

Московский физико-технический институт
(государственный университет)

Современные вычислительные парадигмы

Составитель: аспирант Фомин С.А. (fomin@ispras.ru)

Преподаватель: Печенкин А.А.

Москва

6 сентября 2001 г.

Содержание

1	Введение	2
2	Молекулярные компьютеры	5
2.1	Введение	5
2.2	Устройство	7
2.3	Эксперимент Адельмана	8
2.4	Бинарные задачи	10
2.5	Взламывание алгоритма шифрования DES	11
2.6	Стиkerы	12
2.7	Преимущества и недостатки	13
2.7.1	Рекомендуемая литература для дальнейшего изучения	14
3	Квантовые компьютеры	14
3.1	Введение	14
3.2	Квантовая механика	18
3.2.1	Поляризация фотонов	18
3.2.2	Пространства состояний и обозначения векторов состояний и сопряженных векторов(Bra/Ket Notation).	21
3.3	Квантовые биты	22
3.3.1	Наборы кубитов	22
3.3.2	Измерение	24
3.3.3	EPR парадокс	25
3.4	Квантовые вентили (Quantum Gates)	26
3.4.1	Элементарные квантовые вентили (Simple Quantum Gates)	27
3.4.2	Примеры	29
3.4.3	Квантовый компьютер	32
3.5	Алгоритм Шора (Shor)	36
3.5.1	Квантовое преобразование Фурье	37
3.5.2	Более подробный конспект алгоритма Шора	38
3.6	Задачи поиска	40
3.6.1	Алгоритм поиска Гровера	41
3.6.2	Инверсия над средним значением	43
3.6.3	Изменение знака	44
3.6.4	Структурированный поиск	44
3.7	Квантовая коррекция ошибок (Quantum Error Correction)	48
3.7.1	Характеризация ошибок	48
3.7.2	Восстановление квантового состояния	49
3.7.3	Пример коррекции ошибок	49
3.8	Заключение	51
3.8.1	Рекомендуемая литература для дальнейшего изучения	52
4	Благодарности	53
A	Тензорное произведение	56

1 Введение

В современном высокообразованном обществе на интуитивном уровне даже ребенок знает что такое "считать", подросток — что такое "функция" и "вычислять функцию". Окруженные множеством разнообразных компьютеров мы зачастую даже на задумываемся над тем как это происходит, какие функции могут вычислять компьютеры, и вообще, какими объектами оперируют современные вычислители? Например широко распространено заблуждение, обусловленное курсом алгебры в системе среднего образования, что компьютеры оперируют непосредственно числами, если уж не действительными, то по крайней мере рациональными. Часто считают, что если функция более или менее формализована, то вычислимость ее подразумевается как само собой разумеющееся.

Проблема состоит в том, что в отличие от стандартных математических парадигм, оперирующих такими умозрительными понятиями как "множество", "действительное число", "отношение", "отображение", понятиями о *реальности* или *номинальности* которых можно спорить, но эволюция которых практически совершенно не зависит от технического уровня цивилизации, парадигмы вычислимости непосредственно зависят от наличия в техническом арсенале цивилизации соответствующих устройств.

Все теоретические вопросы, возникающие при анализе вычислимости определенных математических функций, или разрешимости каких-либо так или иначе формализованных задач решаются на основе математического анализа поведения *моделей* этих самых вычислителей. Соответственно эволюция вычислительных парадигм, разрешимость математических задач и их характеристика по различным классам вычислительной сложности обуславливается именно эволюцией моделей вычислителей. Заметим, что некоторые из моделей вычислителей могут на определенном этапе не иметь технического воплощения, но затем могут появиться устройства проявляющие все или некоторые из свойств этих моделей.

Итак, что из себя представляют современные модели вычислителей и чем они оперируют. Исторически сложилось, что в рамках человеческого общества информация циркулировала в рамках различных знако-символьных систем (более того до сих пор не ясно можно ли говорить об информации и обществе в других контекстах). Соответственно любая информация, существующая в обществе, представлялась в виде различных *символьных строк* (мы используем термин *символьная строка* в несколько более широком контексте, чем обычная печатная строка, подразумеваемая произвольную последовательность символов в рамках определенного *алфавита*). Это означает, что любое устройство работающее с информацией, на самом деле работает с ее символьным представлением, а именно с символьными строками. Действительно, наши компьютеры, работают не с "настоящими" действительными или рациональными числами, а с их конечными символьными представлениями, которые, в силу своей конечности, не могут полноценно представлять, ни множество действительных чисел, ни множество рациональных. Опытные специалисты

по алгоритмам в области вычислительных методов всегда проводят анализ ошибок округления, помня, что стандартные модули арифметики с плавающей запятой производят операции над представлением рациональных чисел с фиксированным числом символов, т.е. производят операции с неким конечным набором меток, очень неравномерно распределенных на некоем ограниченном интервале значений. Что например будет, если честно спросить у компьютера, сколько будет $5.1 - 2.4^0? 2.7?$ Вот честно полученный ответ:

2.699999999999999733546474089962430298328399658203125

Соответственно как работа со строками реализованы и все операции над целыми числами. Итак, теперь понятно, что современные компьютеры представляют собой строковые преобразователи, и мы обратимся к вычислительным моделям, которые появились еще до первого современного компьютера, но которые до сих пор используются при анализе алгоритмов.

Трудность задачи в комбинаторике определяется временем, необходимым для ее решения на детерминированной и недетерминированной машинах Тьюринга (ДМТ и НДМТ). Эти воображаемые устройства принципиально отличаются друг от друга. ДМТ исполняет детерминированные алгоритмы: в них все операции выполняются последовательно и предопределены заранее. Примером может служить сложение векторов. НДМТ, напротив, исполняет недетерминированные алгоритмы. В них имеются узловые точки, где производится выбор направления дальнейших действий из некоторого множества возможных вариантов. Рассматриваемая в теории сложности НДМТ может создавать в каждой узловой точке новые копии самой себя в количестве, равном числу ветвлений алгоритма. Дальнейшие вычисления производятся этими копиями параллельно. Забегая вперед, скажу, что в молекулярных вычислениях используется самый простой вариант этой машины — вначале создается исчерпывающее множество всех возможных решений (включая неверные), затем система их проверяет в режиме параллельных вычислений, отбирая лишь те, которые действительно решают поставленную задачу. На основании того, известен или нет для данной задачи эффективный алгоритм для ДМТ и НДМТ, выделяют классы сложности P, NP, NP-complete (NPC). К типу P (polynomial) относятся задачи, решаемые на ДМТ за полиномиальное время, т.е. за время порядка pn , где n определяет масштаб задачи (скажем, длину ключа шифрования или количество вершин графа), а s — некоторое число, характеризующее алгоритм. К NP (nondeterministic polynomial) относятся задачи, которые решаются за экспоненциальное время на ДМТ (т.е. за время порядка sn) и за полиномиальное время на НДМТ. Такой, например, является задача разложения числа на простые множители, на трудности которой базируется алгоритм RSA. Естественно, что любая задача, относящаяся к P, относится и к NP (рис. 2), ведь каждый детерминированный алгоритм можно рассматривать как недетерминированный, но с одним вариантом выбора в том или ином узле. Поэтому НДМТ может решить за полиномиальное время задачу, на которую ДМТ потребуются экспоненциальное время. Заметим, что оценка сложности задач — вещь непостоянная. Она основана на текущем НЕЗНАНИИ эффективного алгоритма решения проблем на детерминированной машине.

⁰Пример взят из книги Рендала Шварца и Тома Кристиансена "Изучаем Perl".

Не факт, что такие алгоритмы не появятся в будущем. Иначе говоря, классификация задач во многом — вопрос веры. Математически не доказано, что множество $NP \setminus P$ не пусто, т. е. что существует хотя бы одна проблема, входящая в NP и не входящая в P . Математик Стивен Кук показал в 1971 году, что в NP существует подмножество задач, обладающих интересным свойством: если находится детерминированный полиномиальный алгоритм для одной из них, то должны существовать аналогичные алгоритмы для всех задач NP , т. е. $P = NP$. Кук назвал этот класс NP -complete (NPC). Все входящие в него задачи эквивалентны с вычислительной точки зрения, и можно показать, что за полиномиальное время любая проблема из NP сводится к NPC-проблеме. NPC-задачи считаются наиболее сложными в NP , и математики верят, что для этих задач полиномиальных детерминированных алгоритмов уж точно не существует. К NPC-задачам относятся, например, проблемы отыскания Гамильтонова пути (иначе — задачи коммивояжера) и ”собираания набора рюкзаков” (knapsack problem, KP). На KP основан криптографический алгоритм Меркли и Хелмана (Merkle and Hellman) и ряд других шифров. Практическая же важность проблем NP в том, что на детерминированной машине они перестают быть физически решаемыми при n гораздо меньшем, чем у задач класса P . Этот принцип широко используется в современной криптографии. Между тем новые типы вычислительных машин — квантовые и молекулярные компьютеры — являются ограниченными реализациями НДМТ и теоретически способны решать NP -проблемы за полиномиальное время.

2 Молекулярные компьютеры

2.1 Введение

Биомолекулярные вычисления или молекулярные компьютеры или даже ДНК- или РНК- вычисления — все эти термины¹ появились совсем недавно, на стыке таких различных наук как молекулярная генетика и *Computer Science*. Настоящий взрыв интереса к этой постановке произошел в 1994 году, когда Леонарду Адельману *Adleman*² [48] удалось практически решить задачу коммивояжера *travelling salesperson problem* для семи городов. Как известно, задача эта NP -полная, и при ее решении на обычном однопроцессорном компьютере требует времени экспоненциального от размера задачи. Разумеется, во многих случаях наращивание мощности компьютера путем масштабирования числа процессоров может некоторым образом уменьшить время выполнения, хотя заметим, что даже и это возможно далеко не всегда. Мы здесь опустим обсуждение моделей параллельных вычислителей (*PRAM*) — *EREW*, *CRCW*, *ERCW* Заметим лишь, что даже если задача такова, что увеличение числа процессоров приводит к пропорциональному увеличению вычислительной скорости, то такие факторы как ”узкое горло фон Неймана”, т.е. стандартная вычислительная архитектура, где процессор конструктивно отделен от

¹В западной литературе применяются термины *Biomolecular Computation*, *BMC*, *DNA computation*, *RNA*.

²Это тот самый Adleman — ”А” из знаменитого криптоалгоритма RSA

обрабатываемых данных и сообщается с ними с помощью фиксированных каналов, и технологические ограничения применяемой сейчас технологии полупроводников, несмотря на достигнутые успехи в миниатюризации, очень сильно ограничивают размерности задач, скорость решения которых казалось бы мы могли бы ускорить с помощью многопроцессорного параллелизма. Итак, основные проблемы (возможно в терминах скорее технологических, чем в понятиях теории сложности) заключаются в следующем:

- Макроскопические, относительно молекулярных размеров, размеры всех конструктивных элементов.
- И соответственно большое потребление энергии. (в расчете, например на одну операцию).
- Конструктивные сложности при масштабировании системы. (т.к. система существенно неоднородна)

Если бы удалось решить эти проблемы, то хотя в рамках классической теории сложности это бы не изменило существующую карту классов сложности, то с практической точки зрения, возможно удалось бы решать большинство из задач, возникающих в реальной жизни. Сразу же приведем преимущества, представляемые ДНК-вычислителями:

- Дешевая масштабируемость при решении параллелизуемых NP задач (*NP search problems*). Первые же опыт Адельмана имел дело именно с такой задачей.
- Потенциально огромная память. В то время как кремниевые полупроводниковые микросхемы уже исчерпали возможности для миниатюризации, то даже небольшой объем молекулярного вычислителя может содержать огромное число молекул. Приведем несколько цифр из отчета [47]:

Слабый раствор ДНК в одном литре воды может кодировать от 10^7 до 10^8 терабайт информации, причем может не только предоставлять ограниченный доступ по шине данных, как в это делает модуль памяти в компьютерах со стандартной архитектурой, но и осуществлять массивно-параллельные ассоциативные поиски.

- Генетические компьютеры способны выполнять триллионы операций в секунду.

2.2 Устройство

В чем же состоит суть биомолекулярных вычислений? На самом деле биомолекулярные вычисления — суть агрегирующее название для множества совершенно различных методов так или иначе связанных с ДНК или РНК. Основная идея состоит в том, что при ДНК-вычислениях данные представляются не в форме единиц и нулей, а в виде молекулярной структуры, построенной на основе спирали ДНК.

Роль программного обеспечения для чтения, копирования и управления данными выполняют особые ферменты.

Так например в опыте Адельмана, описанном в журнале Science в 1994 году, вычислительная задача была решена чисто путем синтеза определенного количества ДНК в тестовой пробирке, где в качестве вычислительных символов были использованы строительные блоки ДНК. Концептуально данная техника³ опирается на использование нитей генетического кода ДНК в качестве замены кодов компьютерной программы.

Фундаментом всей системы хранения биологической информации, а стало быть, и ДНК-компьютеров, является способность атомов водорода, входящих в азотистые соединения аденин (Adenine, A), тимин (Thymine, T), цитозин (Cytosine, C) и гуанин (Guanine, G), при определенных условиях притягиваться друг к другу, образуя нехимически (т. е. невалентно) связанные пары $A = T$ и $C = G$. С другой стороны, эти вещества (или, как еще говорят, основания) могут валентно связываться с фосфатно-дезоксирибозными группами — комбинациями молекулы сахара (дезоксирибозы) и фосфата, образуя так называемые нуклеотиды. Нуклеотиды, в свою очередь, легко образуют полимеры длиной в десятки миллионов оснований. В этих супермолекулах фосфат и дезоксирибоза играют роль поддерживающей структуры (они чередуются в цепочке), а азотистые соединения кодируют информацию.

Молекула получается направленной — она начинается с фосфатной группы и заканчивается дезоксирибозой. Длинные цепочки ДНК называют нитями (*strands*), а короткие — олигонуклеотидами. Каждой молекуле ДНК соответствует еще одна ДНК — так называемое дополнение Ватсона-Крика (*Watson-Crick complementary*). Она имеет противоположную направленность, нежели оригинальная молекула, и получается из последней заменой оснований A, T, C, G на парные к ним. В результате притяжения аденина к тимину и цитозина к гуанину получается знаменитая двойная спираль, обеспечивающая возможность удвоения числа ДНК при размножении клетки. Задача удвоения решается с помощью специального белка (энзима) — полимераза. Этот энзим скользит вдоль ДНК и синтезирует на ее основе новую молекулу, в которой все основания заменены на соответствующие парные. Интересно, что синтез начинается только в том случае, если к ДНК прикреплен коротенький кусочек ее дополнения (или "зацепки" — *primer*). Данное свойство активно используется в молекулярной биологии и молекулярных вычислениях. По сути своей полимеразы — это реализация машины Тьюринга (МТ), которая состоит из двух лент и программируемого пульта управления. Пульт считывает данные с одной ленты, обрабатывает их по некоторому алгоритму и записывает их на другую ленту. Полимераза также последовательно считывает исходные данные с одной ленты (ДНК) и на их основе формирует ленту с результатом вычислений (дополнение Ватсона-Крика). С теоретической точки зрения, запрограммировав пульт, можно заставить МТ делать что угодно — играть в шашки, строить прогноз экономического развития Зимбабве до 3071 года или подсчитывать декалитры выпитой водки. Природа, правда, снабдила нас устройствами, решающими ограниченный круг задач, но именно подобие полимеразы и МТ навело Адельмана на мысль о возможности построения вычислительных систем на базе ДНК.

³В англоязычной литературе, совокупность подобных методов называется DNA Hybridization

В живых организмах полно других аналогов машины Тьюринга. Например, энзим транскриптаза считывает ДНК и синтезирует на ее основе молекулу рибонуклеиновой кислоты (РНК⁴). Другой энзим — обратная транскриптаза — транслирует РНК в ДНК. Именно он является основой жизнедеятельности самых зловредных вирусов — ретровирусов (к ним относятся вирусы гриппа и ВИЧ), которые не имеют собственной ДНК (только РНК) и при помощи этого приводят свое описание в "формат", подходящий для включения в ДНК клетки. Затем вирус запускает механизм размножения клетки, размножающий теперь и его самого. В качестве долговременного хранилища информации РНК по причине низкой надежности в организмах используются редко. В процессе считывания данных из ДНК в РНК (трансляции) получается так называемая "РНК с сообщением" (*message RNA*), передаваемая на вход рибосомы (синтезатора белков из аминокислот) — еще одного аналога машины Тьюринга.

Белок в рибосоме играет роль выходной ленты — для его конструирования обычно требуются тысячи аминокислот, каждая из которых кодируется последовательностью из трех оснований. Сами аминокислоты рибосома различает по присоединенным к ним "биркам", или "транспортным" РНК, также состоящим из трех оснований. Необходимость трансляции объясняется тем, что читать напрямую из ДНК информацию, нужную для синтеза белка, опасно, так как вероятность повредить эту молекулу при подобной часто повторяющейся операции весьма велика. Вся идея синтеза РНК из ДНК — классический пример кэширования.

2.3 Эксперимент Адельмана

Для доказательства возможности вычислений на базе ДНК Адельман выбрал проблему отыскания Гамильтонова пути графа [48]. Суть ее в следующем. Имеется граф с n вершинами, соединенными однонаправленными ребрами. Нужно найти путь из одной заданной вершины в другую, проходящий через все остальные вершины только один раз, или доказать отсутствие такого пути. Задача дискретная, решать ее можно лишь перебором, она имеет прикладное значение (к ней можно свести задачу разложения числа на множители), и все существующие для нее детерминированные алгоритмы имеют экспоненциальное время исполнения. Для решения задачи Адельман воспользовался следующим недетерминированным алгоритмом:

1. Сгенерировать все возможные варианты путей через граф.
2. Исключить все пути, которые не проходят через заданные начальную и конечные вершины.
3. Исключить те пути, что проходят через число вершин, отличное от n .
4. Исключить все пути, которые проходят через какую-либо вершину по несколько раз.
5. Если после этого хотя бы один путь остался, то это именно тот, что нужен.

⁴РНК отличается тем, что в ней дезоксирибоза заменена на другой сахар (рибозу), а тимин — на урацил (Uracil, U).

Если после этого хотя бы один путь остался, то это именно тот, что нужен. Весь фокус в том, как заставить алгоритм работать. Даже для небольшого графа число возможных вариантов пути оказывается колоссальным! И здесь на помощь приходят молекулы. В основе рассуждений Адельмана лежала возможность закодировать каждую вершину графа уникальной последовательностью олигонуклеотидов **A, T, C, G**, например вершине O_2 поставить в соответствие последовательность **TATCGGATCGGTATATCCGA**, O_3 — **GCTATTCGAGCTTAAAGCTA**, O_4 — **GGCTAGGTACCAGCATGCTT** и т. п. (в первоначальном эксперименте использовалась 20-разрядная кодировка). Тогда путь от одной вершины к другой можно определить как вектор из 10 последних оснований начальной вершины и 10 первых оснований конечной вершины, т. е. $O_2O_3 = \mathbf{GTATATCCGAGCTATTCGAG}$, $O_2O_3 = \mathbf{CTTAAAGCTAGGCTAGGTAC}$ и т. д. Как будет видно в дальнейшем, нужны еще дополнения Ватсона-Крика к молекулам O_i . Мы будем обозначать их \hat{O}_i . Скажем, $\hat{O}_2 = \mathbf{CGATAAGCTCGAATTCGAT}$. Синтезировать все эти ДНК на современной биомолекулярной аппаратуре проще простого. Есть даже коммерческие фирмы, которым достаточно послать написанную на бумажке последовательность оснований, а на следующий день они уже пришлют пробирки с синтезированными молекулами.

