца за один такт работы M, то Б выиграли, иначе выиграли Ч.

Если ответ M на слове x действительно «да», то белым нужно всё время говорить правду, это гарантирует им выигрыш.

Если ответ M на слове x — «нет», то при любом ходе белых на одном из промежутков (или на обоих) будет содержаться ошибка. Ч должен указывать каждый раз именно этот промежуток. □

Задача 4.1. Докажите, что класс языков, распознаваемых недетерминированными машинами, работающими на памяти S, содержится в классе языков, распознаваемых детерминированными машинами, работающими на памяти $\mathrm{poly}(S)$.

В качестве следствия теоремы 4.2 получаем включения всех определённых выше классов Σ_k, Π_k в класс PSPACE. Взаимное соотношение этих классов можно изобразить диаграммой включений, показанной слева. На этой диаграмме от бóльшего класса к меньшему можно пройти, двигаясь по стрелкам. Внизу располагается класс P, отвечающий играм с 0 ходов, затем идут дополняющие друг друга классы, отвечающие играм с конечным числом ходов (для одного хода это NP и co-NP, для двух ходов — Σ_2 и Π_2 и т. д.). Завершается эта диаграмма классом PSPACE, который определяется произвольными играми с одним естественным условием — время игры должно быть полиномиально ограничено размером входного слова. Мы уже доказали все включения, изображённые на этой диаграмме. Ни про одно из включений, следующих из этой диаграммы, неизвестно, является ли оно строгим. Быть может, скажем, P = PSPACE. С другой стороны, возможно и так, что PSPACE = EXPTIME, где EXPTIME обозначает (не рассматривавшийся нами) класс языков, вычислимых за экспоненциальное время $2^{\mathrm{poly}(n)}$. Впрочем, наиболее популярна гипотеза о том, что все включения, изображённые на диаграмме — строгие.



Задача 4.2. *Машина Тьюринга с оракулом* A — это МТ с дополнительной *оракульной* лентой, куда она (машина) может записывать слова, а затем за один такт работы проверять, принадлежит ли записанное

на оракульной ленте слово языку A. По двум сложностным классам $\mathcal{X}$ и $\mathcal{Y}$ можно определить класс $\mathcal{X}^{\mathcal{Y}}$ таких языков, которые распознаются машинами из класса $\mathcal{X}$ с оракулами из $\mathcal{Y}$.

Докажите, что $\mathrm{P}^{\Sigma_k} = \mathrm{P}^{\Pi_k} \subseteq \Sigma_{k+1} \cap \Pi_{k+1}$.

В классе PSPACE существуют полные задачи (относительно полиномиальной сводимости). Простейший вариант получается применением предыдущей теоремы.

Задача TQBF. Задаётся предикатом

$TQBF(x) \Leftrightarrow$ x есть истинная булева формула с кванторами (True Quantified Boolean Formula), т. е. формула вида

$$\mathrm{Q}_1 y_1 \ldots \mathrm{Q}_n y_n F(y_1, \ldots, y_n),$$

где $y_i \in \mathbb{B}$, F — некоторая логическая формула, а Q_i — либо $\forall$, либо $\exists$. По определению, $(\forall y_1\, A(y_1)) = (A(0) \wedge A(1))$, а $(\exists y_1\, A(y_1)) = (A(0) \vee A(1))$.

Теорема 4.3. TQBF PSPACE-*полна.*

Доказательство. Построим сведе́ние любого языка $L \in$ PSPACE к задаче TQBF. Для этого превратим МТ, вычисляющую результат игры (предикат $W(\cdot)$), в схему, а ходы игроков закодируем булевыми переменными. Тогда наличие выигрышной стратегии у белых задаётся условием

$$\exists w_1^1\, \exists w_1^2 \ldots \exists w_1^{p(|x|)}\, \forall b_1^1 \ldots \forall b_1^{p(|x|)}\, S(x, w_1^1, w_1^2, \ldots),$$

где $S(\cdot)$ обозначает результат вычисления по схеме.

Чтобы превратить S в булеву формулу, добавим новые переменные y_i (значение, вычисленное при i-м присваивании в схеме) и заменим $S(\cdot)$ на формулу вида

$$\exists y_1, \ldots, \exists y_{\text{размер схемы}}\ (y_1 \Leftrightarrow R_1) \wedge \cdots \wedge (y_s \Leftrightarrow R_s) \wedge y_s,$$

где s — размер схемы, R_i — правая часть i-го присваивания.

После этой подстановки получим квантифицированную булеву формулу, которая истинна в точности для $x \in L$. $\square$

Часть II

Квантовые вычисления

Как уже говорилось во введении, обычные компьютеры не используют всех возможностей, предоставляемых природой. В них выполняются преобразования на конечных множествах состояний (действия с 0 и 1), а в природе есть возможность делать унитарные преобразования, т.е. действовать на бесконечном множестве.[9] Эта возможность описывается квантовой механикой. Устройства (реальные или воображаемые), использующие эту возможность, называются квантовыми компьютерами.

Заранее неясно, увеличиваются ли вычислительные возможности при переходе от преобразований конечных множеств к унитарным преобразованиям конечномерных пространств. Сейчас есть основания полагать, что такое увеличение действительно происходит. В качестве примера можно привести задачу о разложении числа на множители: для обычных компьютеров неизвестны полиномиальные алгоритмы её решения, а для квантовых компьютеров такие алгоритмы есть.

Обычный компьютер работает с состояниями из конечного числа битов. Каждый бит может находиться в одном из двух состояний 0 или 1. Состояние всей системы задаётся указанием значений всех битов. Поэтому множество состояний $\mathbb{B}^n = \{0,1\}^n$ конечно и имеет мощность 2^n.

Квантовый компьютер работает с конечными наборами элементарных состояний, называемых q-битами. Каждый q-бит имеет два выделенных состояния (если считать q-биты спинами, то это состояния «спин вверх» и «спин вниз»). Указание выделенных состояний для каждого q-бита системы задаёт не все возможные состояния системы, а только базисные. Возможны также любые линейные комбинации базисных состояний с комплексными коэффициентами. Базисные состояния

[9] Конечно, настоящей бесконечности в природе не бывает. В данном случае дело в том, что унитарное преобразование можно задать лишь с некоторой точностью — подробности см. в разделе 7.

мы будем обозначать $|x_1, \ldots, x_n\rangle$, где $x_j \in \mathbb{B}$, или $|x\rangle$, где $x \in \mathbb{B}^n$. Произвольное состояние системы может быть представлено в виде[10)]

$$|\psi\rangle = \sum_{(x_1,\ldots,x_n)\in\mathbb{B}^n} c_{x_1,\ldots,x_n}|x_1,\ldots,x_n\rangle, \text{ где } \sum_{(x_1,\ldots,x_n)\in\mathbb{B}^n} |c_{x_1,\ldots,x_n}|^2 = 1.$$

Пространство состояний для такой системы — конечномерное (размерности 2^n) пространство над полем комплексных чисел.

Состояние

обычного компьютера	**квантового компьютера**
$\square\ \square\ \cdots\ \square$ биты	$\square\ \square\ \cdots\ \square$ q-биты
$x_1\ x_2\ \ldots\ x_n\ \ x_j \in \mathbb{B}$	базисное: $\lvert x_1,\ x_2,\ \ldots,\ x_n\rangle$, $\quad x_j \in \mathbb{B}$
	произвольное: $\sum_{x\in\mathbb{B}^n} c_x\lvert x\rangle$, где $\sum_{x\in\mathbb{B}^n} \lvert c_x\rvert^2 = 1$

Небольшое уточнение: если умножить вектор $\sum_x c_x|x\rangle$ на *фазовый множитель* $e^{i\varphi}$ (φ — вещественное), то получится физически неотличимое состояние. Таким образом, состояние квантового компьютера — это вектор единичной длины, заданный с точностью до фазового множителя.

Вычисление можно представлять как последовательность преобразований на множестве состояний системы. Опишем, какие преобразования возможны в классическом, а какие — в квантовом случае.

Классический случай:	**Квантовый случай:**
преобразования — это функции из $\mathbb{B}^n$ в $\mathbb{B}^n$.	преобразования — это унитарные операторы, то есть операторы, сохраняющие длину вектора $\sum_{x\in\mathbb{B}^n} \lvert c_x\rvert^2$.

Замечание. Всё сказанное относится только к замкнутым системам. Реальный квантовый компьютер — это часть большой системы (Вселенной), взаимодействующая с остальным миром. Квантовые состояния и преобразования открытых систем будут рассмотрены в разделах 9–10.

Теперь нужно дать формальное определение квантового вычисления. Как и в классическом случае, можно определить квантовые машины Тьюринга или квантовые схемы. Мы выбираем второй подход, который удобнее по ряду причин.

[10)] Скобки $|\ldots\rangle$ в записи $|\psi\rangle$ не обозначают никакой операции над объектом ψ — они просто указывают на то, что ψ является вектором.

5. Определения и обозначения

Пространство состояний системы из n q-битов $\mathbb{C}^{2^n}$ можно записать в виде тензорного произведения $\mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2 = \left(\mathbb{C}^2\right)^{\otimes n}$. Сомножители соответствуют *пространству состояний одного q-бита.*

Тензорное произведение двух пространств L и M, в которых фиксированы базисы $\{e_1, \ldots, e_l\}$ и $\{f_1, \ldots, f_m\}$, можно определить как пространство с базисом из элементов $e_j \otimes f_k$. (В данном случае $e_j \otimes f_k$ — это то же самое, что (e_j, f_k), т. е. просто пара векторов.) Размерность тензорного произведения равна lm (произведению размерностей сомножителей).

Такое определение *неинвариантно*, т. е. зависит от выбора базисов в перемножаемых пространствах. Можно дать *инвариантное* определение. Для этого рассмотрим вначале пространство (бесконечномерное) с базисом $e \otimes f$, где $e \in L$, $f \in M$ — произвольные векторы из перемножаемых пространств. Тензорное произведение будет факторпространством этого пространства по подпространству, порожденному векторами вида

$$(e_1 + e_2) \otimes f - e_1 \otimes f - e_2 \otimes f,$$
$$e \otimes (f_1 + f_2) - e \otimes f_1 - e \otimes f_2,$$
$$(\lambda e) \otimes f - e \otimes (\lambda f),$$
$$\lambda(e \otimes f) - (\lambda e) \otimes f.$$

Другими словами, указанные векторы считаются равными 0.

Можно доказать, что данные определения эквивалентны.

В нашем случае имеется естественный выделенный базис (соответствующий выделенным состояниям): для $\mathbb{C}^2$ — $\{|0\rangle, |1\rangle\}$, а для $\left(\mathbb{C}^2\right)^{\otimes n}$ — $\{|x_1, \ldots, x_n\rangle\}$, $x_j \in \mathbb{B}$. Пространство $\mathbb{C}^2$ с выделенным базисом обозначается через $\mathcal{B}$. Выделенный базис считается ортонормированным, это задаёт скалярное произведение на пространстве состояний. Коэффициенты $c_{x_1,\ldots,x_n}$ разложения вектора $|\psi\rangle$ по этому базису называются *амплитудами.* Их физический смысл состоит в том, что квадрат модуля амплитуды $|c_{x_1,\ldots,x_n}|^2$ интерпретируется как вероятность обнаружить систему в данном базисном состоянии. Как и должно быть, суммарная вероятность всех состояний равна 1, поскольку длина вектора предполагается единичной. (Вероятности будут подробно обсуждаться позже; до некоторых пор мы будем заниматься линейной алгеброй — изучать унитарные операторы на пространстве $\mathcal{B}^{\otimes n}$).