Шаг 1. Для того чтобы алгоритм заработал, одних ДНК мало. Нужно прибегнуть к другим химикатам, изобретенным живой природой. Одним из них является лигаза, которая валентно склеивает разрывы в двойной спирали ДНК. Если смешать раствор, содержащий ДНК, кодирующие ребра графа, с раствором, содержащим дополнения Ватсона-Крика для ДНК, кодирующих вершины, то за счет взаимного притяжения олигонуклеотидов в парах **A-T** и **G-C** 10-разрядное начало молекулы O_2O_3 притянется (с образованием двойной спирали) к концу молекулы \hat{O}_2 , а конец O_2O_3 — к началу \hat{O}_3 . Лигаза же найдет разрыв между \hat{O}_2 и \hat{O}_3 , склеит его, объединив в результате две разных молекулы в одну. То же самое произойдет и с другими ДНК, и если их достаточно много, то мы получим множество длинных двойных спиралей, кодирующих все возможные пути в Гамильтоновом графе! (По крайней мере, вероятность этого достаточно высока — молекул может быть миллиарды миллиардов штук.) Среди них есть и один путь, являющийся решением рассматриваемой проблемы. Но как его выделить среди массы ненужных путей?

Шаг 2. Найти иголку в стоге сена непросто, но если иголка вдруг начнет размножаться, то одинаковых иголок вскоре станет больше, чем сена. Именно эта идея и используется на втором шаге алгоритма. В молекулярной биологии для увеличения числа ДНК применяется метод, называемый *Polymeraza Chain Reaction* (PCR, цепочечные реакции полимеразы; про полимеразу см. выше). При использовании этого метода раствор, содержащий двойные спирали ДНК, полимеразу, нуклеотиды **A, T, G, C** и молекулы "зацепок", попеременно нагревается и охлаждается. При нагревании каждая двойная спираль распадается на две ДНК, а при охлаждении эти ДНК сначала рекомбинируют с "зацепками", а затем рекомбинировавшие спирали восстанавливаются полностью при помощи полимеразы. Таким образом, правильно подбирая концентрацию веществ, можно добиться удвоения числа нужных молекул ДНК за один цикл нагревания/охлаждения. В нашем случае "зацепками" нужно сделать начальную и конечную вершину — O_0 и \hat{O}_6 . Тогда в PCR-циклах молекулы, содержащие начальные и конечные вершины (в том числе отражающие решение

задачи), будут размножаться экспоненциально быстро, содержащие только одну из этих вершин — линейно, а остальные не будут размножаться вовсе. Иголок станет вскоре больше сена!

Шаг 3. На этом этапе молекулы фильтруются по длине: нужно оставить только ДНК длиной ровно 140 оснований. Делается это при помощи электрогелевого метода, аналогов которому в живой природе нет: ДНК помещаются в гелевый раствор и к нему прикладывается электрическое поле. Молекулы разной длины по-разному поляризуются и ускоряются в электрическом поле. В итоге одни молекулы движутся быстрее других, и через некоторое время ДНК разной длины можно разделить даже визуально — участки геля с ними видны как темные полосы. Аппаратура же позволяет разделять их с точностью до одного основания. Повторяя несколько раз шаги 2 и 3, можно получить раствор, содержащий только молекулы, обладающие нужными свойствами.

Шаг 4. Как гарантировать, что граф проходит через все вершины? Оказывается, это можно сделать с помощью магнита. Методы молекулярной биологии позволяют прицепить к молекуле ДНК крошечный металлический шарик. Если такой молекулой является \hat{O}_2 , то будучи добавлена в раствор, содержащий решение проблемы коммивояжера, она образует двойную спираль с какой-нибудь ДНК, кодирующей путь, проходящий через вершину 2. Если затем приложить к стенке сосуда с раствором магнит, то данная двойная спираль через некоторое время приплывет к этой стенке. Когда в раствор добавлено много молекул с железными шариками, то выход нужных ДНК будет большим. Если произвести подобную фильтрацию последовательно для всех вершин, то останутся только ДНК, содержащие решение проблемы!

Шаг 5. Победа!

Осталось только размножить результат PCR-методом и определить последовательность оснований с помощью типовой машины секвенирования (*sequencing machine*), используемой в молекулярной биологии. Если ни одной молекулы не нашлось, значит с большой долей вероятности можно утверждать, что для данного графа Гамильтонова пути не существует. На выполнение всех этих операций у Адельмана ушло в 1994 г. семь рабочих дней. Самым трудоемким оказалось разделение молекул с помощью магнита. Вскоре другие авторы показали, как, используя аналогичный алгоритм, решать задачу Гамильтона для ребер, обладающих весом, — для этого вес нужно задавать целым числом и кодировать его в ДНК, повторяя нужное число раз последовательность, отображающую исходящую вершину ребра.

2.4 Бинарные задачи

Что еще можно делать с помощью ДНК? Ограничено ли поле деятельности таких машин комбинаторными проблемами? Оказывается, нет. Вскоре после опубликования работы Адельмана разные группы начали исследования в области решения логических задач. В 1995 г. Ричард Липтон из Принстонского университета показал [49], как, используя ДНК, кодировать двоичные числа и решать проблему удовлетворения логического выражения (SAT). Суть этой проблемы состоит в следующем. Пусть имеется некоторое логическое выражение $f(x_1, x_2, \dots, x_n)$. Какие значения нужно присвоить входящим в него логическим переменным x_i , чтобы f давало ис-

тину? Вообще говоря, задачу можно решить только перебором $2n$ комбинаций. И с помощью ДНК легко закодировать их все. Для этого нужно построить граф, описывающий операцию присваивания значений переменным. В нем вершины отображают единичные и нулевые значения x_i , некоторые промежуточные переменные, а пути описывают присваивание. Вершины и ребра этого графа можно представить отрезками ДНК так же, как это делалось в методе Адельмана. Перемешивание всех этих олигонуклеотидов даст раствор, содержащий ДНК, кодирующие все возможные комбинации входных параметров. Логические операции сводятся к извлечению ДНК, содержащих нужные биты в нужном месте, т. е. к нахождению пути, проходящего через конкретную вершину графа (все как в задаче Гамильтона!).

2.5 Взламывание алгоритма шифрования DES

Ни один другой шифр не привлекал к себе так много криптоаналитиков, как американский криптографический стандарт Data Encryption Standard (DES). В принципе эта проблема решаема за приемлемое время и с помощью обычных технологий, но требует концентрации вычислительной мощности десятков тысяч компьютеров. Ученые, занимающиеся исследованием ДНК-машин, просто не могли обойти DES стороной. Массовая параллельность вычислений — как раз то, что требуется для прямой атаки на этот алгоритм. Сразу несколько групп исследователей предложили свои варианты алгоритмов для нахождения ключа DES по известной паре исходного и зашифрованного текста. Первой оказалась группа Липтона [50]. Ее подход базировался на утверждении, что DES — это по сути тоже логическая схема. В частности, исходные данные нужно инициализировать так же, как в общем случае — рабочая строка алгоритма представляет собой последовательность блоков $S_0, B_0, S_1, B_1, \dots, S_{56}$, где B_i — биты ключа, а S_i — служебные последовательности. Далее, используя дополнения к олигонуклеотидам, кодирующим S_{56} , и лигазу, в конец этой ДНК можно добавить любые данные. Алгоритм предусматривает хранение в этой области всех промежуточных результатов вычислений. С помощью дополнительных к S_0 "зацепок" можно размножить нужные молекулы методом PCR. Оригинальным в данном методе является использование природных энзимов для разрезания ДНК (restriction enzymes). Эти белки сканируют ДНК и, если обнаруживают некоторую ключевую последовательность длиной в несколько оснований, режут молекулу пополам. Они используются бактериями для борьбы с вирусами. В данном случае эти энзимы применяются для удаления рабочей информации с конца молекулы. В целом алгоритм группы Липтона довольно рутинный, и мы не будем на нем останавливаться. Большинство предусмотренных им действий сводится к описанной выше операции извлечения данных. В схеме DES, правда, кроме оператора **XOR** и сдвигов используется еще и поиск по таблицам размера 4×16 элементов, но его легко реализовать с помощью битовой выборки — просто выборок оказывается много.

Главное, что задача взламывания DES свелась к следующей: при фиксированном исходном тексте M и полном наборе ключей K_i построить множество из элементов шифрованного текста C_i , таких, где $C_i = DES(M, K_i)$. Так как K_i и C_i будут объединены в итоге на одной ДНК, мы сможем с помощью серии операций побитовой выемки определить по C_i и ключ. Интересно также, что при фиксированном сообщении M раствор с парами (K_i, C_i) достаточно сгенерировать только один

раз. Использовать же его можно многократно — подсуньте разок клиенту нужную строку для зашифровки, и все его записи окажутся в ваших руках! Авторы работы подсчитали, что для достижения результата потребуется осуществить около 900 операций. При современном уровне технологий на это уйдет около 4 месяцев.

2.6 Стикеры

Чрезвычайно оригинальная идея была предложена в 1996 г. в работе ряда авторов из Калифорнийского университета и Университета Южной Калифорнии [51]. Она касается того, как построить ДНК-компьютер, не требующий ни полимеразы, ни лигазы, ни энзимов рестрикции — т. е. "многоразовый". Более того, в таком компьютере можно будет использовать ДНК-память с произвольным доступом. Суть идеи очень проста — кодировать нулевые и единичные состояния бита нужно не разными олигонуклеотидами, а присоединением и отсоединением дополнений Ватсона-Крика к ним.

Иначе говоря, надо сделать следующее:

1. Задать длину отрезка ДНК, представляющей 1 бит (скажем, M оснований).
2. Если объем памяти в цепочке будет N бит, то построить нужно цепочку олигонуклеотидов длиной $N * M$ оснований, такую, чтобы дополнение к любому ее отрезку длиной M смогло образовать двойную спираль только с этим отрезком (т. е. число несовпадений с любым другим участком было бы велико).
3. Тогда эту цепочку ДНК можно логически разделить на отрезки длиной M и назвать их битами.

Если к данному отрезку присоединено его дополнение Ватсона-Крика (так называемый стикер), то бит хранит единицу, в противном случае — ноль. Последовательность олигонуклеотидов, кодирующая отрезок, является адресом в этом блоке памяти. Все операции осуществляются с наборами идентичных ДНК — трубками. С трубками можно делать типовые операции — объединять их, извлекать из них ДНК на основе равенства какого-либо бита нулю или единице. Кроме того, можно добавлять стикеры так, чтобы устанавливать конкретные биты в состояние "1". Операция обнуления всего массива осуществляется нагреванием. Отсоединение отдельных стикеров требует более хитрых действий — использования олигонуклеотидов, идентичных стикерам, но имеющих другую опорную структуру (PNA), что придает им способность образовывать с ДНК тройную спираль (!), но в то же время такая спираль оказывается нестабильной и при нагревании разрушается гораздо быстрее, чем двойная, а стало быть, будут удалены только нужные стикеры. В одном из продолжений своей работы авторы показали, как с помощью этого метода найти ключ алгоритма DES по паре исходный текст–зашифрованный текст.

Вообще говоря, здесь стоит остановиться. В последние годы область быстро развивалась и были предложены методы, как, используя ДНК, моделировать конечные автоматы, осуществлять операции сложения и умножения, перемножать матрицы, вскрывать (по крайней мере теоретически) стойкие шифры типа RC5 и т. п.

2.7 Преимущества и недостатки

Напоследок хотелось бы поговорить о перспективах биокомпьютера. Главное преимущество, которое дает ДНК-компьютер, — это беспрецедентная параллельность вычислений. Производительность отдельной ДНК, оцениваемая в 0,001 операций в секунду, выглядит до безобразия жалкой по сравнению с производительностью обычных ПК, но общая производительность молекул, содержащихся в литре раствора, окажется свыше 10^{14} операций в секунду. Самые мощные на сегодня компьютеры имеют скорость порядка 10^{12} операций в секунду, но это огромные шкафы с тысячами процессоров, а молекулярный компьютер можно (теоретически) разместить на столе. При этом ДНК-память обеспечит хранение данных с плотностью до 1 бит/нм³, в то время как современные магнитные ленты работают с плотностями чуть более 10 — 12 бит/нм³. Сам же ДНК-компьютер будет способен совершать порядка 2×10^{19} необратимых операций на джоуль израсходованной энергии, вплотную приближаясь к теоретическому порогу в $2,4 \times 10^{20}$ оп./Дж, диктуемому соображениями термодинамики. Кремневые системы расходуют на одну операцию в 109 раз больше энергии. Но жизнь не была бы столь сложной, если бы такие красивые идеи легко реализовались на практике. Создать готовый биокомпьютер пока никому не удалось. Было много теоретических построений (типа вскрытия DES), но реально проведено лишь несколько экспериментов, в которых решались относительно простые (с точки зрения современной вычислительной техники) задачи. Можно выделить несколько проблем, с которыми столкнулись ученые, пытаясь построить биокомпьютер. Основная — это сложность и трудоемкость всех совершаемых операций. По идее, их можно автоматизировать, но это пока сделано лишь частично. Например, остра проблема считывания результата — современные способы секвенирования далеки от совершенства: скажем, нельзя за один раз секвенировать цепочки длиной хотя бы в несколько тысяч оснований. Кроме того, это весьма дорогостоящая операция. Вторая проблема — ошибки в вычислениях. Для биологов точность в 1% при синтезе и секвенировании оснований считается очень хорошей. Для вычислений же она абсолютно неприемлема. На других этапах — при РСР-усилении, разрезании ДНК энзимами — также не исключено появление ошибок. Решения задачи могут теряться во время операции битовой выемки (молекулы просто прилипают к стенкам сосудов), нет гарантии, что не возникнут точечные мутации в ДНК, и т. д. Число ошибок экспоненциально растет с числом шагов алгоритма, и весьма возможно, что в конце экспериментатор получит раствор, несколько не похожий на тот, что должен содержать решение. Проблеме ошибок учеными уделяется большое внимание. Например, Липтон и его коллеги показали, как за счет некоторого увеличения времени работы и объема используемого материала можно изменить вычислительный цикл, чтобы вероятность ошибок была минимальной. Другие группы предлагают использовать не трехмерные, а двумерные ДНК-структуры, где олигонуклеотиды прикрепляются к стеклянной подложке. Кроме того, биокомпьютер отличается и еще одним неприятным свойством: составляющие его ДНК имеют тенденцию распадаться с течением времени. Иначе говоря, результаты вычислений тают на глазах! Для борьбы с этим явлением некоторые авторы предлагают использовать специальные белковые взвеси, в которые и помещать ДНК. Также в некоторых работах оспаривается сама возможность масштабирования

ния всей системы уровня, пригодного для решения действительно сложных задач.

Все эти примеры показывают, насколько биокомпьютер пока далек от понятия "практически полезная вещь". Стремительное развитие технологий может существенно изменить ситуацию, но насколько верен базовый подход? Сейчас он состоит в моделировании алгоритмов на макроуровне такими операциями, как сливание и разливание сосудов. Но более перспективным было бы создавать наномашин, похожие на природные ферменты, но формирующие результат по другим законам. Скажем (фантазируя по максимуму), можно вообразить фермент, который на базе существующей ДНК, содержащей ключ DES, синтезирует другую, но уже шифрующую сообщение. Это безумно сложно (что касается DES, то вряд ли вообще возможно), но это то, к чему рано или поздно придет нанотехнология. Вполне вероятно, что от макроопераций не удастся избавиться навсегда, но их число будет многократно уменьшено. В общем, станет ли биокомпьютер реальностью или не выдержит конкуренции с другими нарождающимися технологиями (например, квантовыми вычислениями — см. главу 3), пока неясно. Но сама идея очень красива, как и все то, что придумала природа безо всякого нашего участия и что достойно искреннего восхищения!

2.7.1 Рекомендуемая литература для дальнейшего изучения

Если вы желаете продолжить изучения биомолекулярных вычислений на более продвинутом уровне и узнать текущее состояние дел в этой области то мы рекомендуем известный обзор современных парадигм ДНК-вычислений [47].

Немало источников можно найти в интернете. Например [52, 53]. Что касается информации на русском языке, то рекомендуем посетить сайт [54], агрегирующий информацию из многих источников и содержащий почти все статьи по данной тематике, опубликованные в России.

3 Квантовые компьютеры

3.1 Введение

Еще в начале 80-х годов XX века Ричард Фейнман (*Richard Feynman*) в работе [5] отметил, что определенные квантовые эффекты не могут быть эффективно смоделированы на классических компьютерах. Это наблюдение привело к рассуждениям, что возможно обычные вычисления можно выполнить более эффективно, если использовать эти квантовые эффекты. Однако построение квантовых компьютеров — вычислительных машин, использующих подобные квантовые эффекты, казалось делом сложным. К тому же никто не был уверен, что использование квантовых эффектов ускорит вычисления, и поэтому область квантовых вычислений прогрессировала весьма слабо. И только в 1994, когда Питер Шор (*Peter Shor*) удивил всех, описав полиномиальный квантовый алгоритм для разложения целых чисел на множители [6, 7], квантовым вычислениям было уделено должное внимание. Это открытие привело к суматохе, как среди экспериментаторов, которые стали пытаться построить квантовый компьютер, так и среди теоретиков, пытающихся разработать

другие квантовые алгоритмы. Дополнительный интерес к этому вопросу был подстегнут изобретением квантовой передачи ключей шифрования и более позже сообщениями об экспериментальных успехах квантовой телепортации и демонстрацией двухбитного квантового компьютера.

Мощь квантового компьютера обусловлена квантовым параллелизмом. В классическом случае, время требуемое на некоторые типы вычислений тоже может быть уменьшено при использовании параллельных процессоров. Однако, чтобы достичь экспоненциального уменьшения времени вычисления требуется экспоненциального увеличения числа процессоров и, следовательно, экспоненциального увеличения физического размера вычислительной системы. А в квантовых системах степень параллелизма увеличивается экспоненциально по отношению к размеру системы. Таким образом, экспоненциальное увеличение квантового параллелизма системы требует всего лишь линейного увеличения размера требуемого физического пространства.

К сожалению, тут имеется определенное неудобство. Хотя квантовые системы могут выполнять огромный объем параллельных вычислений, доступ к результатам вычислений ограничен. Получение результатов представляет собой совершение измерения, которое влияет на квантовое состояние системы. В свете вышеизложенного вырисовывается ситуация, которая представляется даже хуже классической модели вычислений: мы можем получить результат только от одного параллельного процесса, причем из-за вероятностной природы измерения, мы даже не можем выбрать этот процесс из огромного множества аналогичных процессов. К счастью, за последние несколько лет различным ученым удалось найти непростые методы извлечения полезной информации из квантового параллелизма. Например, один из методов заключается в нахождении некоего общего свойства всех выходных значений, такого как симметричность или период функции. Этот метод используется в алгоритме факторизации Шора. Другой метод заключается в преобразовании квантового состояния, увеличивающем вероятность измерения интересующего нас значения. Алгоритм поиска Гровера (Grover) использует именно такую технику усиления.

В следующем параграфе мы напомним основные понятия квантовой механики, которые важны для квантовых вычислений. В разделе 3.3 мы рассмотрим понятие квантового бита (quantum bit) или кубита (qubit). В отличие от обычных классических битов квантовый бит может приведен в состоянии суперпозиции, когда он одновременно хранит 0 и 1. Чтобы это представить, нельзя пользоваться классическими представлениями: квантовый бит, представляющий 0 и 1 одновременно, нельзя рассматривать ни как нечто среднее между 0 и 1, ни как скрытое неизвестное состояние, представляющее 0 и 1 с определенными вероятностями. Даже один кубит имеет интересные приложения. Например, серия одиночных кубитов может использоваться для безопасной передачи секретных ключей шифрования [14].

Однако реальная мощь квантовых вычислений происходит от экспоненциальности размера пространства состояний множества квантовых битов: в то время как один кубит может быть в суперпозиции 0 и 1, то набор из n кубитов может быть в суперпозиции всех 2^n возможных значений. Знаменитый EPR⁵ парадокс (см. раздел

⁵EPR = Эйнштейн (Einstein), Подольский (Podolsky) и Розен (Rosen)

3.3.3) есть проявление *сцепленных* состояний, которые образуют некоторую часть квантового пространства состояний, чему нет аналога в классических системах.

Мы обсудим два типа операций над квантовой системой: измерение и преобразование квантового состояния. Большинство квантовых алгоритмов состоят из последовательности квантовых преобразований, за которыми следует измерение. Напомним, что классические компьютеры могут быть представлены как набор логических вентилях, достаточно универсальных, чтобы любое классическое вычисление могло быть выполнено с помощью последовательности этих вентилях. Аналогично, любые квантовые вычисления представляются с помощью набора примитивных квантовых преобразований, называемых *квантовыми вентилями*. Имея достаточное число кубит можно построить универсальную квантовую машину Тьюринга. Квантовая физика налагает определенные ограничения на допустимые преобразования. В частности, все квантовые преобразования, и следовательно все квантовые вентили, а также все квантовые вычисления должны быть обратимы. Заметим, что все классические алгоритмы могут быть вычислены на квантовом компьютере, т.е. обратимым способом. Некоторые квантовые вентили описаны в разделе 3.4.

Два приложения использующих квантовые вентили и сцепленные состояния описаны в разделе 3.4.2: телепортация и плотное кодирование. Телепортация — это передача квантового состояния по классическим каналам связи. Удивительно, что телепортация возможна, т.к. квантовая механика утверждает, что невозможно не только дублировать квантовые состояния, но даже измерять их без изменения оных. Таким образом, на первый взгляд неясно, какую информацию можно было бы переслать по классическим каналам, чтобы иметь возможность восстановить неизвестное квантовое состояние на другом конце канала связи. Плотное кодирование — приложение двойственное к телепортации, использует один кубит для передачи двух бит классической информации. И телепортация и плотное кодирование основаны на использовании сцепленных состояний, описанных в EPR-эксперименте.

И наконец в разделе 3.4.3 мы увидим, откуда может взятаться экспоненциальное ускорение по сравнению с классическими компьютерами. На вход квантовому компьютеру можно подать суперпозицию состояний, представляющую собой все возможные входные значения. Выполнив вычисления над этим начальным состоянием, на выходе мы получим суперпозицию всех соответствующих выходных значений. Таким образом, за то время, что требуется классическому компьютеру для обработки одного входного значения, квантовый компьютер может рассчитать значения для всех входных состояний. Это и называется квантовым параллелизмом. Однако измерение результирующих состояний будет вероятностным образом выдавать одно из значений этой суперпозиции и в тоже время уничтожать все остальные результаты вычисления. Все это будет детально описано в разделе 3.4.3.

В разделе 3.5 будет детально описан полиномиальный алгоритм факторизации Шора (Shor). Лучший известный классический алгоритм факторизации целых чисел требует экспоненциального времени, и общепринято считать, что не существует классического полиномиального алгоритма. Красивый же алгоритм Шора использует преимущество квантового параллелизма с помощью квантового аналога преобразования Фурье (Fourier).

Лов Гровер (Lov Grover) разработал метод поиска в неупорядоченном списке из n элементов за $O(\sqrt{n})$ шагов на квантовом компьютере. Классические компьютеры

не могут выполнить неструктурированный поиск за время меньшее, чем $O(n/2)$, поэтому, как доказано, квантовые компьютеры более эффективны в таком поиске, чем классические. Хотя ускорение всего лишь полиномиальное, а не экспоненциальное, было показано, что алгоритм Гровера оптимален для квантовых компьютеров. Возможно, что алгоритмы поиска работали бы лучше, если бы они смогли бы использовать преимущества структуры поискового пространства. Тэд Хогг (Tad Hogg), как и другие авторы, использовал эти возможности. Различные методы квантового поиска будут описаны в разделе 3.6.

До сих пор неизвестно, может ли мощь квантовых компьютеров быть использована для целого ряда приложений. Одним из мучительных открытых вопросов остается вопрос о возможности квантовых компьютеров решать NP-полные задачи за полиномиальное время.

Возможно величайшим открытым вопросом остается вопрос о возможности построения квантовых компьютеров, пригодных для использования. Существует множество предложений о том, как построить квантовые компьютеры, большинство из которых основывается либо на ионных ловушках, либо на технологии ядерного магнитного резонанса (ЯМР).

В квантовом компьютере на ионных ловушках [8, 9] линейная цепочка ионов (кубитов) помещена в электрическое поле, которое фиксирует их положение. Лазеры, направленные на отдельные атомы, должны работать как однобитные квантовые вентили. Двухбитные операции реализованы с помощью лазера, направленного на один кубит таким образом, чтобы создать импульс, который распространяется через колебания цепочки ионов до следующего кубита, колебания которого останавливаются с помощью другого лазера. Метод ионных ловушек требует абсолютного вакуума и экстремально низких температур.

Наибольшее преимущество метода ЯМР заключается в том, что он может работать и при комнатной температуре. Суть метода состоит в том, чтобы использовать макроскопические объемы материи и представлять кубит как среднее состояние спинов большого числа атомных ядер. Состояниями спинов можно управлять с помощью магнитных полей, а усредненное состояние спинов можно измерять с помощью ЯМР технологий. Основная проблема этого метода состоит в том, что он плохо масштабируется: с ростом числа кубитов n измеряемый сигнал уменьшается как $1/2^n$. Тем не менее, недавно предложенные улучшения [10] могут быть способны преодолеть эту проблему. Были успешно построены ЯМР компьютеры с двумя кубитами [11, 12]. Далее мы больше не будем обсуждать физические и технические проблемы построения квантовых компьютеров.

Величайшим затруднением для построения квантовых компьютеров является некогерентность, т.е. искажение квантового состояния из-за взаимодействия со средой. Какое-то время опасались, что квантовые компьютеры невозможно будет построить, т.к. невозможно достичь необходимой изоляции их от внешней среды. Прорыв произошел не с физической, а с алгоритмической стороны, когда были изобретены методы квантовой коррекции ошибок. Оказалось возможным разработать коды исправляющие ошибки, которые противостояли определенным типам ошибок и позволяли точно восстановить исходное квантовое состояние. Квантовая коррекция ошибок обсуждается в разделе 3.7.

В приложении представлена необходимая информация по тензорному произведе-

дению и непрерывным дробям.

3.2 Квантовая механика

Нелегко понять квантовые явления, из-за того, что нельзя использовать наш повседневный опыт. Конечно, из этой главы вы не получите глубокое понимание квантовой механики (для этих целей см. [13, 14, 15]). Вместо этого мы постараемся дать некоторые представления о природе квантовой механики и математический формализм, необходимый для понимания квантовых вычислений.

Квантовая механика — это строгая математическая теория, порожденная набором аксиом. Следствия этих аксиом описывают поведения квантовых систем. Из этих аксиом следуют и различные наблюдаемые парадоксы: эффект Комптона (Compton effect), когда кажется, что действие-следствие предшествует породившей ее причине; EPR-эксперимент, когда кажется, что возможно действие на расстоянии, распространяющееся быстрее скорости света. Мы подробно обсудим EPR-эксперимент в разделе 3.3.3. Заметим, что проверка большинства предсказаний происходит косвенным образом и требует тщательной постановки эксперимента и специального оборудования. Мы же начнем с описания эксперимента, который можно провести с помощью легкодоступного оборудования и который иллюстрирует ключевые аспекты квантовой механики.

3.2.1 Поляризация фотонов

Фотоны являются единственными частицами, которых мы можем наблюдать непосредственно. Следующий эксперимент можно провести с минимальным оборудованием: лазерная указка (или другой направленный источник света) и три поляроида (поляризационных фильтра), которых можно купить в любом фотомагазине. Эксперимент с помощью фотонов и их поляризации демонстрирует несколько основных принципов квантовой механики.

Эксперимент Луч света направлен на экран. Фильтры A , B , и C , поляризованные соответственно горизонтально, под 45° и вертикально, расположены на пути светового пучка.

Сначала вставим фильтр A . Предположим, что падающий свет поляризован хаотически, тогда интенсивность света на выходе будет половиной от интенсивности падающего света. На выходе фотоны будут поляризованы горизонтально.

Фильтр A нельзя представить как "сито", которое позволяет пролетать только тем фотонам, которые уже были поляризованы горизонтально. В этом случае только мельчайшая часть хаотически поляризованных фотонов была бы поляризована горизонтально, и следовало бы ожидать гораздо более ощутимого ослабления света, прошедшего через этот фильтр.

Затем, когда мы вставим фильтр C , интенсивность на выходе упадет до нуля. Ни один из горизонтально поляризованных фотонов не сможет пройти через вертикальный фильтр. Модель решета может объяснить такое поведение.

$$|\rightarrow\rangle$$

Рис. 1: Измерение как проекция на базис.

И наконец, после того, как фильтр B будет вставлен между фильтрами A и C , появится небольшое освещение экрана, в точности одна восьмая исходной интенсивности света.

Здесь мы наблюдаем другой непривычный эффект. Обычный опыт предполагает, что добавление любого фильтра может только уменьшить интенсивность проходящего света. Как же он может его увеличить?

Объяснение Состояние поляризации фотона может быть смоделировано единичным вектором, указывающим в соответствующем направлении. Любая произвольная поляризация может быть выражена как линейная комбинация $a|\uparrow\rangle + b|\rightarrow\rangle$ двух базисных векторов⁶ $|\rightarrow\rangle$ (горизонтальная поляризация) и $|\uparrow\rangle$ (вертикальная поляризация). Так как нас интересует только направление поляризации (говорить о "величине" тут бессмысленно), то вектор состояний будет единичным вектором, т.е. $|a|^2 + |b|^2 = 1$. В целом, поляризация фотона может быть представлена как $a|\uparrow\rangle + b|\rightarrow\rangle$, где a и b комплексные числа⁷, такие, что $|a|^2 + |b|^2 = 1$.

Заметим, что выбор ортонормированного базиса совершенно произволен: им могут быть любые два ортогональных единичных вектора (в том числе $\{|\swarrow\rangle, |\searrow\rangle\}$).

Постулат измерения утверждает, что каждое измерение имеет свой собственный ортонормированный базис, на который это измерение проецирует измеряемое квантовое состояние. Например, вероятность того, что $\psi = a|\uparrow\rangle + b|\rightarrow\rangle$ будет измерено как $|\uparrow\rangle$ есть $|a|^2$, а вероятность того, что оно будет измерено как $|\rightarrow\rangle$ есть $|b|^2$ (см. рис.1). Так как измерения всегда проводятся по отношению к ортонормированному базису, то далее мы будем полагать все базисы ортонормированными. Заметим, что различные измерительные устройства имеют различные базисы измерения.

Более того, измерение квантового состояния меняет измеряемое квантовое состояние на измеренное состояние. Это значит, что если измерение $\psi = a|\uparrow\rangle + b|\rightarrow\rangle$ дает $|\uparrow\rangle$, то состояние ψ меняется на $|\uparrow\rangle$, и если это состояние еще раз измерить с помощью того же базиса, то будет получено $|\uparrow\rangle$ с вероятностью 1. Таким образом, если исходное состояние не было одним из базисных векторов измерения, то измерение меняет это состояние и более невозможно узнать, каким было это исходное состояние раньше.

Квантовая механика следующим образом объясняет поляризационный эксперимент. Каждый поляризатор измеряет квантовое состояние фотонов по отношению к базису, состоящему из вектора соответствующего поляризации поляризатора и ортогонального к ней вектора. Фотоны проходят сквозь фильтр только тогда, когда их

⁶Обозначение $|\rightarrow\rangle$ объяснено в разделе 3.2.2.

⁷Коэффициенты мнимой части соответствуют круговой поляризации.

измеренная поляризация совпадает с поляризацией фильтра. Фотоны, которые после измерения имеют поляризацию фильтра летят дальше, а остальные отражаются и получают поляризацию перпендикулярную к поляризации фильтра. Например, фильтр A измеряет поляризацию фотона по отношению к базисному вектору $|\rightarrow\rangle$, соответствующему поляризации фильтра A , и все прошедшие через фильтр фотоны имеют поляризацию $|\rightarrow\rangle$. Все отраженные фотоны имеют поляризацию $|\uparrow\rangle$.

Полагая, что источник света порождает хаотически поляризованные фотоны, получаем, что фильтр A измеряет 50% фотонов, как поляризованных горизонтально. Эти фотоны пропускаются фильтром и их состояние становится $|\rightarrow\rangle$. Фильтр C измеряет эти фотоны по отношению к $|\uparrow\rangle$. Но состояние $|\rightarrow\rangle = 0|\uparrow\rangle + 1|\rightarrow\rangle$ проецируется на $|\uparrow\rangle$ с вероятностью 0 и фотоны не пропускаются фильтром C .

Окончательно, фильтр B измеряет квантовое состояние по отношению к базису

$$\left\{ \frac{1}{\sqrt{2}}(|\uparrow\rangle + |\rightarrow\rangle), \frac{1}{\sqrt{2}}(|\uparrow\rangle - |\rightarrow\rangle) \right\},$$

которое мы будем записывать как $\{|\nearrow\rangle, |\nwarrow\rangle\}$. Те фотоны, которые будут измерены как $|\nearrow\rangle$ пропускаются фильтром. Фотоны прошедшие сквозь A с состоянием $|\rightarrow\rangle$ будут измерены B как $|\nearrow\rangle$ с вероятностью 1/2, и, таким образом, 50% фотонов, прошедших через A , пройдут через B и будут в состоянии $|\nearrow\rangle$. Как и раньше, эти фотоны будут измерены фильтром C как $|\uparrow\rangle$ с вероятностью 1/2. Итак, только одна восьмая часть от исходным фотонов сможет пройти через серию фильтров A , B и C .

3.2.2 Пространства состояний и обозначения векторов состояний и сопряженных векторов (Bra/Ket Notation).

Квантовое состояние системы, включающее в себя положение, моменты, поляризации, спины, и т.п. различных частиц, эволюционирует с течением времени подчиняясь уравнению Шрёдингера (Schrödinger). Пространство состояний квантовой системы моделируется гильбертовым (Hilbert) пространством волновых функций.

Для того, чтобы разобраться в квантовых вычислениях, нам придется иметь дело только с конечными квантовыми системами, и нам достаточно рассмотреть лишь конечномерное комплексное пространство со скалярным произведением, которое натянуто на абстрактные волновые функции, такие как $|\rightarrow\rangle$. В частности, для такого пространства состояний могут быть найдены базисы состоящие из единичных векторов.

Пространства квантовых состояний и преобразования действующие над ними могут быть описаны в терминах векторов и матриц или, что более удобно, в терминах сопряженных векторов (bra/ket notation). Эти обозначения были введены Дираком (Dirac) [16]. Ket-вектор $|x\rangle$ обозначает вектор-колонку и обычно используется для описания квантовых состояний. Соответствующий bra-вектор $\langle x|$, обозначает вектор сопряженный к $|x\rangle$. Например, ортонормированный базис $\{(1, 0)^T, (0, 1)^T\}$ для двухмерного комплексного векторного пространства может быть представлен как $\{|0\rangle, |1\rangle\}$. Любая комплексная линейная комбинация $|0\rangle$ и $|1\rangle$, $a|0\rangle + b|1\rangle$, может быть записана как $(a, b)^T$. Заметим, что выбор порядка базисных векторов произволен. Например, представление $|0\rangle$ как $(0, 1)^T$ и $|1\rangle$ как $(1, 0)^T$ тоже будет вполне приемлемо, если последовательно его придерживаться.

Соединение bra-вектора и ket-вектора $\langle x|y\rangle$ или $\langle x|y\rangle$ обозначает скалярное произведение двух векторов. Например, так как $|0\rangle$ - единичный вектор, то $\langle 0|0\rangle = 1$, и, так как $|0\rangle$ и $|1\rangle$ ортогональны, получается $\langle 0|1\rangle = 0$.

$|x\rangle\langle y|$ обозначает внешнее произведение $|x\rangle$ и $\langle y|$. Например, $|0\rangle\langle 1|$ есть такое преобразование, которое отображает $|1\rangle$ в $|0\rangle$, $|0\rangle$ в $(0, 0)^T$, так как

$$\begin{aligned} |0\rangle\langle 1|1\rangle &= |0\rangle\langle 1|1\rangle = |0\rangle \\ |0\rangle\langle 1|0\rangle &= |0\rangle\langle 1|0\rangle = 0|0\rangle = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{aligned}$$

Произведение $|0\rangle\langle 1|$ может быть также записано в матричной форме, где $|0\rangle = (1, 0)^T$, $\langle 0| = (1, 0)$, $|1\rangle = (0, 1)^T$ и $\langle 1| = (0, 1)$. Тогда

$$|0\rangle\langle 1| = \begin{pmatrix} 1 \\ 0 \end{pmatrix} (0, 1) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

Такие обозначения представляют нам удобную возможность задавать преобразования квантовых состояний в терминах преобразования базисных векторов (см. раздел 3.4). Например, преобразование, которое переставляет местами $|0\rangle$ и $|1\rangle$, выражается матрицей

$$X = |0\rangle\langle 1| + |1\rangle\langle 0|.$$

Мы будем придерживаться более интуитивно понятного обозначения

$$\begin{aligned} X : |0\rangle &\rightarrow |1\rangle \\ |1\rangle &\rightarrow |0\rangle, \end{aligned}$$

где явно определяются преобразования над базисными векторами.

3.3 Квантовые биты

Квантовый бит или кубит — это единичный вектор в двумерном векторном пространстве над полем комплексных чисел, для которого выбран и зафиксирован некий базис, обозначаемый $\{|0\rangle, |1\rangle\}$. Ортонормированный базис $|0\rangle$ и $|1\rangle$ может соответствовать $|\uparrow\rangle$ и $|\rightarrow\rangle$ — вертикальной и горизонтальной поляризациям фотона или "спин-вверх", "спин-вниз" состояниям электрона. В квантовых вычислениях базисные состояния $|0\rangle$ и $|1\rangle$ выбраны для представления классических битовых значений 0 и 1 соответственно.

В отличие от классических битов, кубиты могут находиться в суперпозиции $|0\rangle$ и $|1\rangle$, такой как $a|0\rangle + b|1\rangle$, где a и b есть комплексные числа, для которых $|a|^2 + |b|^2 = 1$. Как и в случае с поляризациями фотона, если такая суперпозиция измеряется по отношению к базису $\{|0\rangle, |1\rangle\}$, то вероятность, что измеренное значение — $|0\rangle$ будет $|a|^2$, а вероятность того, что измеренное значение — $|1\rangle$ будет $|b|^2$.

Для того, чтобы говорить о кубитах и квантовых вычислениях в целом, необходимо заранее выбрать некоторый фиксированный базис, по отношению к которому и будут делаться все утверждения. В частности, если не будет указано обратное, все измерения будут проводиться по отношению к стандартному базису для квантовых вычислений $\{|0\rangle, |1\rangle\}$.

Хотя квантовые биты можно привести в состояние суперпозиции, каждым кубитом можно представить только один классический бит. С точки зрения теории информации, один кубит содержит в точности такое же количество информации, как и классический бит, несмотря на то, что кубит может находиться в бесконечном числе состояний. Дело в том, что количество информации в кубите, которое можно извлечь измерением, такое же как и у обычного бита. Напомним, что когда происходит измерение кубита, то состояние кубита становится одним из базисных состояний измерителя, что мы и наблюдали в эксперименте с поляризациями фотонов. Так как каждое измерение имеет свой собственный базис, и, так как кубит живет в двумерном пространстве, то любое измерение даст в результате один из двух базисных векторов из базиса этого измерения. Получается, что как и в классическом случае, когда мы выясняем состояние определенного бита, так и в квантовом, мы можем получить не больше двух различных результатов. Так как измерение изменяет состояние, то невозможно совершить измерение сначала в одном базисе, затем в другом. Более того, как мы увидим в разделе 3.4.1, квантовые состояния не могут быть клонированы, и тем самым невозможно измерить один кубит двумя различными методами даже косвенным образом, скажем скопировав кубит и измерив его копию.