Мы будем использовать (и уже использовали) принятые в физике обозначения, относящиеся к векторам и скалярному произведению в гильбертовом пространстве (их ввёл Дирак). Векторы обозначаются

$|\xi\rangle$, скалярное произведение — $\langle\xi|\eta\rangle$. Если $|\xi\rangle = \sum_x a_x|x\rangle$ и $|\eta\rangle = \sum_x b_x|x\rangle$, то $\langle\xi|\eta\rangle = \sum_x a_x^* b_x$. (Здесь и далее a^* обозначает комплексное сопряжение.) В записи векторов скобки нужны лишь «для красоты» — они указывают на тип объекта и придают симметрию обозначениям (см. ниже). Вместо $|\xi\rangle$ можно было бы написать просто ξ, хотя это и не принято. Поэтому $|\xi_1+\xi_2\rangle = |\xi_1\rangle + |\xi_2\rangle$ — и то, и другое обозначает вектор $\xi_1+\xi_2$.

Скалярное произведение антилинейно по *первому* аргументу[11] и линейно по второму, т. е.

$$\langle\xi_1+\xi_2|\eta\rangle = \langle\xi_1|\eta\rangle + \langle\xi_2|\eta\rangle, \qquad \langle\xi|\eta_1+\eta_2\rangle = \langle\xi|\eta_1\rangle + \langle\xi|\eta_2\rangle,$$
$$\langle c\xi|\eta\rangle = c^*\langle\xi|\eta\rangle, \qquad \langle\xi|c\eta\rangle = c\langle\xi|\eta\rangle.$$

Если в обозначении скалярного произведения взять левую половину, то получим *бра-вектор* $\langle\xi|$, т. е. линейный функционал на *кет-векторах* (векторах нашего пространства). Бра- и кет-векторы находятся во взаимно однозначном соответствии. (Тем не менее, нужно их как-то различать — именно для этого и были введены угловые скобки.) Из-за антилинейности скалярного произведения по первому аргументу имеем равенство $\langle c\xi| = c^*\langle\xi|$. Бра-вектор можно записать в виде строки, а кет-вектор — в виде столбца (чтобы его можно было умножить слева на матрицу):

$$\langle\xi| = c_0^*\langle 0| + c_1^*\langle 1| = (c_0^*, c_1^*), \qquad |\xi\rangle = c_0|0\rangle + c_1|1\rangle = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}.$$

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 $|\eta\rangle$. Так появляется сопряженный оператор $A^\dagger$: по определению, $\langle A^\dagger\xi|$ (бра-вектор, соответствующий $A^\dagger|\xi\rangle$) равен линейному функционалу $\langle\xi|A$. Из определения сразу следует, что

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

Унитарный оператор — это линейный оператор, сохраняющий скалярное произведение. Условие

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

эквивалентно тому, что $U^\dagger U = I$ (где I — тождественный оператор).

Наше определение скалярного произведения в $\mathcal{B}^{\otimes n}$ согласовано с тензорным произведением:

$$\Big(\langle\xi_1|\otimes\langle\xi_2|\Big)\Big(|\eta_1\rangle\otimes|\eta_2\rangle\Big) = \langle\xi_1|\eta_1\rangle\,\langle\xi_2|\eta_2\rangle\,.$$

[11] Обратите внимание, что математики обычно считают, что скалярное произведение в унитарном пространстве антилинейно по *второму* аргументу.

В дальнейшем будет использоваться *тензорное произведение операторов*. Оно действует в тензорном произведении пространств, на которых действуют сомножители, по правилу

$$(A \otimes B)|\xi\rangle \otimes |\eta\rangle = A|\xi\rangle \otimes B|\eta\rangle.$$

Если операторы заданы в матричном виде в некотором базисе, т. е.

$$A = \sum_{j,k} a_{jk}|j\rangle\langle k|, \quad B = \sum_{j,k} b_{jk}|j\rangle\langle k|$$

(легко понять, что $|j\rangle\langle k|$ — линейный оператор: $|j\rangle\langle k|\,|\xi\rangle = \langle k|\xi\rangle|j\rangle$), то матричные элементы оператора $C = A \otimes B$ имеют вид $c_{(jk)(lm)} = a_{jl}b_{km}$.

Вычисление состоит из преобразований, считаемых элементарными (выполняемых за единицу времени).

Элементарное преобразование **в классическом случае**: такая функция из $\mathbb{B}^n$ в $\mathbb{B}^n$, которая зависит от небольшого (не зависящего от n) числа битов и изменяет также небольшое число битов.

Элементарное преобразование **в квантовом случае**: тензорное произведение произвольного унитарного оператора, действующего на части сомножителей $\mathcal{B}^{\otimes r}$, где r мало ($r = O(1)$), и тождественного оператора, действующего на остальных сомножителях.

Тензорное произведение некоторого оператора U, действующего на множестве q-битов A, и тождественного оператора, действующего на остальных q-битах, будем обозначать $U[A]$. (В частности, $U[1,\ldots,r] = U \otimes I$ обозначает действие на первых r q-битах.)

Пример 5.1. Приведём матрицу оператора $H[2]$, действующего в пространстве $\mathcal{B}^{\otimes 3}$. Оператор $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ действует на второй q-бит, на остальных q-битах действие тождественное. Базисные векторы расположены в лексикографическом порядке: от $|000\rangle$ до $|111\rangle$.

$$H[2] = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{pmatrix}$$

С этого места начинается вычислительная сложность. Пусть $r = 2$, тогда U — некоторая матрица 4×4, $U \otimes I$ — матрица размерности $2^n \times 2^n$, у которой по диагонали стоят блоки из матриц U. Эта ма-

трица представляет один элементарный шаг. Когда применяется несколько таких операторов к разным парам q-битов, результат будет выглядеть гораздо сложнее. Не видно способа определить этот результат, кроме прямого перемножения матриц. Поскольку размеры матриц экспоненциально велики, потребуется экспоненциальное время для их перемножения.

Заметим однако, что вычисление матричных элементов возможно на полиномиально ограниченной памяти. Пусть нужно найти матричный элемент U_{xy} оператора

$$U = U^{(l)}[j_l, k_l]\, U^{(l-1)}[j_{l-1}, k_{l-1}] \cdot \ldots \cdot U^{(2)}[j_2, k_2]\, U^{(1)}[j_1, k_1].$$

Очевидно, что

$$\left(U^{(l)} \cdot \ldots \cdot U^{(1)}\right)_{x_l x_0} = \sum_{x_{l-1}, \ldots, x_1} U^{(l)}_{x_l x_{l-1}} \cdot \ldots \cdot U^{(1)}_{x_1 x_0}. \tag{5.1}$$

(Здесь $x_0, \ldots, x_l$ — строки длиной n битов.) Для вычисления этой суммы достаточно $(l-1)$-го регистра для хранения текущих значений $x_{l-1}, \ldots, x_1$, ещё одного регистра для хранения частичной суммы и некоторого фиксированного количества регистров для вычисления промежуточных произведений.

Определение 5.1. ***Квантовая схема.*** *Пусть $\mathcal{A}$ — некоторое множество унитарных операторов (базис). Тогда квантовая схема в базисе $\mathcal{A}$ — это последовательность $U_1[A_1], \ldots, U_l[A_l]$, где A_j — множества q-битов, $U_j \in \mathcal{A}$.*

Оператор, реализуемый квантовой схемой. *Это оператор $U\colon \mathcal{B}^{\otimes n} \to \mathcal{B}^{\otimes n}$, равный $U_l[A_l] \cdot \ldots \cdot U_1[A_1]$.*

Это определение не очень хорошо, так как не учитывает возможность использования дополнительной памяти в процессе вычисления. Поэтому дадим ещё одно определение.

Оператор $U\colon \mathcal{B}^{\otimes n} \to \mathcal{B}^{\otimes n}$, реализуемый схемой в расширенном смысле. *Это такой оператор, что произведение*

$$W = U_l[A_l] \cdot \ldots \cdot U_1[A_1],$$

действующее на N q-битов, $N \geqslant n$, для любого вектора $|\xi\rangle \in \mathcal{B}^{\otimes n}$ удовлетворяет условию $W(|\xi\rangle \otimes |0^{N-n}\rangle) = (U|\xi\rangle) \otimes |0^{N-n}\rangle$.

Таким образом, мы «берём напрокат» дополнительную память, заполненную нулями, и должны возвратить ее в прежнем состоянии. Какой смысл имеет такое определение? Зачем нужно требовать, чтобы дополнительные q-биты вернулись в состояние $|0^{N-n}\rangle$? На самом деле это условие чисто техническое, однако важно, чтобы вектор состояния в конце вычисления был *разложим*, т.е. имел вид $|\xi'\rangle \otimes |\eta'\rangle$ (с

произвольным $|\eta'\rangle$). Если это так, то первая подсистема находится в определённом состоянии $|\xi'\rangle$, поэтому про вторую подсистему (дополнительную память) можно забыть. В противном случае, совместное состояние двух подсистем оказывается «запутанным» (entangled), поэтому первую подсистему нельзя отделить от второй.

«... Нельзя отсоединить квантовый принтер от квантового компьютера, пока идёт работа, иначе один или оба прибора испортятся.»

А. Шень

6. Соотношение между классическим и квантовым вычислением

Классический объект, соответствующий унитарному оператору, — перестановка. Любой перестановке $G\colon \mathbb{B}^k \to \mathbb{B}^k$ естественно сопоставляется унитарный оператор $\widehat{G}$ в пространстве $B^{\otimes k}$, действующий по правилу $\widehat{G}|x\rangle \stackrel{\text{def}}{=} |Gx\rangle$.

Аналогично определению 5.1, можно определить обратимые классические схемы, реализующие перестановки.

Определение 6.1. ***Обратимая классическая схема.*** *Пусть $\mathcal{A}$ — некоторое множество перестановок вида $G\colon \mathbb{B}^k \to \mathbb{B}^k$ (базис). Обратимая классическая схема в базисе $\mathcal{A}$ — это последовательность перестановок $U_1[A_1], \ldots, U_l[A_l]$, где A_j — множества битов, $U_j \in \mathcal{A}$.*

Перестановка, реализуемая обратимой схемой. *Это произведение перестановок $U_l[A_l] \cdot \ldots \cdot U_1[A_1]$.*

Перестановка U, реализуемая схемой в расширенном смысле. *Это такая перестановка, что произведение перестановок*

$$W = U_l[A_i] \cdot \ldots \cdot U_1[A_1]$$

(действующее на N битов, $N \geqslant n$) для любого $x \in \mathbb{B}^n$ удовлетворяет условию $W(x, 0^{N-n}) = \big(Ux, 0^{N-n}\big)$.