3.3.1 Наборы кубитов

Как только начинаешь рассматривать системы, состоящие более чем из одного кубита, то сразу же возникает интуитивное понимание, откуда берется мощь квантовых вычислителей. Как мы уже говорили, состояние кубита есть вектор в двумерном комплексном пространстве, натянутом на $|0\rangle$ и $|1\rangle$. В классической механике, возможные состояния системы, состоящей из n частиц, чьи отдельные состояния можно описать вектором в двумерном пространстве, образуют $2n$ -мерное векторное пространство. В квантовой же системе порождаемое пространство состояний намного шире: система из n кубитов имеет 2^n -мерное пространство состояний⁸. Именно это экспоненциальное увеличение размерности пространства состояний относительно числа частиц и предполагает возможное экспоненциальное ускорение классических вычислений на квантовых компьютерах. В классической механике пространства состояний отдельных частиц объединяли с помощью декартового произведения (cartesian product). Для квантовых же состояний объединение происходит с помощью тензорного произведения. Подробное описание свойств тензорного произведения и его способы его выражения через векторы и матрицы дано в приложении А. Давайте же кратко взглянем на важные для понимания квантовых вычислений отличия тензорного произведения от декартового.

Пусть V и W два двумерных комплексных векторных пространства с базисами $\{v_1, v_2\}$ и $\{w_1, w_2\}$ соответственно. Декартово произведение этих двух пространств будет иметь базисом объединение базисов этих пространств: $\{v_1, v_2, w_1, w_2\}$. Заметим, что порядок векторов в этом базисе выбран произвольно. В частности, размерность пространства состояний классической системы из нескольких частиц растет

⁸ Действительно, как мы увидим в дальнейшем, пространство состояний — это множество нормализованных векторов в 2^n -мерном пространстве, когда для состояния кубита $a|0\rangle + b|1\rangle$ выполняется $|a|^2 + |b|^2 = 1$.

линейно с ростом их числа, т.к. $\dim(X \times Y) = \dim(X) + \dim(Y)$. Что касается тензорного произведения V и W , то оно будет иметь базис $\{v_1 \otimes w_1, v_1 \otimes w_2, v_2 \otimes w_1, v_2 \otimes w_2\}$. Заметим, что порядок векторов в базисе также произволен. Таким образом, пространство состояний системы из двух кубитов, где базис каждого $\{|0\rangle, |1\rangle\}$, имеет базис $\{|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle\}$, что можно записать более компактно в форме $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$. Вообще, условимся считать, что $|x\rangle$ будет означать $|b_n b_{n-1} \dots b_0\rangle$, где b_i — биты двоичного разложения числа x .

Базисом для трехкубитной системы будет

$$\{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle\},$$

и в общем случае для n -кубитной системы он будет содержать 2^n базисных векторов. Мы видим экспоненциальное увеличение размерности пространства состояний с ростом числа частиц. Размерность тензорного произведения $X \otimes Y$ будет $\dim(X) \times \dim(Y)$.

Вообразим некий макроскопический физический объект, разделенный на множество отдельных частей, разлетающихся в разных направлениях. Состояние такой системы будет полностью описано, если будет описано состояние каждой из составляющих ее частей.

Удивительным и непривычным свойством пространства состояний системы из n квантовых частиц будет то, что состояние этой системы нельзя всегда описывать в терминах отдельных одночастичных состояний. Например, состояние $|00\rangle + |11\rangle$ не может быть разложено на отдельные состояния для каждого из кубитов. Другими словами, мы не можем найти a_1, a_2, b_1, b_2 , для которых

$$(a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) = |00\rangle + |11\rangle,$$

так как

$$(a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) = a_1 a_2 |00\rangle + a_1 b_2 |01\rangle + b_1 a_2 |10\rangle + b_1 b_2 |11\rangle,$$

и из $a_1 b_2 = 0$ следует, что либо $a_1 a_2 = 0$, либо $b_1 b_2 = 0$. Состояния, которые не могут быть разложены подобным образом, называются *сцепленными* (entangled) состояниями. Такие состояния образуют ситуации, которым нет классических аналогов, и в которых наша интуиция бессильна. Именно эти состояния обеспечивают экспоненциальный рост квантового пространства состояний с ростом числа частиц.

Заметим, что для того, чтобы имитировать даже небольшую квантовую систему на обычном компьютере, потребуются огромные вычислительные ресурсы, и эта имитация должна будет записывать эволюцию экспоненциального числа состояний.

Потенциальная мощь квантовых компьютеров обусловлена возможностью использования эволюции квантового состояния как вычислительного механизма.

3.3.2 Измерение

В результате измерения одной или более частиц в любой квантовой системе мы получаем проекцию состояния этой системы до измерения на некоторое подпространство. Затем амплитуда вектора-проекции масштабируется таким образом, чтобы получился единичный вектор состояния. Вероятность, что в результате измерения

мы получим определенное значение, есть сумма квадратов амплитуд всех векторных компонентов, параллельных этому значению. Рассмотрим пример измерения двухкубитной системы. Далее будем считать, что все измерения отдельных кубитов производятся с базисом $\{|0\rangle, |1\rangle\}$, если не будет указано противное. Любое состояние двухкубитной системы может быть выражено как $a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$, где a , b , c и d суть комплексные числа, для которых выполняется $|a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$. Если мы измеряем первый кубит по отношению к базису $\{|0\rangle, |1\rangle\}$, вероятность, что в результате мы получим $|0\rangle$ будет $|a|^2 + |b|^2$. Кроме того, если в результате измерения мы получили для первого кубита $|0\rangle$, состояние системы будет спроецировано на подпространство, содержащее этот вектор $|0\rangle$ — результат измерения, т.е. на подпространство, натянутое на вектора $|00\rangle$ и $|01\rangle$. В результате этой проекции мы получим $a|00\rangle + b|01\rangle$. Чтобы получить состояние системы после измерения, мы должны произвести коррекцию коэффициентов, чтобы общая вероятность была равна 1:

$$\frac{1}{\sqrt{|a|^2 + |b|^2}}(a|00\rangle + b|01\rangle).$$

Измерение дает нам дополнительный способ определить сцепленные состояния. Частицы не сцеплены, если измерение одной не влияет на другую. Например, состояние $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ является сцепленным. Действительно, вероятность того, что первый кубит будет измерен как $|0\rangle$ есть $1/2$, если второй кубит еще не был измерен, но если второй бит уже был измерен, то вероятность того, что первый бит будет измерен как $|0\rangle$ будет 0 или 1 в зависимости от того, как был измерен второй бит: как 1 или как 0 соответственно. Таким образом, мы показали, что результаты измерения первого бита меняются в зависимости от измерения второго бита. С другой стороны, состояние $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle)$ не является сцепленным, так как $\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) = |0\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, и любые измерения первого бита будут давать $|0\rangle$, вне зависимости от того, как был измерен второй бит. Аналогично, второй бит имеет равные шансы быть измеренным как $|0\rangle$ или $|1\rangle$, вне зависимости от того, был ли измерен первый бит.

Заметим, что определение сцепления в терминах измерения одной частицы, влияющего на измерение другой частицы, эквивалентно предыдущему определению сцепленных состояний, как состояний, которые не могут быть записаны в виде тензорного произведения состояний отдельных частиц.

3.3.3 EPR парадокс

Эйнштейн (Einstein), Подольский (Podolsky) и Розен (Rosen) предложили некий эксперимент, который использовал сцепленные состояния таким образом, что казалось было нарушаются фундаментальные принципы теории относительности. Вообразим некое устройство, которое порождает две максимально сцепленные частицы $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, называемые EPR-парой, и посылает одну из них Алисе, другую Бобу.

Алиса и Боб могут быть сколь угодно далеко друг от друга. Предположим, что Алиса измеряет свою частицу и обнаруживает состояние $|0\rangle$. В этом случае объединенное состояние частиц становится $|00\rangle$, и, когда Боб измеряет свою частицу, он также обнаружит её в состоянии $|0\rangle$. Аналогично, если Алиса при измерении получила $|1\rangle$, то тоже получит и Боб. Заметим, что изменение объединенного квантового состояния происходит мгновенно, несмотря на то, насколько далеко находятся частицы друг от друга. Кажется, что такая ситуация позволит Алисе и Бобу обмениваться информацией быстрее, чем скорость света. Дальнейший анализ, как мы еще увидим, показывает, что несмотря на подобное сопряжение между двумя частицами, невозможно использовать этот механизм для коммуникации.

Существуют два стандартных объяснения, которыми описывают сцепленные состояния и свойства их измерений. Оба имеют свои положительные стороны, но оба они неправильные и могут привести к недоразумениям. Давайте разберемся с ними по очереди. Эйнштейн, Подольский и Розен предложили считать, что каждая частица имеет некоторое внутреннее состояние, которое полностью определяет, каким будет результат любого измерения. Это состояние, по крайней мере на данный момент, от нас скрыто, и все что нам остается, это делать вероятностные предсказания. Подобные утверждения называются *теориями скрытых параметров* (local hidden variable theory). Простейшая теория скрытых параметров для EPR-пары состоит в том, что частицы либо обе находятся в состоянии $|0\rangle$, либо обе находятся в состоянии $|1\rangle$, только мы не знаем, в котором именно. В этом случае не требуется предполагать сообщение между удаленными частицами, чтобы объяснить скоррелированные измерения. Однако, с подобной точки зрения невозможно объяснить результаты измерений по отношению к различным базисам. Более того, Белл (Bell) показал, что из любой теории скрытых параметров следует, что определенные измерения должны удовлетворять некоторому неравенству, известному как неравенство Белла. Но результаты проведенных экспериментов показали нарушение этого неравенства. Таким образом, квантовая механика не может быть объяснена никакой разновидностью теории скрытых параметров. В работе [14] представлен легкочитаемый отчет о теореме Белла и связанных с ней экспериментах.

Второе стандартное объяснение основано на понятиях причины и следствия. Например, ранее мы заметили, что измерение выполненное Алисой, влияет на измерение выполняемое Бобом. Однако такая точка зрения также неверна, и приводит, как заметили Эйнштейн, Подольский и Розен, к глубоким противоречиям с теорией относительности. Возможно такое наблюдение EPR-эксперимента, когда внешний наблюдатель видит сначала измерение совершаемое Алисой, затем измерение совершаемое Бобом, но возможно и такое, когда наоборот, другой наблюдатель сначала видит измерение Боба и лишь затем измерение Алисы. Так как наша причинно-следственная терминология неприменима одновременно к обоим наблюдателям, то экспериментальные значения инвариантны относительно выбора наблюдателя. Из этой "причинно-следственной" симметрии следует, что Алиса и Боб не могут использовать EPR-пары для коммуникации со скоростью большей скорости света, и тем самым, кажущийся EPR-парадокс можно считать разрешенным. Единственная корректная трактовка этого эксперимента — это считать, что Алиса и Боб наблюдают одно и тоже вероятностное поведение измеряемых частиц.

Как мы увидим в разделах посвященных плотному кодированию и телепортации, EPR-пары могут быть использованы для коммуникации, хотя и более медленной, чем скорость света.

3.4 Квантовые вентили (Quantum Gates)

До сих пор мы рассматривали только статические квантовые системы, которые изменялись только при измерении. Эволюция любой квантовой системы, пока она не подвергается измерениям, подчиняется уравнению Шрёдингера (Schrödinger) — и при изменении состояний должна сохраняться ортогональность. Для комплексного векторного пространства сохранение ортогональности обеспечивают только унитарные преобразования, частный случай линейных преобразований. Любое линейное преобразование комплексного векторного пространства может быть представлено матрицей. Пусть M^* обозначает сопряженную транспозицию матрицы M . Матрица M унитарна (описывает унитарное преобразование) если $MM^* = I$. Любое унитарное преобразование квантового пространства состояний есть допустимое квантовое преобразование и наоборот. Заметим, что унитарные преобразования можно рассматривать как вращения в комплексном векторном пространстве.

Одним важным следствием из унитарности квантовых преобразований является их обратимость. Таким образом, квантовые вентили должны быть обратимы. Ранее Беннетт (Bennett), Фридкин (Fredkin) и Тоффоли (Toffoli) уже исследовали обратимые вариации стандартных моделей вычислений. Для понимания основных идей их исследования можно посмотреть Фейнмановские лекции о вычислимости [22].

3.4.1 Элементарные квантовые вентили (Simple Quantum Gates)

Далее мы рассмотрим несколько примеров полезных однокубитных преобразований квантового пространства состояний. Так как преобразования линейны, то они полностью будут определены, если задать их действие на базовые векторы. Мы также будем приводить матрицы этих преобразований.

$$\begin{array}{l}
 I : \begin{array}{l} |0\rangle \rightarrow |0\rangle \\ |1\rangle \rightarrow |1\rangle \end{array} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
 X : \begin{array}{l} |0\rangle \rightarrow |1\rangle \\ |1\rangle \rightarrow |0\rangle \end{array} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
 Y : \begin{array}{l} |0\rangle \rightarrow |1\rangle \\ |1\rangle \rightarrow -|0\rangle \end{array} \quad \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\
 Z : \begin{array}{l} |0\rangle \rightarrow |0\rangle \\ |1\rangle \rightarrow -|1\rangle \end{array} \quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}
 \end{array}$$

Приведем общепринятые наименования этих преобразований. I — тождественное преобразование (identity transformation), X — отрицание (negation), Z — операция сдвига фазы (phase shift operation), $Y = ZX$ — некая комбинация двух последних.

Преобразование X мы уже обсуждали в разделе 3.2.2. Легко проверить унитарность всех этих вентилях. Например:

$$YY^* = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = I.$$

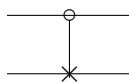
Вентиль управляемое-НЕ(controlled-NOT), C_{not} , действует на два кубита следующим образом: он изменяет состояние второго бита, если первый бит 1, и оставляет второй бит неизменным в противном случае.

$$C_{not} : \begin{array}{l} |00\rangle \rightarrow |00\rangle \\ |01\rangle \rightarrow |01\rangle \\ |10\rangle \rightarrow |11\rangle \\ |11\rangle \rightarrow |10\rangle \end{array} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Преобразование C_{not} унитарно, так как $C_{not}^* = C_{not}$ и $C_{not}C_{not} = I$. Отметим элементарность вентиль C_{not} — он не может быть представлен как тензорное произведение двух однобитных преобразований.

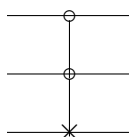
Очень полезно пользоваться графическим представлением квантовых преобразований, особенно если они комбинируются.

Вентиль управляемое-НЕ, C_{not} , обычно представляется следующей графической схемой:

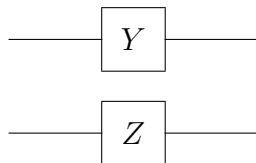


Незакрашенная окружность обозначает управляющий бит, а \times обозначает условное отрицание подчиненного бита. В общем случае, может быть несколько управляющих битов. Некоторые авторы используют закрашенный круг для обозначения отрицания управляющего бита, когда подчиненный бит переключается, если управляющий бит 0.

Аналогично управление-управление-НЕ (controlled-controlled-NOT), который выполняет отрицание последнего бита тогда и только тогда, когда оба первых управляющих бита равны 1, рисуется следующим образом.



Однобитные операции изображаются как соответствующим образом помеченные прямоугольники.



Преобразование Уолша-Адамара (The Walsh-Hadamard Transformation)
 Другим важным однобитным преобразованием является преобразование Адамара (Hadamard Transformation):

$$H : \begin{array}{l} |0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{array}$$

Преобразование H имеет множество полезных приложений. Будучи примененным к $|0\rangle$, H создает суперпозицию состояний $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Будучи примененным к n битам по отдельности, H создает суперпозицию всех 2^n возможных состояний, которую можно рассматривать как двоичное представление чисел от 0 до $2^n - 1$.

$$\begin{aligned} & (H \otimes H \otimes \dots \otimes H)|00\dots 0\rangle \\ &= \frac{1}{\sqrt{2^n}} ((|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes \dots \otimes (|0\rangle + |1\rangle)) \\ &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle. \end{aligned}$$

Преобразование, применяющее H сразу к n битам называется преобразованием Уолша или преобразованием Уолша-Адамара (Walsh-Hadamard). Его можно определить рекурсивно:

$$W_1 = H, W_{n+1} = H \otimes W_n.$$

Невозможность клонирования состояний (No Cloning) Из свойства унитарности следует, что квантовые состояния не могут быть клонированы. Доказательство этого факта приведенное ниже, опубликованное Уотерсом (Wootters) и Зуреком (Zurek) ([23]), суть простейшее следствие линейности унитарных преобразований. Докажем от противного. Допустим U — клонирующее унитарное преобразование, такое, что $U(|a0\rangle) = |aa\rangle$ для всех квантовых состояний $|a\rangle$. Пусть $|a\rangle$ и $|b\rangle$ — два ортогональных квантовых состояния. Тогда $U(|a0\rangle) = |aa\rangle$ и $U(|b0\rangle) = |bb\rangle$. Теперь рассмотрим $|c\rangle = (1/\sqrt{2})(|a\rangle + |b\rangle)$. Из линейности U следует, что

$$\begin{aligned} U(|c0\rangle) &= \frac{1}{\sqrt{2}}(U(|a0\rangle) + U(|b0\rangle)) \\ &= \frac{1}{\sqrt{2}}(|aa\rangle + |bb\rangle). \end{aligned}$$

Но с другой стороны, т.к. U — клонирующее преобразование, мы имеем

$$U(|c0\rangle) = |cc\rangle = 1/2(|aa\rangle + |ab\rangle + |ba\rangle + |bb\rangle),$$

что не равно $(1/\sqrt{2})(|aa\rangle + |bb\rangle)$. Противоречие. Таким образом, не существует унитарных преобразований, клонирующих любое квантовое состояние.

Важно понимать, клонирование какого рода является невозможным. Например, возможно скопировать любое известное квантовое состояние. Принцип невозможности клонирования утверждает, что невозможно клонировать произвольное, неизвестное квантовое состояние. Также возможно получить n частиц в сцепленном состоянии $a|00\dots 0\rangle + b|11\dots 1\rangle$ из неопределенного состояния $a|0\rangle + b|1\rangle$. Каждая из этих частиц будет вести себя абсолютно одинаково, при измерении стандартным базисом квантовых вычислений $\{|0\dots 00\rangle, |0\dots 01\rangle, \dots, |1\dots 11\rangle\}$, но этого не случится, если измерения будут проводится другими базисами. Невозможно создать n частичное состояние $(a|0\rangle + b|1\rangle) \otimes \dots \otimes (a|0\rangle + b|1\rangle)$ из неизвестного состояния $a|0\rangle + b|1\rangle$.

3.4.2 Примеры

Использование элементарных квантовых вентиляей может быть изучено на двух простых примерах: плотном кодировании и телепортации.

В плотном кодировании используется один квантовый бит вместе с EPR-парой для кодирования и передачи двух обычных битов информации. Так как EPR-пары можно распространять заранее, до момента передачи, то необходима только физическая передача всего одного кубита (частицы) для пересылки двух бит информации. Этот результат кажется удивительным, так как в разделе 3.3 утверждалось, что один кубит может содержать не больше одного бита информации.

Телепортация, в отличие от плотного кодирования, использует два классических бита для передачи одного кубита. В свете вышеприведенного принципа невозможности клонирования квантовых состояний возможность телепортации кажется удивительной, так как благодаря ей можно передавать произвольные квантовые состояния.

Основой, как для плотного кодирования, так и для телепортации, является использование сцепленных частиц. Постановка и начало для обоих процессов совпадают. В коммуникации собираются участвовать двое, назовем их Алиса и Боб. Каждый из участников получает по одной из частиц, образующих EPR-пару

$$\psi_0 = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Условимся считать, что Алиса получает первую частицу, а Боб вторую. Затем, до начала передачи, только Алиса может трансформировать свою частицу, и только Боб может трансформировать свою.

Плотное кодирование (Dense Coding)

Действия Алисы: Алиса получает два обычных бита, представляющих целое число от 0 до 3. В зависимости от этого числа, Алиса выполняет одно из преобразований $\{I, X, Y, Z\}$ над своим кубитом из сцепленной пары ψ_0 . Преобразования над одним битом из сцепленной пары подразумевают выполнение идентичных преобразований над другим битом.

Получившиеся состояния приведены ниже.

значение	преобразование	новое состояние
0	$\psi_0 = (I \otimes I)\psi_0$	$\frac{1}{\sqrt{2}}(00\rangle + 11\rangle)$
1	$\psi_1 = (X \otimes I)\psi_0$	$\frac{1}{\sqrt{2}}(10\rangle + 01\rangle)$
2	$\psi_2 = (Y \otimes I)\psi_0$	$\frac{1}{\sqrt{2}}(- 10\rangle + 01\rangle)$
3	$\psi_3 = (Z \otimes I)\psi_0$	$\frac{1}{\sqrt{2}}(00\rangle - 11\rangle)$

Затем Алиса посылает свой кубит Бобу.

Действия Боба: Боб применяет управляемое-НЕ к обоим кубитам сцепленной пары.

начальное состояние	управляемое-НЕ	первый бит	второй бит
$\psi_0 = \frac{1}{\sqrt{2}}(00\rangle + 11\rangle)$	$\frac{1}{\sqrt{2}}(00\rangle + 10\rangle)$	$\frac{1}{\sqrt{2}}(0\rangle + 1\rangle)$	$ 0\rangle$
$\psi_1 = \frac{1}{\sqrt{2}}(10\rangle + 01\rangle)$	$\frac{1}{\sqrt{2}}(11\rangle + 01\rangle)$	$\frac{1}{\sqrt{2}}(1\rangle + 0\rangle)$	$ 1\rangle$
$\psi_2 = \frac{1}{\sqrt{2}}(- 10\rangle + 01\rangle)$	$\frac{1}{\sqrt{2}}(- 11\rangle + 01\rangle)$	$\frac{1}{\sqrt{2}}(- 1\rangle + 0\rangle)$	$ 1\rangle$
$\psi_3 = \frac{1}{\sqrt{2}}(00\rangle - 11\rangle)$	$\frac{1}{\sqrt{2}}(00\rangle - 10\rangle)$	$\frac{1}{\sqrt{2}}(0\rangle - 1\rangle)$	$ 0\rangle$

Заметим, что теперь Боб может измерить второй кубит, не изменяя его квантовое состояние. Если измерение вернуло $|0\rangle$, то значит закодированным значением было либо 0, либо 3, в противном случае — 1 или 2.

Затем Боб применяет H к первому биту:

начальное состояние	первый бит	H (первый бит)
ψ_0	$\frac{1}{\sqrt{2}}(0\rangle + 1\rangle)$	$\frac{1}{\sqrt{2}}(\frac{1}{\sqrt{2}}(0\rangle + 1\rangle) + \frac{1}{\sqrt{2}}(0\rangle - 1\rangle)) = 0\rangle$
ψ_1	$\frac{1}{\sqrt{2}}(1\rangle + 0\rangle)$	$\frac{1}{\sqrt{2}}(\frac{1}{\sqrt{2}}(0\rangle - 1\rangle) + \frac{1}{\sqrt{2}}(0\rangle + 1\rangle)) = 0\rangle$
ψ_2	$\frac{1}{\sqrt{2}}(- 1\rangle + 0\rangle)$	$\frac{1}{\sqrt{2}}(-\frac{1}{\sqrt{2}}(0\rangle - 1\rangle) + \frac{1}{\sqrt{2}}(0\rangle + 1\rangle)) = 1\rangle$
ψ_3	$\frac{1}{\sqrt{2}}(0\rangle - 1\rangle)$	$\frac{1}{\sqrt{2}}(\frac{1}{\sqrt{2}}(0\rangle + 1\rangle) - \frac{1}{\sqrt{2}}(0\rangle - 1\rangle)) = 1\rangle$

Окончательно, Боб измеряет получившийся бит, который позволяет ему сделать выбор между 0 и 3 или между 1 и 2.

Телепортация Целью телепортации является передача квантового состояния частицы с помощью обычных битов и восстановление точного квантового состояния на стороне получателя. Так как квантовые состояния не могут быть клонированы, то при этом необходимо разрушить передаваемое квантовое состояние частицы. Однобитовая телепортация была экспериментально реализована в 1997 году ([24]).

Действия Алисы: Алиса желает передать состояние кубита

$$\phi = a|0\rangle + b|1\rangle$$

Бобу посредством классических каналов связи. Также как и в случае плотного кодирования, каждый из участников имеет по одному кубиту из сцепленной пары

$$\psi_0 = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Алиса применяет декодирующее преобразование из плотного кодирования к кубиту ϕ , которого необходимо передать, и к своей половинке сцепленной пары. Исходным квантовым состоянием будет

$$\begin{aligned} \phi \otimes \psi_0 &= \frac{1}{\sqrt{2}}(a|0\rangle \otimes (|00\rangle + |11\rangle) + b|1\rangle \otimes (|00\rangle + |11\rangle)) \\ &= \frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|100\rangle + b|111\rangle), \end{aligned}$$

в котором Алиса управляет первыми двумя битами, а Боб управляет последним битом. К этому состоянию Алиса применяет преобразования $C_{not} \otimes I$ и $H \otimes I \otimes I$:

$$\begin{aligned}
& (H \otimes I \otimes I)(C_{not} \otimes I)(\phi \otimes \psi_0) \\
&= (H \otimes I \otimes I)(C_{not} \otimes I) \frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|100\rangle + b|111\rangle) \\
&= (H \otimes I \otimes I) \frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|110\rangle + b|101\rangle) \\
&= \frac{1}{2}(a(|000\rangle + |011\rangle + |100\rangle + |111\rangle) + b(|010\rangle + |001\rangle - |110\rangle - |101\rangle)) \\
&= \frac{1}{2}(|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(a|1\rangle + b|0\rangle) + |10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(a|1\rangle - b|0\rangle)).
\end{aligned}$$

Алиса измеряет первых два кубита, чтобы получить один из равновероятных результатов $|00\rangle$, $|01\rangle$, $|10\rangle$, или $|11\rangle$. В зависимости от результатов измерения квантовое состояние кубита Боба будет спроецировано на $a|0\rangle + b|1\rangle$, $a|1\rangle + b|0\rangle$, $a|0\rangle - b|1\rangle$, или $a|1\rangle - b|0\rangle$ соответственно. Алиса посылает результат своего измерения Бобу двумя обычными битами.

Заметим, что когда Алиса совершит измерение, она необратимо изменит состояние исходного кубита ϕ . Поэтому телепортация не противоречит принципу невозможности клонирования состояний.

Действия Боба: Когда Боб получает два классических бита от Алисы, он узнает способ найти соответствие между своей половинкой сцепленной пары и исходным состоянием кубита Алисы.

полученные биты	состояние	декодирование
00	$a 0\rangle + b 1\rangle$	I
01	$a 1\rangle + b 0\rangle$	X
10	$a 0\rangle - b 1\rangle$	Z
11	$a 1\rangle - b 0\rangle$	Y

Боб может восстановить исходное состояние кубита Алисы, ϕ , применив соответствующее декодирующее преобразование к своей части сцепленной пары. Заметим, что это преобразование — кодирующее преобразование для плотного кодирования.

3.4.3 Квантовый компьютер

В этом разделе мы обсудим, каким образом квантовая механика может быть использована для выполнения вычислений и каковы их качественные отличия от вычислений, выполняемых на обычном компьютере. Вспомним, что все квантовые преобразования должны быть обратимы. Что касается классических логических вентилей, то логический вентиль NOT — обратим, а вентили AND, OR и NAND — необратимы. Так что совсем неочевидно, как квантовые преобразования смогут реализовать все обычные вычисления.

В первом подразделе описаны полные наборы обратимых логических вентилей с помощью которых можно выполнить любое классическое вычисление на квантовом компьютере. Затем будут описаны наборы вентилей для выполнения любого квантового вычисления. Во втором подразделе обсуждается квантовый параллелизм.

Блоки квантовых вентилях (Quantum Gate Arrays) Удобно использовать bra/ket-обозначения (обозначения векторов состояний, и сопряженных им с помощью угловых скобок) при определении сложных унитарных операций. Для двух произвольных унитарных преобразований U_1 и U_2 , "условное" преобразование $|0\rangle\langle 0| \otimes U_1 + |1\rangle\langle 1| \otimes U_2$ будет также унитарным. Преобразование управляемое-НЕ может быть определено как

$$C_{not} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X.$$

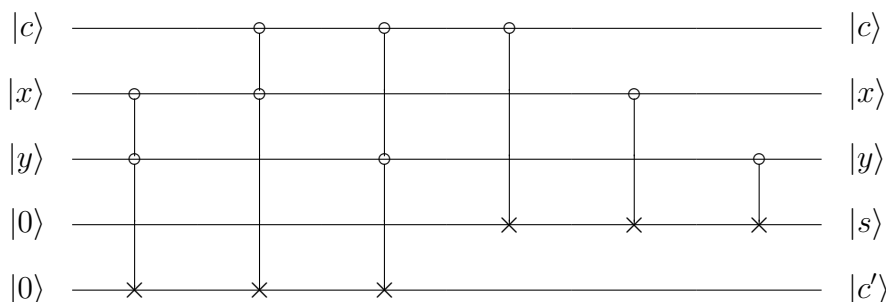
Трехбитное преобразование управление-управление-НЕ, оно же преобразование Тоффоли (Toffoli gate) из раздела 3.4, также можно определить как

$$T = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes C_{not}.$$

С помощью T можно построить полный набор булевых вентилях, так как операторы NOT и AND конструируются следующим образом:

$$\begin{aligned} T|1, 1, x\rangle &= |1, 1, \neg x\rangle \\ T|x, y, 0\rangle &= |x, y, x \wedge y\rangle. \end{aligned}$$

Таким образом, с помощью вентиля T можно построить произвольную комбинаторную схему. Например, следующая схема реализует однобитный сумматор, в котором использованы вентиля Тоффоли и управляемое-НЕ:



Здесь x и y — биты данных, s — их сумма по модулю 2, c — входной бит переноса и c' — выходной бит переноса. Видрел (Vedral), Баренцо (Barenco) и Эккерт (Ekert) ([25]) определили более сложные схемы, включая непосредственное сложение, без разложения на биты.

Вентиль Фридкина (Fredkin gate) представляет собой "управляемую перестановку" и может быть определен, как

$$F = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes S,$$

где S — операция перестановки

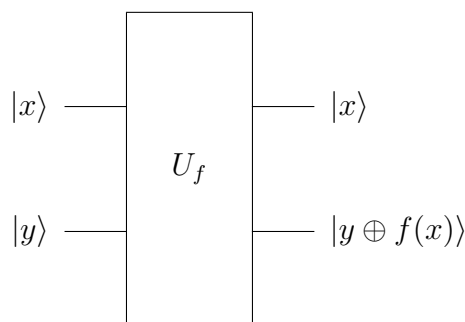
$$S = |00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11|.$$

Из нижеприведенной таблицы видно, что F , как и T , достаточен для представления любой комбинаторной схемы:

$$\begin{aligned} F|x, 0, 1\rangle &= |x, x, \neg x\rangle \\ F|x, y, 1\rangle &= |x, y \vee x, y \vee \neg x\rangle \\ F|x, 0, y\rangle &= |x, y \wedge x, y \wedge \neg x\rangle. \end{aligned}$$

Дойч (Deutsch) в работе [26] показал, что возможно построить обратимые квантовые преобразования для любой классически вычислимой функции. Действительно, можно построить универсальную квантовую машину Тьюринга ([27]). При этом построении нам необходимо достаточное число рабочих кубитов, соответствующих ячейкам на ленте обычной машины Тьюринга.

Из того, что произвольная классическая функция f вычислима на квантовом компьютере, мы полагаем существование *квантового блока вентиляей* (quantum gate-array) U_f , реализующего f . Это преобразование имеет вид $U_f|x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$, где \oplus обозначает не прямую векторную сумму, а поразрядное исключающее-ИЛИ. Определенное таким образом U_f унитарно для любой функции f . Для того, чтобы вычислить $f(x)$ мы применяем $U(f)$ к $|x, 0\rangle$. Так как $f(x) \oplus f(x) = 0$, мы получаем $U_f U_f = I$. Преобразование $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$ можно представить графически:



Хотя преобразования T и F достаточно полны, чтобы реализовать произвольную булеву схему, с их помощью нельзя представить произвольное квантовое преобразование. Для того, чтобы реализовать произвольное квантовое преобразование, необходимо добавить однобитные вращения. Баренцо(Barrenco) и другие в работе [28] показали, что C_{not} вместе со всеми однобитными квантовыми вентилями образует универсальный набор. Достаточно включить в набор следующие однобитные вращения и преобразование сдвига фазы

$$\begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix}, \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix}, \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$$

для всех α , добавить вентиль C_{not} и получится универсальный набор вентиляей.

Как мы увидим, подобные неклассические вращения и сдвиг фазы, являются решающим фактором для использования возможностей квантового компьютера.

Квантовый параллелизм Давайте задумаемся, что произойдет, когда U_f будет применена к входному вектору, находящемуся в состоянии суперпозиции? Ответ прост, но крут: так как U_f — линейное преобразование, то оно будет одновременно применено ко всем базисным векторам суперпозиции, и на выходе будет суперпозиция этих результатов. Таким образом, можно вычислить $f(x)$ для n значений x за одно применение U_f . Этот эффект называется квантовым параллелизмом. Мощь квантовых алгоритмов обусловлена именно этим.

Большинство квантовых алгоритмов начинают с вычисления интересующей функции от суперпозиции всех возможных значений следующим образом. Начинаем с

n -кубитного состояния $|00\dots 0\rangle$. Применяем преобразование Уолша-Адамара W из раздела 3.4.1, чтобы получить суперпозицию

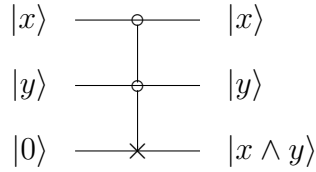
$$\frac{1}{\sqrt{2^n}}(|00\dots 0\rangle + |00\dots 1\rangle + \dots + |11\dots 1\rangle) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle,$$

которую можно рассматривать как суперпозицию всех целых чисел $x : 0 \leq x < 2^n$. Из линейности преобразования следует, что

$$\begin{aligned} U_f\left(\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, 0\rangle\right) &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} U_f(|x, 0\rangle) \\ &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, f(x)\rangle, \end{aligned}$$

где $f(x)$ — интересующая функция. Заметим, что так как n кубитов позволяют работать одновременно с 2^n состояниями, то квантовый параллелизм, за счет экспоненциального увеличения вычислительного пространства при линейном увеличении физического размера системы, обходит компромис времени выполнения и размера системы, свойственный обычному параллелизму.

Рассмотрим простейший пример, состоящий из вентиля Тоффоли, (управление-управление-НЕ) T , который вычисляет логическое умножение двух значений:



Теперь подадим на вход суперпозицию всевозможных однобитных комбинаций x и y (вместе с обязательным 0):

$$\begin{aligned} H|0\rangle \otimes H|0\rangle \otimes |0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\ &= \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle). \end{aligned}$$

Суперпозиция на входе порождает суперпозицию результатов, а именно

$$T(H|0\rangle \otimes H|0\rangle \otimes |0\rangle) = \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |111\rangle)$$

Получившуюся суперпозицию можно рассматривать как таблицу истинности для логического произведения, или, в более общем случае, как функциональный граф. На выходе значения x , y , и $x \wedge y$ сцеплены, то есть измерение результата даст одну строчку в таблице истинности функции, или одну точку в функциональном графе в более общем случае. Заметим, что кубиты можно измерять в любом порядке: измерение результата спроецирует состояние на подпространство всех значений, для

которых f дает измеренный результат, измерение входных значений даст соответствующее значение функции.

Основой любого квантового алгоритма является метод, которым используют квантовый параллелизм, чтобы требуемые результаты были измерены с большой вероятностью. Таким методам нет классических аналогов и поэтому требуются нетрадиционные технологии программирования. Перечислим пару таких технологий, известных на сегодняшний день.

- ”Усилить” интересующие выходные значения. Основная идея состоит в том, чтобы преобразовать состояние таким образом, чтобы интересующее значение имело наибольшую амплитуду, и, следовательно, большую вероятность быть результатом измерения. Примеры такого подхода будут описаны в разделе 3.6.
- Найти общие свойства у всех значений $f(x)$. Эта идея используется в алгоритме Шора (Shor), где используется квантовое преобразование Фурье для получения периода f .

3.5 Алгоритм Шора (Shor)

В 1994 году, побуждаемый результатами работы Даниеля Симона (Daniel Simon) [29], Питер Шор (Peter Shor) нашел вероятностный полиномиальный квантовый алгоритм для разложения n -битных целых чисел на множители. С 1970 года люди ищут эффективный алгоритм для факторизации (разложения на множители) целых чисел. Наиболее эффективный классический алгоритм — это алгоритм Ленстры и Ленстры [30], который экспоненциален по времени относительно размера входа. Входом является список цифр M , размера $n \sim \log M$. Люди настолько уверены, что не существует эффективных алгоритмов факторизации, что надежность криптографических систем, таких как RSA, зависит от сложности этой задачи. Результат же Шора, удивил все общество, породив широкий интерес к квантовым вычислениям. Большинство алгоритмов факторизации, включая алгоритм Шора, используют стандартное сведение задачи факторизации к задаче нахождения периода функции. Шор обычным образом использовал квантовый параллелизм, чтобы получить суперпозицию всех значений функции за один шаг. Затем он вычисляет квантовое преобразование Фурье, которое, как и классическое преобразование Фурье, сопоставляет амплитуде функции частоты ее периодов. Измерение получившегося состояния с большой вероятностью дает период функции, который в свою очередь, используется для разложения целого M . Конечно, вышеприведенное описание в некотором роде чрезмерное упрощение алгоритма. Величайшая сложность состоит в том, что квантовое преобразование Фурье основывается на быстром преобразовании Фурье (fast Fourier transform), и, тем самым, в большинстве случаев дает только приблизительные результаты. При этом извлечение периода становится более трудной задачей, чем описано выше. Также имеется некоторое усложнение из-за масштабирования квантового преобразования Фурье.

Сначала мы опишем квантовое преобразование Фурье, а затем дадим подробное описание алгоритма Шора.

3.5.1 Квантовое преобразование Фурье

Квантовое преобразование Фурье представляет собой вариант дискретного преобразования Фурье (DFT). DFT преобразует дискретную функцию в другую дискретную функцию, определенную на равномерно распределенных точках $k\frac{2\pi}{N}$ интервала $[0, 2\pi)$ для некоторого N . Масштабируя область определения на $\frac{N}{2\pi}$, квантовое преобразование Фурье дает на выходе функцию, определенную на целых числах от 0 до $N - 1$.

Квантовое преобразование Фурье действует на амплитуды квантового состояния, совершая преобразование

$$\sum_x g(x)|x\rangle \rightarrow \sum_c G(c)|c\rangle,$$

где $G(c)$ дискретное преобразование Фурье от $g(x)$, а x и c суть двоичные разложения целых чисел от 0 до $N - 1$. Если измерить состояние после преобразования Фурье, то вероятность того, что результат будет $|c\rangle$ составит $|G(c)|^2$. Заметим, что квантовое преобразование Фурье работает не так, как работает преобразование U_f — никаких выходных результатов в дополнительных регистрах.