В каких случаях функцию, заданную булевой схемой, можно реализовать обратимой схемой? Обратимые схемы реализуют только перестановки. Преодолеть эту трудность можно так. Вместо вычисления функции $F\colon \mathbb{B}^n \to \mathbb{B}^m$ будем вычислять функцию $F_\oplus\colon \mathbb{B}^{n+m} \to \mathbb{B}^{n+m}$, заданную соотношением $F_\oplus(x, y) = (x, y \oplus F(x))$ (здесь $\oplus$ означает побитовое сложение по модулю 2). Тогда значение $F(x)$ можно получить так: $F_\oplus(x, 0) = (x, F(x))$.

Чтобы можно было вычислять функции, заданные булевыми схемами в полном базисе, недостаточно взять базис для обратимых схем из перестановок на двух битах. Оказывается, что любая перестановка на двух битах $g\colon \mathbb{B}^2 \to \mathbb{B}^2$ является линейной функцией (при естественном отождествлении множества $\mathbb{B}$ и поля из двух элементов $\mathbb{F}_2$): $g(x,y) = (ax \oplus by \oplus c, dx \oplus ey \oplus f)$, где $a, b, c, d, e, f \in \mathbb{F}_2$. Поэтому все функции, вычисляемые обратимыми схемами в базисе из перестановок на двух битах, являются линейными.

А вот перестановок на трёх битах уже достаточно, чтобы реализовать любую функцию. При этом не обязательно использовать все перестановки, достаточно включить в базис лишь две функции — отрицание $\neg$ и *элемент Тоффоли* $\wedge_\oplus\colon (x,y,z) \mapsto (x,y,z \oplus xy)$. При этом имеется в виду реализуемость в расширенном смысле, т. е. можно брать напрокат биты в состоянии 0 и возвращать их после окончания вычислений в том же состоянии.

Задача 6.1. Докажите для обратимых схем полноту базиса, состоящего из отрицания и элемента Тоффоли.

Лемма 6.1. *Пусть функция $F\colon \mathbb{B}^n \to \mathbb{B}^m$ реализуется булевой схемой размера L в некотором базисе $\mathcal{A}$. Тогда можно реализовать функцию $(x,0) \mapsto (F(x), G(x))$ обратимой схемой размера $O(L)$ в базисе $\mathcal{A}_\oplus$, состоящем из функций $f_\oplus$ ($f \in \mathcal{A}$), а также функции $\copy\colon (x,y) \mapsto (x, x \oplus y)$.*

Замечание 6.1. Помимо «полезного» ответа $F(x)$ схема, указанная в формулировке леммы, производит «мусор» $G(x)$.

Замечание 6.2. Содержательный смысл операции $\copy$ — обратимое копирование бита (если начальное значение y равно 0). В литературе эта операция обычно называется Controlled NOT по причинам, которые станут ясными из дальнейшего.

Замечание 6.3. Применяя функцию $(\leftrightarrow)\colon (a,b) \mapsto (b,a)$ можно менять биты местами в записи. Обратите внимание, что для перестановок битов достаточно также иметь в базисе $\copy$, так как

$$(\leftrightarrow)[j,k] = \copy[j,k] \copy[k,j] \copy[j,k].$$

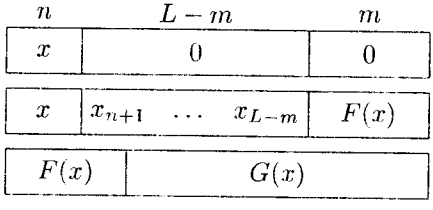
Доказательство. Возьмём схему, вычисляющую F. Пусть входные переменные — это $x_1, \ldots, x_n$. Вспомогательные переменные схемы и биты результата — это $x_{n+1}, \ldots, x_{n+L}$; в обратимой схеме сопоставим им дополнительные биты, имеющие в начальном состоянии значение 0.

Каждое присваивание в схеме имеет вид $x_{n+k} := f_k(x_{j_k}, \ldots, x_{l_k})$, $f_k \in \mathcal{A}$, $j_k, \ldots, l_k < n + k$. В обратимой схеме аналогом присваивания будет действие перестановки $(x_{j_k}, \ldots, x_{l_k}, x_{n+k}) := (f_k)_\oplus(x_{j_k}, \ldots, x_{l_k}, x_{n+k})$, т. е. $x_{n+k} := x_{n+k} \oplus f_k(x_{j_k}, \ldots, x_{l_k})$.

Поскольку начальные значения дополнительных переменных были равны 0, их конечные значения будут такими же, как и в булевой схеме.

Осталось поменять местами биты, чтобы получить указанный в условии формат ответа.

Весь процесс вычисления удобно представить следующей схемой (над прямоугольниками подписано количество битов, внутри — их содержимое):



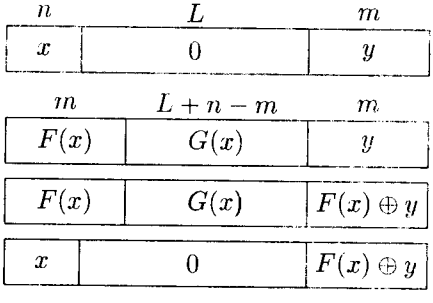
присваивания по схеме

перестановки битов

□

Лемма 6.2 (очистка мусора). *В условиях леммы* 1 *можно произвести вычисление функции $F_\oplus$ обратимой схемой размера $O(L + n + m)$ (с использованием дополнительных битов).*

Доказательство. Для очистки мусора будет использована обратимость. Изобразим процесс вычисления $F_\oplus$ схемой, аналогичной той, что приведена в доказательстве леммы 6.1.