Обычно Фурье преобразование отображает временные интервалы на интервалы частот. При этом преобразование Фурье отображает функции периода r в функции, которые имеют ненулевые значения только в точках кратных частоте $\frac{1}{r}$. Таким образом, применяя квантовое преобразование Фурье к периодической функции $g(x)$ с периодом r , мы должны получить $\sum_c G(c)|c\rangle$, где $G(c)$ равно нулю, за исключением значений кратных $\frac{N}{r}$. И когда состояние будет измерено, результатом должно быть число кратное $\frac{N}{r}$, скажем $j\frac{N}{r}$.

Как было выше сказано, квантовое преобразование Фурье работает приближенно. Квантовое преобразование Фурье есть разновидность быстрого преобразования Фурье (FFT), основанного на использовании степени 2, и дающего только приближенные результаты для периодов, не являющихся степенью двойки. Однако, чем большая степень 2 используется в преобразовании, тем лучше достигаемое приближение. Квантовое преобразование Фурье U_{QFT} с основанием 2^m определено как

$$U_{QFT} : |x\rangle \rightarrow \frac{1}{\sqrt{2^m}} \sum_{c=0}^{2^m-1} e^{\frac{2\pi icx}{2^m}} |c\rangle.$$

Для того, чтобы алгоритм Шора был полиномиальным, квантовое преобразование Фурье должно быть эффективно вычислимо. Шор показал, что квантовое преобразование Фурье с основанием 2^m может быть построено с использованием всего лишь $\frac{m(m+1)}{2}$ вентилях. В этом построении используются два типа вентилях. Один вентиль — выполнение уже известного нам преобразования Адамара H . Мы будем использовать обозначение H_j , когда преобразование Адамара применяется к j -му биту. Второй тип вентилях выполняет следующее преобразование:

$$S_{j,k} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta_{k-j}} \end{pmatrix},$$

где $\theta_{k-j} = \pi/2^{k-j}$. Оно действует на k -ый элемент, в зависимости от значения j -го элемента. Квантовое преобразование Фурье задается как

$$H_0 S_{0,1} \dots S_{0,m-1} H_1 \dots H_{m-3} S_{m-3,m-2} S_{m-3,m-1} H_{m-2} S_{m-2,m-1} H_{m-1},$$

с последующей обратной перестановкой битов. Для изучения дальнейших подробностей см. [7].

3.5.2 Более подробный конспект алгоритма Шора

Шаг 1. Квантовый параллелизм Выберем произвольное целое число a . Если a не взаимно простое с M , то мы нашли делитель M . В противном случае продолжаем.

Пусть m таково, что $M^2 \leq 2^m < 2M^2$. Используем квантовый параллелизм, описанный в разделе 3.4.3, для того, чтобы вычислить $f(x) = a^x \bmod M$ для всех целых чисел от 0 до $2^m - 1$. Эта функция будет представлена квантовым состоянием

$$\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x, f(x)\rangle.$$

Шаг 2. Состояние, амплитуда которого имеет период равный периоду f Квантовое преобразование Фурье действует на амплитуду функции, представляющей входное состояние, поэтому, чтобы получить период f , конструируется состояние, амплитудная функция которого имеет такой же период, как f . Чтобы построить такое состояние, измерим кубиты, полученные на первом шаге и представляющие $f(x)$. Будет получено вероятностное значение u , значение которого само по себе нам не интересно. Нам нужно лишь влияние этого измерения на нашу суперпозицию. Это измерение проецирует наше пространство состояний на подпространство измеренного значения, то есть состояние после измерения будет

$$C \sum_x g(x) |x, u\rangle,$$

где множитель C - масштабирующий, а

$$g(x) = \begin{cases} 1 & \text{if } f(x) = u \\ 0 & \text{в противном случае} \end{cases}$$

Заметим, что элементы x , которые действительно появляется в сумме, такие, что $g(x) \neq 0$, отличаются друг от друга на число кратное искомому периоду, и, таким образом, $g(x)$ — искомая функция. Если бы мы смогли измерить два последовательных элемента x в сумме, мы бы получили период. К сожалению, законы квантовой механики разрешают провести только одно такое измерение.

Шаг 3. Применяем квантовое преобразование Фурье Часть нашего состояния $|u\rangle$ нам более не потребуется, поэтому далее мы будем его опускать. Применим квантовое преобразование Фурье к состоянию, полученному на Шаге 2:

$$U_{QFT} : \sum_x g(x) |x\rangle \rightarrow \sum_c G(c) |c\rangle.$$

Из обычного Фурье-анализа следует, что если бы период r функции $g(x)$ был степенью 2-х, то результатом квантового преобразования Фурье было бы

$$C' \sum_j \rho_j |j \frac{2^m}{r}\rangle,$$

где $|\rho_j| = 1$. Если период r не является делителем 2^m , то это преобразование выполняет такую аппроксимацию, что большая часть амплитуды приходится на целые числа, близкие к кратным $\frac{2^m}{r}$.

Шаг 4. Извлечение периода Измерим состояние стандартным базисом для квантовых вычислений, и назовем получившийся результат v . В случае, когда период окажется степенью 2-х, и квантовое преобразование Фурье вернет точные множители от масштабированной частоты, период извлечь легко. В этом случае $v = j \frac{2^m}{r}$ для какого-либо j . Как правило j и r взаимно просты, в этом случае приведение дроби $\frac{v}{2^m}$ даст дробь, чей знаменатель q и будет периодом r . Как уже говорилось, в общем случае квантовое преобразование Фурье дает лишь приблизительные множители масштабированной частоты, что усложняет извлечение периода из результатов измерения. Если период не является степенью 2, хорошее приближение для периода можно получить с помощью расширения непрерывных дробей для $\frac{v}{2^m}$. Эта техника описана в приложении В.

Шаг 5. Нахождение делителя M Если наше приближение для периода четное, то используем алгоритм Евклида для эффективной проверки, что либо $a^{q/2} + 1$, либо $a^{q/2} - 1$ имеют нетривиальный общий множитель с M . Дело в том, что если q на самом деле является периодом $f(x) = a^x \bmod M$, то для всех x $a^q a^x = a^x \bmod M$, то $a^q = 1 \bmod M$.

Шаг 6. При необходимости повторяем алгоритм Следующие причины могут привести к тому, что наш процесс не найдет делителя M :

1. Значение v будет сильно отличаться от ближайшего целого кратного $\frac{2^m}{r}$.
2. Период r и множитель j могут иметь общий множитель, и поэтому знаменатель q , который рассматривается нами как период, в случае взаимной простоты r и j , таковым являться не будет.
3. Шаг 5 вернет M как тривиальный делитель M .
4. Период $f(x) = a^x \bmod M$ будет нечетным.

Однако многократные повторения описанного алгоритма с большой вероятностью дадут нам делитель M .

Комментарий ко второму шагу алгоритма В общем можно полностью пропустить второй шаг. Применим тензорную комбинацию квантового преобразования Фурье и эквивалентного преобразования, $U_{QFT} \otimes I$, к $C \sum_{x=0}^{2^n-1} |x, f(x)\rangle$, чтобы получить

$$C' \sum_{x=0}^{2^n-1} \sum_{c=0}^{2^m-1} e^{\frac{2\pi i x c}{2^m}} |c, f(x)\rangle,$$

что будет равно

$$C' \sum_u \sum_{x|f(x)=u} \sum_c e^{\frac{2\pi i x c}{2^m}} |c, u\rangle,$$

для u , принадлежащих области значений $f(x)$. Это дает суперпозицию результатов, полученных на третьем шаге алгоритма для всевозможных значений u . Квантовое преобразование Фурье применяется к группе функций g_u , проиндексированных u , где

$$g_u = \begin{cases} 1 & \text{если } f(x) = u \\ 0 & \text{в противном случае} \end{cases}$$

Примененное выше преобразование $U_{QFT} \otimes I$ может быть представлено в виде

$$U_{QFT} \otimes I : C \sum_{u \in R} \sum_{x=0}^{2^n-1} g_u(x) |x, f(x)\rangle \rightarrow C' \sum_{u \in R} \sum_{x=0}^{2^n-1} \sum_{c=0}^{2^n-1} G_u(c) |c, u\rangle,$$

где $G_u(c)$ — дискретное преобразование Фурье от $g_u(x)$, а R — область значений $f(x)$. Измеряем c и как раньше выполняем шаги 4 и 5.

3.6 Задачи поиска

Большой класс задач может быть определен, как задачи поиска, имеющих формулировку вида "найти такое x , для которого $P(x)$ —истина", где P — некоторый предикат. У этого класса задач широкий диапазон: от задач сортировки и раскраски графа до поиска в базе данных. Например:

- Дан n -элементный вектор A , нужно найти перестановку π набора $[1..n]$, такую что $\forall 1 \leq i < n : A_{\pi(i)} < A_{\pi(i+1)}$.
- Дан граф (V, E) с n вершинами V и e ребрами $E \subseteq V \times V$ и набор из k цветов C , нужно найти отображение c из V в C (раскраску вершин), такую что $\forall (v_1, v_2) \in E : c(v_1) \neq c(v_2)$.

Для определенных типов задач, там где можно использовать структуру задачи, известны эффективные алгоритмы. Многие задачи поиска, такие как 3-SAT (3-выполнимость), раскраска графа или поиск в отсортированном списке, имеют структурированное поисковое пространство, в котором полные решения могут быть построены из небольших частных решений. Но в общем случае, когда нет структуры поискового пространства, лучшее, что можно предложить в классическом случае — это проверять предикат $P(x_i)$ на каждом вероятностным образом выбранном x_i .

Для поискового пространства размера N , в общем случае неструктурированной поисковой задачи, вычислительная сложность составляет $O(n)$ умноженное на время, требуемое для проверки предиката P на одном значении. А на квантовом компьютере, как показал Гровер (Grover), неструктурированная задача поиска может быть решена с фиксированной вероятностью за время $O(\sqrt{N})$. Таким образом, доказано ([31]), что алгоритм поиска Гровера более эффективен, чем любой алгоритм поиска, выполняемый на классическом компьютере.

Алгоритм поиска Гровера осуществляет поиск на полностью неструктурированном поисковом пространстве. Хотя алгоритм Гровера оптимален [32] [33] [34] для

полностью неструктурированного поиска, многие задачи поиска допускают поиск по структурированному пространству решений. Можно ожидать, что в этом случае структура поиска обусловит наличие более эффективных поисковых стратегий. Например, проблемы выполнения ограничений (constraint satisfaction problems), такие как SAT или раскраска графа, имеют структурированные поисковые пространства, в которых полные решения могут быть построены из небольших частных случаев.

Тэд Хогг (Tad Hogg) разработал квантовые алгоритмы, которые используют структуру задачи таким же образом, как и классические эвристические алгоритмы поиска. Одна из проблем такого подхода состоит в том, что введение структуры задачи в алгоритм делает последний достаточно сложным для анализа вероятности правильного ответа за одну итерацию. Из-за этого до сих пор неизвестно, насколько эффективны алгоритмы Хогга. В классическом случае эффективность эвристических алгоритмов оценивается эмпирически, путем тестирования алгоритма. Но из-за экспоненциального замедления, которое имеет место при имитировании квантовых компьютеров на классических, в данный момент тестирование возможно только для небольших задач. Проведенные эксперименты на небольших задачах показали, что алгоритмы Хогга более эффективны, чем алгоритм Гровера, но похоже, что ускорение всего лишь полиномиально. Так что пока не будут построены достаточно большие квантовые компьютеры, или будут найдены лучшие методы анализа подобных алгоритмов, нельзя будет с уверенностью определить эффективность.

3.6.1 Алгоритм поиска Гровера

Алгоритм поиска Гровера осуществляет поиск в неупорядоченном списке размера N . Пусть n таково, что $2^n \geq N$. Положим, что предикат P от n -битного значения x реализован квантовым вентилем U_P :

$$U_P : |x, 0\rangle \rightarrow |x, P(x)\rangle,$$

где *истина* представлена единицей.

Первый шаг стандартен для квантовых вычислений, и описан в разделе 3.4.3. Вычислим P для всевозможных входных значений x_i , применяя U_P к регистру, содержащему суперпозицию $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$ всех 2^n возможных входных значений x , вместе с регистром, содержащим 0:

$$U_P : \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, 0\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, P(x)\rangle.$$

Самое сложное — это вытащить что-нибудь полезное из этой суперпозиции. Для любого x_0 , для которого $P(x_0)$ истина, $|x_0, 1\rangle$ будет частью суперпозиции $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, P(x)\rangle$, но так как его амплитуда всего лишь $\frac{1}{\sqrt{2^n}}$, то вероятность того, что измерение этой суперпозиции даст x_0 равна всего лишь 2^{-n} . Трюк заключается в том, чтобы изменить квантовое состояние $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, P(x)\rangle$ таким образом, чтобы значительно увеличить амплитуду векторов $|x, 1\rangle$, для которых выполняется заданный предикат, и уменьшить амплитуду остальных векторов $|x, 0\rangle$, для которых предикат не выполняется. Как только мы выполним такое преобразование, нам останется всего лишь измерить последний кубит, представляющий $P(x)$. Из-за произведенного изменения

амплитуды велика вероятность, что результат будет 1. В этом случае, измерение проецирует состояние $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, P(x)\rangle$ на подпространство $\frac{1}{\sqrt{2^k}} \sum_{i=1}^k |x_i, 1\rangle$, где k — количество решений. Дальнейшее измерение оставшихся кубитов даст нам одно из этих решений. Если в результате измерения кубита $P(x)$ мы получим 0, то мы заново запускаем весь алгоритм, и суперпозиция $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x, P(x)\rangle$ должна быть заново получена.

Алгоритм Гровера состоит из следующих шагов:

1. Подготовить регистр, содержащий все возможные значения $x_i \in [0 \dots 2^n - 1]$.
2. Вычислить $P(x_i)$ от данного регистра.
3. Изменить амплитуду a_j на $-a_j$ для тех x_j , для которых $P(x_j) = 1$. Эффективный алгоритм для такого избирательного изменения знаков описан в разделе 3.6.3. Ниже показана диаграмма амплитуд после применения такого преобразования.
4. Применить инверсию относительно среднего значения, чтобы увеличить амплитуду тех x_j , у которых $P(x_j) = 1$. Квантовый алгоритм для эффективного осуществления такой инверсии приведен в разделе 3.6.2. Ниже приведены получившиеся амплитуды. Заметим, что амплитуды всех x_i , у которых $P(x_i) = 0$, уменьшились на незаметную величину.
5. Повторить шаги со второго по четвертый $\frac{\pi}{4}\sqrt{2^n}$ раз.
6. Считать результат.

Бойер(Boyer) и другие ([33]) провел детальный анализ производительности алгоритма Гровера. Было доказано, что алгоритм Гровера оптимален (с точностью до постоянного множителя), то есть никакой другой квантовый алгоритм не сможет быстрее осуществить неструктурированный поиск. Также было показано, что если есть только один x_0 , такой, что $P(x_0)$ истинен, то после $\frac{\pi}{8}\sqrt{2^n}$ повторений шагов со второго по четвертый, вероятность ошибки будет 0.5. После $\frac{\pi}{4}\sqrt{2^n}$ повторений вероятность ошибки упадет до 2^{-n} . Забавно, что последующие итерации увеличивают вероятность ошибки. Например, после $\frac{\pi}{2}\sqrt{2^n}$ итераций вероятность ошибки будет близка к 1. Существует множество классических алгоритмов, в которых повторение определенных действий может приводить только к более лучшим результатам. Повторение же квантовых процедур может некоторое время улучшать результаты, но после достаточного числа повторений результаты опять ухудшатся. Квантовые процедуры суть унитарные преобразования, вращения в комплексном пространстве, и таким образом повторение квантовых преобразований может вращать квантовое состояние, делая его все ближе и ближе к желаемому, но со временем вращение будет удалять вращаемое состояние все дальше и дальше от желаемого. Поэтому, чтобы получить полезные результаты повторяя квантовые преобразования, надо знать когда остановиться.

3.6.2 Инверсия над средним значением

Чтобы найти инверсию над средним значением для квантового компьютера, заметим, что такая инверсия должна быть унитарным преобразованием. Кроме этого, чтобы алгоритм в целом мог решить задачу за $O(\sqrt{N})$, эта инверсия должна выполняться эффективно. Как будет далее показано, эта инверсия может быть реализована с помощью $O(n) = O(\log(N))$ квантовых вентилей. Легко видеть, что преобразование

$$\sum_{i=0}^{N-1} a_i |x_i\rangle \rightarrow \sum_{i=0}^{N-1} (2A - a_i) |x_i\rangle,$$

описывается следующей $N \times N$ матрицей:

$$D = \begin{pmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \cdots & \frac{2}{N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} - 1 \end{pmatrix}.$$

Так как $DD^* = I$, то D унитарно и тем самым является допустимым квантовым преобразованием. Теперь мы вернемся к вопросу эффективности этого преобразования и покажем, что оно может быть реализовано с помощью $O(n) = O(\log(N))$ элементарных квантовых вентилей. Следуя Гроверу, заметим, что D может быть определено как $D = WRW$, где W — преобразование Уолша-Адамара, описанное в разделе 3.4 и

$$R = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots \\ 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & -1 \end{pmatrix}.$$

Чтобы убедиться, что $D = WRW$, рассмотрим $R = R' - I$, где I — эквивалентное преобразование и

$$R' = \begin{pmatrix} 2 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots \\ 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Теперь $WRW = W(R' - I)W = WR'W - I$. Легко проверить, что

$$WR'W = \begin{pmatrix} \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} & \frac{2}{N} & \cdots \\ \frac{2}{N} & \cdots & \cdots & \frac{2}{N} \\ \frac{2}{N} & \cdots & \frac{2}{N} & \frac{2}{N} \end{pmatrix},$$

и, таким образом, $WR'W - I = D$.

3.6.3 Изменение знака

Теперь нам осталось объяснить, как инвертировать амплитуду искомого результата. Мы покажем простой и удивительный способ инвертирования амплитуды в точности у тех состояний, у которых $P(x) = 1$ для произвольного P . Пусть U_P будет

блок квантовых вентилей, вычисляющих $U_P : |x, b\rangle \rightarrow |x, b \oplus P(x)\rangle$. Применим U_P к суперпозиции $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$ и специально подобранному $b = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, чтобы и знак всех $x : P(x) = 1$ изменился, и b остался неизменным.

Чтобы в этом убедиться, давайте положим $X_0 = \{x | P(x) = 0\}$ и $X_1 = \{x | P(x) = 1\}$ и рассмотрим применение U_P .

$$\begin{aligned}
U_P(|\psi, b\rangle) &= \frac{1}{\sqrt{2^{n+1}}} U_P\left(\sum_{x \in X_0} |x, 0\rangle + \sum_{x \in X_1} |x, 0\rangle - \sum_{x \in X_0} |x, 1\rangle - \sum_{x \in X_1} |x, 1\rangle\right) \\
&= \frac{1}{\sqrt{2^{n+1}}} \left(\sum_{x \in X_0} |x, 0 \oplus 0\rangle + \sum_{x \in X_1} |x, 0 \oplus 1\rangle - \sum_{x \in X_0} |x, 1 \oplus 0\rangle - \sum_{x \in X_1} |x, 1 \oplus 1\rangle\right) \\
&= \frac{1}{\sqrt{2^{n+1}}} \left(\sum_{x \in X_0} |x, 0\rangle + \sum_{x \in X_1} |x, 1\rangle - \sum_{x \in X_0} |x, 1\rangle - \sum_{x \in X_1} |x, 0\rangle\right) \\
&= \frac{1}{\sqrt{2^n}} \left(\sum_{x \in X_0} |x\rangle - \sum_{x \in X_1} |x\rangle\right) \otimes b
\end{aligned}$$

Итак, амплитуда состояний из X_1 инвертирована, как и следовало ожидать.