вычисление по схеме из доказательства леммы 6.1

сложение $F(x)$ и y по модулю 2

обращение вычислений, сделанных на первом шаге

□

Замечание 6.4. Обратимыми вычислениями заинтересовались при попытке ответить на вопрос, какая энергия необходима для вычислений (классических). Анализ показал, что потери энергии можно устремить к нулю для всех вычислительных операций, кроме необратимых. Когда

Справедливо следующее утверждение: если $\tilde{U}$ приближает (в расширенном смысле) U с точностью δ, то $\tilde{U}^{-1}$ приближает U^{-1} с той же точностью δ. Это следует из того, что $\|W_1 X W_2\| = \|X\|$ для унитарных операторов W_1, W_2. Умножая выражение под нормой в (7.10) слева на $\tilde{U}^{-1}$, а справа — на U^{-1}, получим следствие из неравенства (7.10): $\|\tilde{U}^{-1}V - VU^{-1}\| \leqslant \delta$.

Определение 7.5. *Будем называть базис $\mathcal{A}$ полным, если любой унитарный оператор U можно с любой точностью представить в расширенном смысле квантовой схемой в базисе $\mathcal{A}$.*

Теорема 7.2 (см. [4]). *Базис $\mathcal{Q} = \{H, K, \Lambda(\sigma^x), \Lambda^2(\sigma^x)\}$, где*

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad K = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},$$

является полным. (Такой базис будем называть **стандартным**.)

Доказательство этой теоремы следует из решения задач 7.5–7.9.

Замечание 7.2. Если убрать из базиса $\mathcal{Q}$ квантовый элемент Тоффоли, он перестает быть полным. Однако многие важные вычисления можно делать и в таком усеченном базисе. В частности, как будет видно в дальнейшем, схемы, исправляющие ошибки, можно реализовать без элемента Тоффоли.

Можно оценить сложность реализации оператора U в этом базисе. Если $U\colon \mathcal{B}^{\otimes n} \to \mathcal{B}^{\otimes n}$, то можно реализовать этот оператор с точностью δ квантовой схемой в базисе $\mathcal{Q}$ размера $L = \exp(O(n)) \cdot \operatorname{poly}(\log(1/\delta))$. Если матричные элементы U заданы в двоичной записи, то эта схема строится по U с помощью некоторого алгоритма примерно за то же время (множители и степени полинома могут отличаться). Идея построения такого алгоритма легко усматривается из задач 7.1 и 7.11.

Задачи

7.1. Докажите, что все операторы на одном q-бите в сочетании с оператором $\Lambda(\sigma^x)$ образуют полный базис. Решение должно быть достаточно эффективным: должен существовать алгоритм, который строит схему, реализующую произвольный оператор U на n q-битах, за время $\exp(O(n)) \cdot \operatorname{poly}(\log(1/\delta))$.

7.2. Докажите свойства операторной нормы (7.6–7.8).

7.3. Пусть операторы $\tilde{U}_k$ приближают в расширенном смысле операторы U_k с точностью δ_k, $1 \leqslant k \leqslant L$. Докажите, что оператор $\tilde{U}_L \cdot \ldots \cdot \tilde{U}_1$

приближает в расширенном смысле оператор $U_L \cdot \ldots \cdot U_1$ с точностью $\sum \delta_k$.

7.4. Пусть оператор $\tilde{U}$ приближает в расширенном смысле оператор U с точностью δ. Докажите, что существует оператор W, точно представляющий U в расширенном смысле, т. е. выполняется равенство

$$W\left(|\xi\rangle \otimes |0^{N-n}\rangle\right) = (U|\xi\rangle) \otimes |0^{N-n}\rangle,$$

и такой, что $\|W - \tilde{U}\| \leqslant O(\delta)$.

7.5. Пусть унитарный оператор $U\colon \mathcal{B}^{\otimes n} \to \mathcal{B}^{\otimes n}$ удовлетворяет условию $U|0\rangle = |0\rangle$. Постройте реализующую $\Lambda(U)$ схему размера $6n+1$ в базисе $Q \cup \{U\}$, использующую оператор U один раз.

7.6. Пусть X, Y — некоммутирующие элементы группы $\mathbf{SO}(3)$ — повороты на углы, несоизмеримые с π. Докажите, что группа, порождённая X и Y, образует всюду плотное подмножество в $\mathbf{SO}(3)$.

7.7. Пусть $\mathcal{M}$ — унитарное пространство размерности $\geqslant 3$. Рассмотрим подгруппу $H \subset \mathbf{U}(\mathcal{M})$ — стабилизатор одномерного подпространства, порождённого некоторым единичным вектором $|\xi\rangle \in \mathcal{M}$. Пусть V — произвольный унитарный оператор, не сохраняющий подпространство $\mathbb{C}(|\xi\rangle)$. Докажите, что множество операторов $H \cup V^{-1}HV$ порождает всю группу $\mathbf{U}(\mathcal{M})$.

(Заметим, что в условии этой задачи $\mathbf{U}(\mathcal{M})$ и H можно профакторизовать по подгруппе фазовых сдвигов $\mathbf{U}(1)$).

7.8. Докажите, что операторы из стандартного базиса порождают всюду плотное множество в $\mathbf{U}(\mathcal{B}^{\otimes 2})/\mathbf{U}(1)$.

7.9. Докажите, что фазовые сдвиги можно реализовать в стандартном базисе, используя напрокат дополнительные q-биты.

7.10. Докажите, что отрицание σ^x и *элемент Дойча* $\Lambda^2(R)$, где $R = -i\exp(\pi i \alpha \sigma^x)$, α — иррациональное, образуют полный базис для квантового вычисления.

7.11. Докажите, что любой оператор U, действующий на одном q-бите, может быть приближённо реализован в расширенном смысле с точностью δ схемой размера $O(\log^3(1/\delta))$ в стандартном базисе, и есть полиномиальный алгоритм построения этой схемы по описанию U.

Эта задача довольно сложна, к её решению лучше приступать после знакомства с разделами 11 и 12 и решения задачи 12.3 (квантовое преобразование Фурье). Предлагаемый путь решения является достаточно изощрённым. В статье [4] был использован более прямой (но тоже неочевидный) подход, при котором получается схема размера $\mathrm{poly}(\log(1/\delta))$.

8. Определение квантового вычисления. Примеры

«...Мы не можем применить здесь здравый смысл, мы можем только стремиться понять внутреннюю логику этого безумия, которая наверняка есть...»

А. Шень

Пока мы описали работу квантового компьютера. Теперь пора определить, когда эта работа приводит к решению интересующей нас задачи. Определение будет похоже на определение вероятностного вычисления.

Пусть есть функция $F\colon \mathbb{B}^n \to \mathbb{B}^m$. Рассмотрим квантовую схему, работающую с N битами: $U = U_L \cdot \ldots \cdot U_2 U_1 \colon \mathcal{B}^{\otimes N} \to \mathcal{B}^{\otimes N}$. Неформально говоря, эта схема вычисляет F, если после применения U к начальному состоянию $|x, 0^{N-n}\rangle$, мы, «посмотрев» на первые m битов, *с большой вероятностью* «увидим» $F(x)$. (Остальные q-биты могут содержать произвольный мусор.)

Нужно только оговорить, что такое эта вероятность. Слова «посмотрев» и «увидим» в точном смысле означают, что производится *измерение* значений соответствующих q-битов. В результате измерения могут получаться разные ответы, каждому соответствует своя вероятность. Ниже (раздел 9) этот вопрос рассматривается подробно. Для того, чтобы дать определение квантового вычисления функции F, достаточно (не вдаваясь в обсуждение физических объяснений этого факта) принять следующее: *вероятность получения базисного состояния x при измерении состояния $|\psi\rangle = \sum_x c_x |x\rangle$ равна*

$$\mathbf{P}(|\psi\rangle, x) = |c_x|^2. \tag{8.1}$$

Нас интересует вероятность того, что компьютер закончит работу в состоянии вида $(F(x), z)$, где z — любое.

Определение 8.1. *Схема $U = U_L \cdot \ldots \cdot U_2 U_1$ вычисляет F, если для любого x выполнено*

$$\sum_z \left|\langle F(x), z| U |x, 0^{N-n}\rangle\right|^2 \geqslant 1 - \varepsilon,$$

где ε — некоторое фиксированное число, меньшее $1/2$. (Обратите внимание, что $F(x)$ и x состоят из разного количества битов, хотя суммарная длина $(F(x), z)$ и $(x, 0^{N-n})$ одинакова и равна N.)

Как и для вероятностных вычислений, выбор ε несущественен, поскольку можно запустить несколько экземпляров схемы независимо и выбрать тот результат, который получается чаще всего. Из оценки,

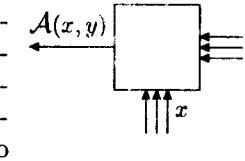
приведённой на с. 36, следует, что для уменьшения вероятности неудачи в N раз нужно взять $O(\log N)$ экземпляров схемы U. Выбор самого частого результата реализуется классической схемой, использующей функцию голосования $MAJ(x_1, \ldots, x_n)$ (она равна 1, когда более половины её аргументов равны 1, и равна 0 в противном случае). Функция $MAJ(x_1, \ldots, x_n)$ реализуется в полном базисе схемой размера $O(n \log n)$, так что потеря эффективности при уменьшении вероятности неудачи в N раз задаётся множителем $O(m \log N \log\log N)$.

Задача 8.1. Докажите, что приведенное рассуждение является корректным в квантовом случае: функция $MAJ_{\oplus}$ реализована в виде обратимой схемы, на вход которой подаются выходные q-биты n копий схемы U.

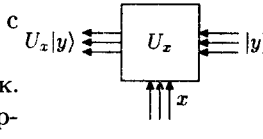
8.1. Квантовый поиск: алгоритм Гровера. Итак, мы имеем определение квантового вычисления. Теперь можно заняться сравнением эффективности классического и квантового вычисления. Во введении упоминались три основных примера, для которых квантовое вычисление оказывается, по-видимому, эффективнее классического. Мы начнём с того из них, в котором квантовое вычисление заведомо эффективнее (хотя ускорение лишь «полиномиальное»).

Дадим определение *универсальной переборной задачи* в классической и квантовой постановке.

Пусть имеется устройство (см. рисунок), которое по входам x и y определяет значение некоторого предиката $\mathcal{A}(x, y)$. Нас интересует предикат $F(x) = \exists y\, \mathcal{A}(x, y)$. Это похоже на определение класса NP, но сейчас нам недоступна внутренняя структура устройства, вычисляющего предикат $\mathcal{A}$. В таких условиях на классическом компьютере значение предиката $F(x)$ нельзя вычислить быстрее, чем за $N = 2^n$ шагов, где n — количество битов в записи y.



Оказывается, что на квантовом компьютере можно вычислить значение предиката $F(x)$ и даже найти y, на котором выполнено $\mathcal{A}(x, y)$, за время $O(\sqrt{N})$. Получены также и нижние оценки, показывающие, что в этой постановке квантовые устройства дают лишь полиномиальное ускорение по сравнению с классическими.