3.6.4 Структурированный поиск

Замечание по преобразованию Уолша-Адамара Помимо представления преобразования Уолша-Адамара, описанного в разделе 3.4.1, существует другое представление, полезное для понимания применения этого преобразования в квантовых алгоритмах. Любое n -битное преобразование Уолша-Адамара представляется $2^n \times 2^n$ матрицей W с элементами W_{rs} , где r и s оба изменяются от 0 до $2^n - 1$. Мы покажем, что

$$W_{rs} = \frac{1}{\sqrt{2^n}} (-1)^{r \cdot s},$$

где $r \cdot s$ есть число единиц, стоящих на одних и тех же позициях, в двоичных разложениях r и s .

Чтобы убедиться в этом равенстве, заметим, что

$$W(|r\rangle) = \sum_s W_{rs} |s\rangle.$$

Пусть $r_{n-1} \dots r_0$ будет бинарным представлением r , а $s_{n-1} \dots s_0$ — бинарным представлением s .

$$\begin{aligned}
W(|r\rangle) &= (H \otimes \dots \otimes H)(|r_n - 1\rangle \otimes \dots \otimes |r_0\rangle) \\
&= \frac{1}{\sqrt{2^n}} (|0\rangle + (-1)^{r_{n-1}} |1\rangle) \otimes \dots \otimes (|0\rangle + (-1)^{r_0} |1\rangle) \\
&= \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{s_{n-1} r_{n-1}} |s_{n-1}\rangle \otimes \dots \otimes (-1)^{s_0 r_0} |s_0\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{s \cdot r} |s\rangle.
\end{aligned}$$

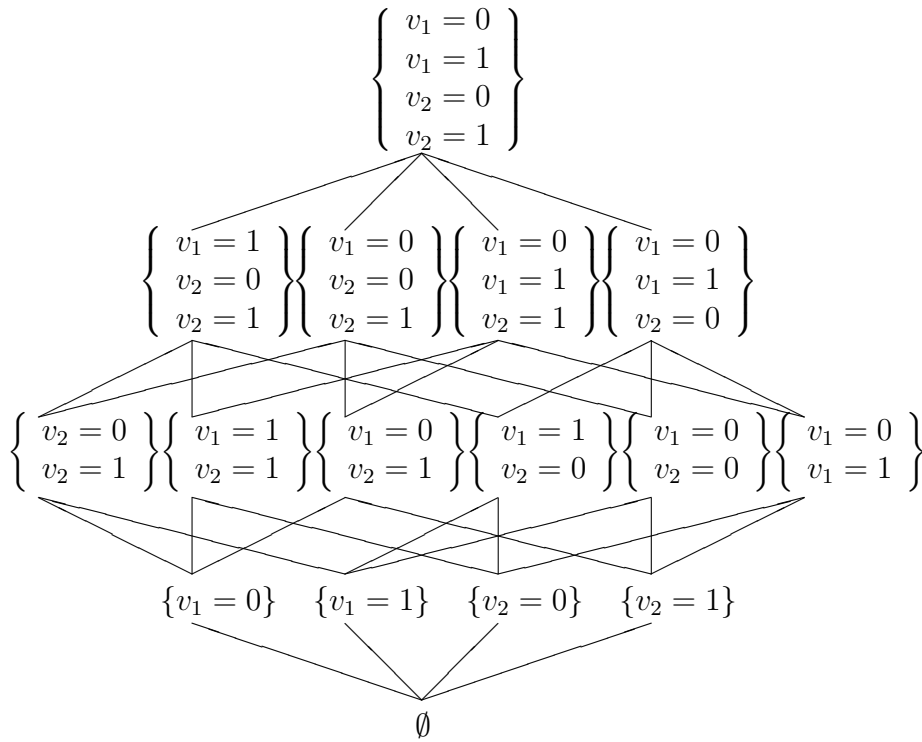


Рис. 2: Решетка присваиваний переменным в CSP

Обзор алгоритмов Хогга В задаче выполнения ограничений (constraint satisfaction problem — CSP) имеются n переменных $V = \{v_1, \dots, v_n\}$, которые могут принимать m различных значений $X = \{x_1, \dots, x_m\}$, подчиняясь некоторым определенным ограничениям C_1, \dots, C_l .

Решениями CSP являются присваивания значений x_i переменным v_j , представляемые отношением (подмножеством) $V \times X$. В задачах такого рода имеется естественная решеточная структура, определяемая наборами присваиваний.

На рисунке 2 показано пространство присваиваний и его решеточная структура для $n = 2$, $m = 2$, $x_1 = 0$, и $x_2 = 1$. Заметим, что эта решетка включает как пустые, так и неоднозначные присваивания.

Используя стандартное соответствие между наборами пронумерованных элементов и двоичными последовательностями, в которых 1 на n -ом месте соответствует включению n -ного элемента, и соответственно 0 — исключению, то мы легко можем поставить этим наборам в соответствие стандартный квантовый базис. Например, на рисунке 3 показана решетка из рисунка 2, представленная с помощью ket-обозначений, где элементы $v_1 = 0$, $v_1 = 1$, $v_2 = 0$ и $v_2 = 1$ пронумерованы в приведенном порядке.

Если некоторое состояние нарушает ограничения, то тоже будут делать все состояния выше него по соответствующей решетке. Метод Хогга по проектированию квантовых алгоритмов для CSP-задач начинает с состояния, в котором вся амплитуда сконцентрирована в состоянии $|0 \dots 0\rangle$ и итеративно "перемещает" амплитуду вверх по решетке, от множеств к надмножествам, избегая наборов, которые нарушают ограничения. Отметим, что этот алгоритм начинает совсем не с того, с чего начинают алгоритмы Шора и Гровера, которые вычисляют функцию от суперпозиции всех входных значений. Хогг предлагает два способа [35, 36] построения унитарной

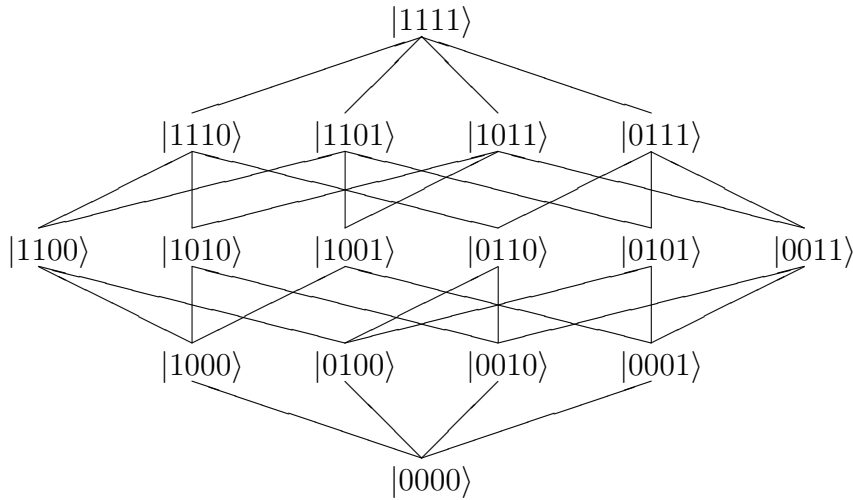
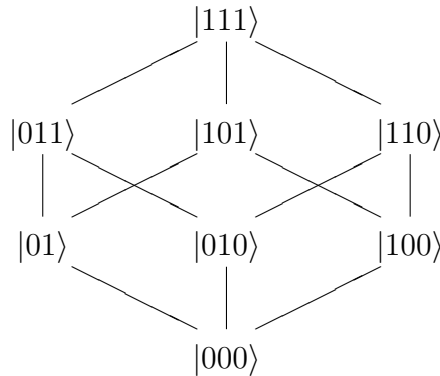


Рис. 3: Решетка присваиваний переменным в ket-обозначениях

матрицы для перемещения амплитуды вверх по решетке. Сначала мы опишем эти два метода, а затем — как избегать плохих наборов.

”Подъем” амплитуды: первый метод. Есть достаточно очевидная матрица, описывающая перемещение амплитуды от множеств к надмножествам. Вся амплитуда состояния соответствующего одноэлементному множеству будет распределена поровну между состояниями соответствующими двухэлементным множествам и так далее. Для трехэлементной решетки



матрица будет выглядеть следующим образом:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

К сожалению, эта матрица не унитарна. Оказалось [35], что ближайшая (в некоторой подходящей метрике) унитарная матрица U_M к произвольной матрице M

может быть найдена с помощью разложения $M = UDV^T$, где D — диагональная матрица, а U и V — унитарные матрицы. Произведение $U_M = UV^T$ дает ближайшую к M унитарную матрицу. Вследствие того, что U_M достаточно близко к M , U_M будет вести себя аналогично M и будет тем самым приемливо перебрасывать амплитуду с множеств на их надмножества.

”Подъем” амплитуды: второй метод. Второй подход [36] использует преобразование Уолша-Адамара. Хогг полагает, что требуемая матрица представляется в виде WDW , где W — преобразование Уолша-Адамара и D — диагональная матрица, элементы которой зависят только от размера наборов. Хогг вычислил элементы D , которые максимизируют перемещение амплитуды к надмножествам. Его расчет основывался на свойстве

$$W_{rs} = \frac{1}{\sqrt{N}}(-1)^{|r \cdot s|} = \frac{1}{\sqrt{N}}(-1)^{|r \cap s|},$$

рассмотренном в разделе 3.6.4.

Перемещение амплитуды от плохих наборов к хорошим. Чтобы это сделать, Хогг ввел регулирование фаз у наборов, в зависимости от степени нарушения ограничений таким образом, что амплитуда, передаваемая множествам от плохих подмножеств, отсекалась, в то время как амплитуда, передаваемая от всех хороших подмножеств, добавлялась. Тут можно применять различные варианты, которые будут работать более или менее эффективно в зависимости от конкретной задачи. В одном случае Хогг предлагает инвертировать фазу у всех плохих наборов, что приведет к некоторому аннулированию передачи амплитуды от плохих наборов к хорошим. Такая инверсия может быть выполнена как в алгоритме Гровера (см. раздел 3.6.3), с использованием предиката P , который проверяет данное состояние на нарушение ограничений.

Другое предложение состоит в том, чтобы присвоить плохим множествам хаотично выбранные фазы, чтобы их вклад в амплитуды надмножеств в среднем компенсировал друг друга и давал 0. Возможны и другие варианты. Из-за того, что методы подобного аннулирования для каждой CSP-задачи свои, то трудно оценить вероятность получения правильного результата. Небольшая серия проведенных экспериментов показала, что временные затраты на поиск растут экспоненциально, хотя значительно менее быстро, чем в неструктурированном случае.

Но пока не будут построены достаточно большие квантовые компьютеры, или не будут найдены лучшие методы анализа подобных алгоритмов, нельзя будет с уверенностью определять их эффективность.

3.7 Квантовая коррекция ошибок (Quantum Error Correction)

Одной из основных проблем построения квантовых компьютеров является необходимость изолирования квантового состояния. Взаимодействие частиц, представляющих кубиты, с внешней средой нарушает квантовое состояние, и приводит к некогерентности или к неунитарным преобразованиям. Стин (Steane) [37] рассчитал, что некогерентность любой системы, которую можно надеяться построить, будет в 10^7 раз больше, чем максимальная некогерентность, позволяющая системе выполнить алгоритм Шора для 130-знакового числа. И все же, добавление алгоритмов

коррекции ошибок к алгоритмам Шора, смягчает эффект некогерентности, позволяя опять надеяться на возможность построения систем, которые смогут выполнить алгоритм Шора для больших чисел.

Поверхностно, квантовая коррекция ошибок аналогична классическим кодам, исправляющим ошибки, в которых используются дополнительные биты для обнаружения и исправления ошибок. Но из-за того, что мы имеем дело с квантовыми состояниями, а не с двоичными данными, сложившаяся ситуация получается несколько более сложной, чем в классическом случае. Квантовая коррекция ошибок должна точно восстановить зашифрованное квантовое состояние. В силу невозможности клонирования или копирования квантового состояния, такое восстановление оказывается более сложным, чем в классическом случае. Тем не менее, оказалось, что классические методы можно модифицировать для работы с квантовыми системами.

3.7.1 Характеризация ошибок

Далее мы подразумеваем, что все ошибки появляются в результате квантового взаимодействия между набором кубитов и внешней средой. Возможные ошибки для каждого отдельного кубита представляются линейной комбинацией из преобразований *нет ошибок* (*no errors*(I), *переворот бита* (*bit flip errors*(X), *ошибка фазы* (*phase errors*(Z) и *ошибка переворота фазы* (*bit flip phase errors*(Y). Таким образом, в общем случае однобитная ошибка представляется преобразованием $e_1I + e_2X + e_3Y + e_4Z$. Взаимодействие с внешней средой преобразует одиночный кубит следующим образом:

$$|\psi\rangle \rightarrow (e_1I + e_2X + e_3Y + e_4Z)|\psi\rangle = \sum_i e_i E_i |\psi\rangle.$$

В общем случае нескольких квантовых регистров, возможные ошибки представляются линейной комбинацией унитарных операторов ошибки E_i . Они могут быть комбинациями однобитных ошибок, вроде тензорных комбинаций от однобитных преобразований-ошибок $\{I, X, Y, Z\}$, или в более общих многобитных преобразованиях. В любом случае, произвольная ошибка может быть записана как $\sum_i e_i E_i$ для некоторого оператора E_i и коэффициентов e_i .

3.7.2 Восстановление квантового состояния

Код, исправляющий ошибки для набора ошибок E_i , состоит из отображения C , которое встраивает n бит данных в $n + k$ битный код, плюс оператор диагностики (*syndrome extraction operators*), который отображает $n + k$ битный код в набор индексов исправляемых ошибок E_i : $i = S_C(E_i(C(x)))$. Если $y = E_j(C(x))$ для некоторой неизвестной, но корректируемой ошибки, то ошибка с номером $S_C(y)$ может быть использована для восстановления правильного закодированного значения $C(x)$, так как $E_{S_C(y)}^{-1}(y) = C(x)$.

Теперь рассмотрим ситуацию с квантовым регистром. Во-первых, состояние регистра может быть суперпозицией базисных векторов. Далее, ошибка может быть комбинацией корректируемых ошибок E_i . Оказывается, что даже в этом случае возможно восстановить закодированное квантовое состояние.

Для данного кода исправляющего ошибки C и оператора диагностики S_C , n -битное квантовое состояние $|\psi\rangle$ будет закодировано в $n+k$ битное квантовое состояние $|\phi\rangle = C|\psi\rangle$.

Предположим, что некогерентность привела к ошибочному состоянию $\sum_i e_i E_i |\phi\rangle$ для некоторой комбинации корректируемых ошибок E_i . Исходное состояние $|\phi\rangle$ может быть восстановлено следующим образом:

1. Применяем оператор диагностики S_C к квантовому состоянию, дополненному достаточным числом битов $|0\rangle$:

$$S_C(\sum_i e_i E_i |\phi\rangle) \otimes |0\rangle = \sum_i e_i (E_i |\phi\rangle \otimes |i\rangle).$$

Квантовый параллелизм даст нам суперпозицию различных ошибок, каждая из которых связана с соответствующим индексом ошибки i .

2. Измеряем компоненту $|i\rangle$ от полученного результата. Мы получаем некоторое вероятностное значение i_0 и проецируем состояние на

$$E_{i_0} |\phi, i_0\rangle$$

3. Применяем преобразование обратное к ошибке $E_{i_0}^{-1}$ к первым $n+k$ кубитам $E_{i_0} |\phi, i_0\rangle$, чтобы получить восстановленное состояние $|\phi\rangle$.

Заметим, что второй шаг проецирует суперпозицию преобразований от нескольких ошибок на одноошибочное преобразование. Следовательно, на третьем шаге нужно всего лишь одно "антиошибочное" преобразование.

3.7.3 Пример коррекции ошибок

Рассмотрим тривиальный код исправляющий ошибки C , который отображает $|0\rangle \rightarrow |000\rangle$ и $|1\rangle \rightarrow |111\rangle$. C может исправить однобитную ошибку переворота

$$E = \{I \otimes I \otimes I, X \otimes I \otimes I, I \otimes X \otimes I, I \otimes I \otimes X\}.$$

Оператором диагностики будет

$$S : |x_0, x_1, x_2, 0, 0, 0\rangle \rightarrow |x_0, x_1, x_2, x_0 \text{ хог } x_1, x_0 \text{ хог } x_2, x_1 \text{ хог } x_2\rangle,$$

соответствующие корректирующие операторы приведены в таблице. Заметим, что в данном случае $E_i = E_i^{-1}$.

№ поврежденного бита	Диагноз	Коррекция ошибки
нет таких	$ 000\rangle$	отдыхать
0	$ 110\rangle$	$X \otimes I \otimes I$
1	$ 101\rangle$	$I \otimes X \otimes I$
2	$ 011\rangle$	$I \otimes I \otimes X$

Рассмотрим кубит $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, который будет закодирован как

$$C|\psi\rangle = |\phi\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

и ошибку

$$E = \frac{4}{5}X \otimes I \otimes I + \frac{3}{5}I \otimes X \otimes I.$$

Получившееся ошибочное состояние будет

$$\begin{aligned} E|\phi\rangle &= \left(\frac{4}{5}X \otimes I \otimes I + \frac{3}{5}I \otimes X \otimes I\right)\left(\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)\right) \\ &= \frac{4}{5}X \otimes I \otimes I\left(\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)\right) + \frac{3}{5}I \otimes X \otimes I\left(\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)\right) \\ &= \frac{4}{5\sqrt{2}}X \otimes I \otimes I(|000\rangle - |111\rangle) + \frac{3}{5\sqrt{2}}I \otimes X \otimes I(|000\rangle - |111\rangle) \\ &= \frac{4}{5\sqrt{2}}(|100\rangle - |011\rangle) + \frac{3}{5\sqrt{2}}(|010\rangle - |101\rangle) \end{aligned}$$

Теперь применим оператор диагностики к $(E|\phi\rangle) \otimes |000\rangle$ следующим образом:

$$\begin{aligned} S_C((E|\phi\rangle) \otimes |000\rangle) &= S_C\left(\frac{4}{5\sqrt{2}}(|100000\rangle - |011000\rangle) + \frac{3}{5\sqrt{2}}(|010000\rangle - |101000\rangle)\right) \\ &= \frac{4}{5\sqrt{2}}(|100110\rangle - |011110\rangle) + \frac{3}{5\sqrt{2}}(|010101\rangle - |101101\rangle) \\ &= \frac{4}{5\sqrt{2}}(|100\rangle - |011\rangle) \otimes |110\rangle + \frac{3}{5\sqrt{2}}(|010\rangle - |101\rangle) \otimes |101\rangle \end{aligned}$$

Измерение последних трех битов этого состояние даст либо $|110\rangle$, либо $|101\rangle$. Предположим измерение даст первый результат, тогда состояние станет

$$\frac{1}{\sqrt{2}}(|100\rangle - |011\rangle) \otimes |110\rangle.$$

Это измерение производит едва ли не магическое действие — исчезают все слагаемые ошибки, кроме одного. Оставшийся от ошибки нанесенный вклад может быть удален путем применения антиошибочного преобразования $X \otimes I \otimes I$ к первым трем битам:

$$\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) = C|\psi\rangle = |\phi\rangle.$$

3.8 Заключение

Квантовые вычисления — это недавно возникшая область, которая обладает достаточным потенциалом, чтобы изменить наши представления о вычислениях, программировании и сложности. Возникло состязание между специалистами в Computer

Science и не только, в разработке новых методов программирования для квантовых компьютеров. Квантовое сцепление и аннулирование фазы представляют новое измерение в методах вычислений. Программирование более не сводится к простому пошаговому формулированию алгоритмов, а требует новых методов регулирования фаз, смешивания и разделения амплитуд для извлечения нужных результатов.

Здесь мы пытались привести аккуратный отчет о состоянии дел в квантовых вычислениях для специалистов по Computer Science и других не-физиков. Были описаны некоторые квантомеханические эффекты, такие как экспоненциальность пространства состояний, сцепленные состояния, линейность преобразований квантовых состояний, именно те эффекты, которые делают возможным квантовый параллелизм. Несмотря на то, что квантовые вычисления должны быть линейны и обратимы, любой классический алгоритм может быть реализован на квантовом компьютере. Однако настоящая мощь этих новых вычислителей, экспоненциальный параллелизм, может быть использован только с помощью новых, передовых технологий программирования. Только недавно такие технологии начали разрабатываться и изучаться.

Был описан полиномиальный по времени алгоритм факторизации Шора, который подстегнул развитие квантовых вычислений. Если был бы построен реальный квантовый компьютер, то алгоритм Шора вывел бы из игры большинство существующих методов шифрования. Алгоритм поиска Гровера, хотя и обеспечивший всего лишь полиномиальное ускорение, показал, что квантовые компьютеры существенно более мощны, чем их классические собратья. Несмотря на доказанную неулучшаемость алгоритма Гровера, есть надежда, что будут найдены более быстрые алгоритмы, использующие структуру и свойства конкретных задач. Был описан один из таких подходов, реализованный Хоггом.

Несколько известных квантовых алгоритмов, не были приведены в данном обзоре. Джонс (Jones) и Моска (Mosca) ([38]) описали реализацию своего алгоритма на двухбитном квантовом компьютере. Этот алгоритм [39], в постоянной временной сложности, может распознать сбалансированную функцию или константу. Гровер ([40]) описал эффективный алгоритм для оценки медианы от набора значений, а Терхал (Terhal) и Смолин (Smolin) ([41]) смогли решить задачу о взвешивании монет (the coin weighing problem) за один шаг.

Кроме этих алгоритмов мало что известно о том, что можно реализовать на реальном квантовом компьютере. Открытым остается вопрос о совпадении или несовпадении классов P и NP на квантовом компьютере.

В тоже время между физиками бродят разные спекуляции о том, что квантовые преобразования может быть "слегка" нелинейны. До сих пор все проведенные эксперименты демонстрировали согласие со стандартной линейной моделью квантовой механики, но слабая нелинейность все же возможна.

Абрамс (Abrams) и Ллойд (Lloyd) ([42]) показали, что даже слабая нелинейность может быть использована для решения всех NP -трудных задач на квантовом компьютере за полиномиальное время. Этот результат еще раз подчеркивает тот факт, что вычисление является фундаментальным физическим процессом, и то, как оно осуществляется, зависит от множества нетривиальных физических следствий.

И конечно, самая страшная физическая проблема квантовых вычислений будет решена, если кому-нибудь удастся построить пригодный для серьезных вычис-

лений квантовый компьютер. Некогерентность — искажение квантового состояния из-за взаимодействия со внешней средой является ключевой проблемой. Большой прорыв в деле борьбы с некогерентностью произошел на алгоритмическом фронте, когда были разработаны методы квантовой коррекции ошибок, а отнюдь не на физическо-технологическом фронте. Здесь уже были вкратце описаны некоторые из изобретенных методов.

Дальнейшие успехи в квантовой коррекции ошибок и в разработке мощных алгоритмов также важны, как и техническая сторона построения квантовых компьютеров, которых уже можно будет поставить на службу народному хозяйству.

3.8.1 Рекомендуемая литература для дальнейшего изучения

Физики любят обзор Эндрю Стина (Andrew Steane) “Quantum computing” [37]. Многие считают, что он слишком растянут в части описания классической теории вычислений и слишком сжат в части квантовой механики. Возможно, после прочтения данной работы читателям будет легче читать статью Стина. Рекомендуем чтение его статьи, так как в ней изложена его собственная точка зрения на данный предмет, особенно в части описания взаимосвязей между информационной теорией и квантовыми вычислениями и в области обсуждения коррекции ошибок, в которой он был одним из главных исследователей и разработчиков. Также там приведен обзор физики непосредственно связанной с построением квантовых компьютеров и отчет об этом, заверченный к июлю 1997. В его статье содержится более подробная история идей, связанных с квантовыми вычислениями, чем в данной работе и, соответственно, имеется большее число ссылок.

Фейнмановские лекции о вычислимости (Richard Feynman’s *Lectures on Computation*) [22] содержат репринт лекции “Квантомеханические компьютеры” (“Quantum Mechanical Computers”) [43], с которой собственно все и началось. Там также обсуждается термодинамика вычислений, которая сильно связана с обратимостью вычислений и теорией информации.

Книга Колина Уильямса (Colin Williams) и Скотта Кливотера (Scott Clearwater) *Explorations in Quantum Computing* [44] поставляется с программами в формате файлов алгебраического пакета символьных вычислений Mathematica (Mathematica notebooks), которые имитируют некоторые квантовые алгоритмы, такие как алгоритм Шора.

Вторая половина октябрьского выпуска the SIAM Journal of Computing содержала шесть конструктивных статей по квантовым вычислениям [32] [27] [7] [29].

В данной работе мы ссылаемся на большинство из этих статей, и более того, их можно выкачать с сервера Лос-Аламосской лаборатории <http://xxx.lanl.gov.archive.quant-ph> или с его российского зеркала <http://ru-arxiv-org.archive.quant-ph>.

Множество другой интересной информации по квантовым вычислениям может быть найдено в интернете. Одним из неплохих мест для начала могут быть странички проекта Stanford-Berkeley-MIT-IBM Quantum Computation Research Project по адресу <http://feynman-stanford-edu.qcomp>, на которых помимо значительного количества информации по квантовым вычислениям имеется множество ссылок на другие интересные сайты.

4 Благодарности

Хотелось бы выразить благодарность Владу Борисову⁰⁰, написавшему отличную обзорную статью для `pcweek-ru`. К сожалению, не представляется возможность сослаться на оригинальную версию статьи, но несколько усеченный вариант можно найти на [54]. Содержание данной статьи легло в основу главы 2.

Также я благодарен Элеонор Риффель (Eleanor Rieffel) и Вольфгангу Полаку (Wolfgang Polak) за работу [4], перевод которой составляет содержание главы 3.

Список литературы

- [1] Компьютерра 47(224) 1997. Квантовые компьютеры и квантовые вычисления. *Беседа с кандидатом физико-математических наук, специалистом по теории алгоритмов Михаилом Вялым (Вычислительный центр РАН)* (<http://www-computerra-ru.1997.47.3-html?inside>)
- [2] Ю. И. Манин, "Вычислимое и невычислимое", М.: Советское радио (1980).
- [3] А. Ю. Китаев, "Квантовые вычисления: алгоритмы и исправление ошибок", Успехи математических наук, (в печати).
- [4] Eleanor Rieffel, Wolfgang Polak An Introduction to Quantum Computing for Non-Physicists <http://xxx.lanl.gov/archive/quant-ph/9809016>. <http://ru.arxiv.org/ps/quant-ph/9809016>.
- [5] R. Feynman, International Journal of Theoretical Physics 21 (1982) 467.
- [6] P.W. Shor, Proceedings of the 35th Annual Symposium on Foundations of Computer Science, pp. 124–134, Institute of Electrical and Electronic Engineers Computer Society Press, 1994, <ftp://netlib.att.com/netlib/att/math/shor/quantum.algorithms.ps.Z>.
- [7] P.W. Shor, Society for Industrial and Applied Mathematics Journal on Computing 26 (1997) 1484, Expanded version of [6].
- [8] J.I. Cirac and P. Zoller, Physical Review Letters 74 (1995) 4091.
- [9] A. Steane, The ion trap quantum information processor, 1996, [quant-ph/9608011](http://arxiv.org/abs/quant-ph/9608011).
- [10] L.J. Schulman and U. Vazirani, Scalable NMR quantum computation, 1998, [quant-ph/9804060](http://arxiv.org/abs/quant-ph/9804060).
- [11] N.A. Gershenfeld and I.L. Chuang, Science 275 (1997) 350.
- [12] R. Laflamme et al., NMR GHZ, 1997, [quant-ph/9709025](http://arxiv.org/abs/quant-ph/9709025).
- [13] R. Feynman, The Feynman Lectures on Physics, Vol. III (Addison-Wesley, Reading, Mass, 1965).

⁰⁰e-mail: vladik@pcweek.ru

- [14] G. Greenstein and A.G. Zajonc, *The Quantum Challenge* (Jones and Bartlett Publishers, Sudbury, Mass, 1997).
- [15] R.L. Liboff, *Introductory Quantum Mechanics* (3rd edition) (Addison-Wesley, Reading, Mass, 1997).
- [16] P. Dirac, *The Principles of Quantum Mechanics*, 4th ed. (Oxford University Press, 1958).
- [17] C.H. Bennett and G. Brassard, *SIGACTN: SIGACT News* (ACM Special Interest Group on Automata and Computability Theory) 18 (1987).
- [18] C.H. Bennett, G. Brassard and A.K. Ekert, *Scientific American* 267 (1992) 50.
- [19] A.K. Ekert et al., *Physical Review Letters* 69 (1992).
- [20] C.H. Bennett, *Physical Review Letters* 68 (1992).
- [21] R.J. Hughes et al., *Photonic Quantum Computing*, edited by S.P. Hotaling and A.R. Pirich Vol. 3076, pp. 2–11, 1997.
- [22] R. Feynman, *Feynman lectures on computation*, 1996.
- [23] W.K. Wootters and W.H. Zurek, *Nature* 299 (1982) 802.
- [24] D. Bouwmeester et al., *Nature* 390 (1997) 575.
- [25] V. Vedral, A. Barenco and A.K. Ekert, *Quantum networks for elementary arithmetic operations*, *Physical Review A*, 1996, quant-ph/9511018.
- [26] D. Deutsch, *Proceedings of the Royal Society of London Ser. A* A400 (1985) 97.
- [27] E. Bernstein and U.V. Vazirani, *Society for Industrial and Applied Mathematics Journal on Computing* 26 (1997) 1411, A preliminary version of this paper appeared in the *Proceedings of the 25th Association for Computing Machinery Symposium on the Theory of Computing*.
- [28] A. Barenco et al., *Physical Review A* 52 (1995) 3457, quant-ph/9503016,
- [29] D.R. Simon, *Society for Industrial and Applied Mathematics Journal on Computing* 26 (1997) 1474, A preliminary version of this paper appeared in the *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*.
- [30] A. Lenstra and H. Lenstra, editors, *The Development of the Number Field Sieve*, *Lecture Notes in Mathematics* Vol. 1554 (Springer Verlag, 1993).
- [31] L.K. Grover, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pp. 212–219, Philadelphia, Pennsylvania, 1996.
- [32] C.H. Bennett et al., *Society for Industrial and Applied Mathematics Journal on Computing* 26 (1997) 1510, quant-ph/9701001.

- [33] M. Boyer et al., Proceedings of the Workshop on Physics of Computation: PhysComp '96, Los Alamitos, CA, 1996, Institute of Electrical and Electronic Engineers Computer Society Press, quant-ph/9605034.
- [34] C. Zalka, Grover's quantum searching algorithm is optimal, 1997, quant-ph/9711070.
- [35] T. Hogg, Journal of Artificial Intelligence Research 4 (1996) 91, quant-ph/9508012.
- [36] T. Hogg, Physical Review Letters 80 (1998) 2473, quant-ph/9508012.
- [37] A. Steane, Reports on Progress in Physics 61 (1998) 117, quant-ph/9708022.
- [38] J.A. Jones and M. Mosca, Journal of Chemical Physics 109 (1998) 1648, quant-ph/9801027.
- [39] D. Deutsch and R. Jozsa, Proceedings of the Royal Society of London Ser. A A439 (1992) 553.
- [40] L.K. Grover, Proceedings of the 30th annual ACM symposium on the theory of computing (1998) 53, quant-ph/9711043.
- [41] B.M. Terhal and J.A. Smolin, Single quantum querying of a database, 1997, quant-ph/9705041.
- [42] D.S. Abrams and S. Lloyd, Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and $\#P$ problems, 1998, quant-ph/9801041.
- [43] R. Feynman, Optics News 11 (1985), Also in *Foundations of Physics*, 16(6):507–531, 1986.
- [44] C.P. Williams and S.H. Clearwater, Explorations in Quantum Computing (Telos, Springer-Verlag, 1998).
- [45] T.A. Hungerford, Algebra (Springer Verlag, New York, Heidelberg, Berlin, 1974).
- [46] G.H. Hardy and E.M. Wright, An Introduction to the Theory of Numbers (Oxford University Press, 1979).
- [47] John H. Reif, Paradigms for Biomolecular Computation, *Unconventional Models of Computation*, 1998.
- [48] Leonard M. Adleman, Molecular Computation of Solutions to Combinatorial Problems, *Science*, 266/94, p. 1021.
- [49] Richard J. Lipton, DNA solution of hard computational problems, *Science*, Number 268/95, p. 542.
- [50] Dan Boneh, Christopher Dunworth, Richard J. Lipton, Breaking DES using a molecular computer. *Technical Report CS-TR-489-95*, Princeton University, May 1995.

- [51] Sam Roweis et al., A sticker based architecture for DNA computation. <ftp://hope.caltech.edu/pub/roweis/DIMACS/stickers.ps>
- [52] <http://www.geocities.com/ResearchTriangle/Lab/5831/dnaftp.html>
- [53] <http://design.alfred.edu/DNAcomputing>
- [54] <http://brd.dorms.spbu.ru>

А Тензорное произведение

Тензорное произведение (\otimes) n -мерного и k -мерного векторов есть nk -мерный вектор. Аналогично, если A и B являются преобразованиями в n и k -мерных пространствах, то $A \otimes B^{00}$ будет преобразованием в nk -мерном пространстве.

Оставим за рамками данной работы точные математические подробности тензорного произведения (для углубленного изучения см. например [45]). Для наших целей достаточно узнать следующие алгебраические правила вычисления тензорных произведений. Для матриц A, B, C, D, U , векторов u, x, y , и скаляров a, b имеют место быть следующие равенства:

$$\begin{aligned} (A \otimes B)(C \otimes D) &= AC \otimes BD \\ (A \otimes B)(x \otimes y) &= Ax \otimes By \\ (x + y) \otimes u &= x \otimes u + y \otimes u \\ u \otimes (x + y) &= u \otimes x + u \otimes y \\ ax \otimes by &= ab(x \otimes y) \end{aligned}$$

$$\begin{aligned} \begin{pmatrix} A & B \\ C & D \end{pmatrix} \otimes U &= \begin{pmatrix} A \otimes U & B \otimes U \\ C \otimes U & D \otimes U \end{pmatrix}, \\ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \otimes U &= \begin{pmatrix} aU & bU \\ cU & dU \end{pmatrix}. \end{aligned}$$

Сопряженная транспозиция дистрибутивна относительно тензорного произведения:

$$(A \otimes B)^* = A^* \otimes B^*.$$

Матрица U унитарна, если ее сопряженная транспозиция является также ее обратной: $U^*U = I$.

Тензорное произведение нескольких матриц унитарно, если каждая из входящих в произведение матриц унитарна с точностью до константы. Пусть $U = A_1 \otimes A_2 \otimes \dots \otimes A_n$. Тогда U унитарно, тогда и только тогда, когда $A_i^*A_i = k_iI$ и $\prod_i k_i = 1$.

$$\begin{aligned} U^*U &= (A_1^* \otimes A_2^* \otimes \dots \otimes A_n^*)(A_1 \otimes A_2 \otimes \dots \otimes A_n) \\ &= A_1^*A_1 \otimes A_2^*A_2 \otimes \dots \otimes A_n^*A_n \\ &= k_1I \otimes \dots \otimes k_nI \\ &= I \end{aligned}$$

⁰⁰С технической точки зрения это правое произведение Кронекера (right Kronecker product).

Например, правило дистрибутивности позволяет провести следующие преобразования:

$$\begin{aligned}
& (a_0|0\rangle + b_0|1\rangle) \otimes (a_1|0\rangle + b_1|1\rangle) \\
&= (a_0|0\rangle \otimes a_1|0\rangle) + (b_0|1\rangle \otimes a_1|0\rangle) + (a_0|0\rangle \otimes b_1|1\rangle) + (b_0|1\rangle \otimes b_1|1\rangle) \\
&= a_0a_1(|0\rangle \otimes |0\rangle) + b_0a_1(|1\rangle \otimes |0\rangle) + a_0b_1(|0\rangle \otimes |1\rangle) + b_0b_1(|1\rangle \otimes |1\rangle) \\
&= a_0a_1|00\rangle + b_0a_1|10\rangle + a_0b_1|01\rangle + b_0b_1|11\rangle.
\end{aligned}$$

В Непрерывные дроби и извлечение периода из результата измерения в алгоритме Шора

В общем случае, когда период r не является делителем 2^m , значение v , измеряемое на четвертом шаге алгоритма Шора, будет с большой вероятностью близко к какому-нибудь кратному $\frac{2^m}{r}$, скажем $j\frac{2^m}{r}$.

Итак, необходимо извлечь период r из измеренного значения v . Шор показал, что с большой вероятностью для некоторого целого j

$$\left| v - j\frac{2^m}{r} \right| < \frac{1}{2},$$

откуда следует, что

$$\left| \frac{v}{2^m} - \frac{j}{r} \right| < \frac{1}{2 \cdot 2^m} < \frac{1}{2M^2}.$$

Разность между двумя различными дробями $\frac{p}{q}$ и $\frac{p'}{q'}$, знаменатели которых меньше, чем M , ограничена:

$$\left| \frac{p}{q} - \frac{p'}{q'} \right| = \left| \frac{pq' - p'q}{qq'} \right| > \frac{1}{M^2}.$$

Таким образом, существует по крайней мере одна дробь $\frac{p}{q}$, знаменатель которой $q < M$, и для которой $\left| \frac{v}{2^m} - \frac{p}{q} \right| < \frac{1}{M^2}$. В высоковероятном случае, когда v отличается от $j\frac{2^m}{r}$ не больше, чем на $\frac{1}{2}$, эта дробь будет $\frac{j}{r}$. Единственная дробь со знаменателем меньшим чем M , отличающаяся от $\frac{v}{2^m}$ не больше, чем на $\frac{1}{M^2}$, может быть эффективно получена из продолжения непрерывных дробей (continued fraction expansion) следующим образом. Используя последовательности

$$\begin{aligned}
a_0 &= \left[\frac{v}{2^m} \right] \\
\epsilon_0 &= \frac{v}{2^m} - a_0 \\
a_n &= \left[\frac{1}{\epsilon_{n-1}} \right] \\
\epsilon_n &= \frac{1}{\epsilon_{n-1}} - a_n \\
p_0 &= a_0 \\
p_1 &= a_1a_0 + 1 \\
p_n &= a_n p_{n-1} + p_{n-2}
\end{aligned}$$

$$\begin{aligned}
q_0 &= 1 \\
q_1 &= a_1 \\
q_n &= a_n q_{n-1} + q_{n-2}
\end{aligned}$$

вычислим первую дробь $\frac{p_n}{q_n}$, такую, что $q_n < M \leq q_{n+1}$. Обоснование правильности такого метода можно найти в любой книге по теории чисел (например см [46]).

Итак, в нашем высоковероятном случае, когда $\frac{v}{2^m}$ отличается от $j\frac{2^m}{r}$ — некоего кратного $\frac{1}{r}$, не больше, чем на $\frac{1}{M^2}$, дробь, полученная вышеприведенной процедурой будет $\frac{j}{r}$, так как ее знаменатель меньше, чем M .

Значит, если j и r будут взаимно просты, знаменатель q от полученной дроби подойдет в качестве пробного периода.