В квантовой постановке задача выглядит так. Вход x по-прежнему классический, но сам «чёрный ящик» — квантовое устройство, и вход y

(варианты ответа) мы будем считать квантовым. Поэтому наш оракул (или «чёрный ящик») задаёт оператор U_x, действующий по правилу

$$U_x|y\rangle = \begin{cases} |y\rangle, & \text{если } \mathcal{A}(x,y) = 0, \\ -|y\rangle, & \text{если } \mathcal{A}(x,y) = 1. \end{cases}$$

Нужно вычислить значение $F(x)$ и найти «ответ» y (при котором выполнен $\mathcal{A}(x,y)$).

Результаты, о которых уже упоминалось, формулируются так (см. [31, 48]): *существуют две константы C_1 и C_2 такие, что есть схема размера $\leqslant C_1\sqrt{N}$, решающая задачу для любого предиката $\mathcal{A}(x,y)$; а для любой схемы размера $\leqslant C_2\sqrt{N}$ существует предикат $\mathcal{A}(x,y)$, при котором задача не решается на этой схеме* (*т. е. схема даёт неправильный ответ с вероятностью $> 1/3$*).

Мы разберём упрощённую постановку: считаем, что «ответ» существует и единствен, обозначим его через y_0; нужно найти y_0. Схема, которую мы для этого построим, будет примером «прямого» квантового вычисления; она будет описана в терминах преобразований базисных векторов.

Рассмотрим два оператора:

$$U = I - 2|y_0\rangle\langle y_0|$$
$$\text{и} \quad V = I - 2|\xi\rangle\langle\xi|, \quad \text{где } |\xi\rangle = \frac{1}{\sqrt{N}}\sum_y |y\rangle.$$

Оператор V в матричной форме может быть записан так (напомним, что $N = 2^n$):

$$V = \begin{pmatrix} 1 - \frac{2}{N} & \dots & -\frac{2}{N} \\ \vdots & \ddots & \vdots \\ -\frac{2}{N} & \dots & 1 - \frac{2}{N} \end{pmatrix}.$$

Оператор U нам задан (это оракул). Построим квантовую схему, вычисляющую V. Действовать будем так: переведем $|\xi\rangle$ в $|0^n\rangle$ некоторым оператором W, затем применим оператор $Y = I - 2|0^n\rangle\langle 0^n|$, после чего применим W^{-1}.

Построить оператор W, который переводит $|\xi\rangle$ в $|0^n\rangle$, просто. Это $W = H^{\otimes n}$, где оператор H — из стандартного базиса (см. с. 64). Действительно, $|\xi\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle)^{\otimes n}$, а $H\colon \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \mapsto |0\rangle$.

Теперь построим реализацию оператора Y. Используем обратимую классическую схему, реализующую оператор $Z\colon \mathbb{B}^{n+1} \to \mathbb{B}^{n+1}$,

$$Z|a_0, \ldots, a_n\rangle = |a_0 \oplus f(a_1, a_2, \ldots, a_n), a_1, \ldots, a_n\rangle;$$

$$f(a_1, \ldots, a_n) = \begin{cases} 1, & \text{если } a_1 = \ldots = a_n = 0, \\ 0, & \text{если } \exists j : a_j \neq 0. \end{cases}$$

(С точностью до перестановки аргументов, $Z = \widehat{f_{\oplus}}$.) Поскольку f имеет малую схемную сложность (в классическом смысле), по лемме 6.2 для вычисления Z существует небольшая схема (в которой «берутся напрокат» дополнительные q-биты).

Схема, реализующая оператор V, изображена на рис. 4. Центральная часть, включающая в себя Z, σ^z и Z, реализует оператор Y. В схеме используется оператор $\sigma^z = K^2$ (K из стандартного базиса).

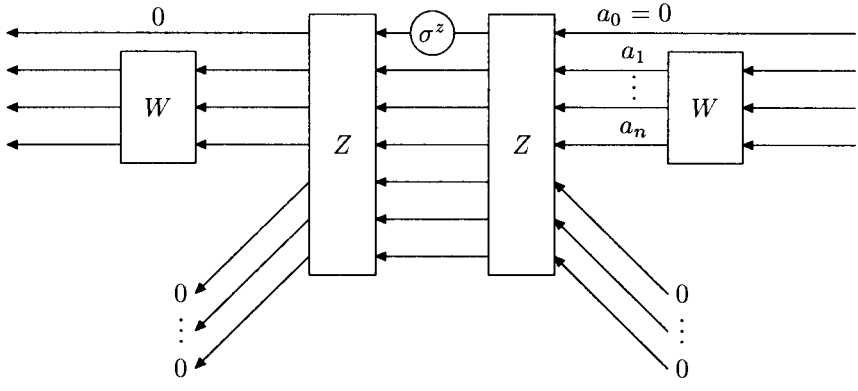


Рис. 4.

Заметим, что W^2 и Z^2 действуют тождественно на векторах с нулевыми значениями q-битов, взятых напрокат. Поэтому решающую роль играет оператор σ^z, действующий на вспомогательный q-бит, который также не меняется после всего вычисления.

Пусть вас не смущает то, что σ^z действует только на «управляемый» q-бит, а меняется в результате весь вектор. Вообще, различие между «чтением» и «записью» в квантовом случае неабсолютно и зависит от выбора базиса. Приведём соответствующий пример.

Напишем матрицу $\Lambda(\sigma^x)\colon |a, b\rangle \mapsto |a, a \oplus b\rangle$ в базисе $\frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ для каждого из q-бит. Другими словами, запишем матрицу для оператора $X = (H \otimes H)\ \Lambda(\sigma^x)\ (H \otimes H)$. Схема для этого оператора изображена