Квантовые алгоритмы: возможности и ограничения. Лекция 9: Класс BQP

М. Вялый

Вычислительный центр
им. А.А.Дородницына
Российской Академии наук

Санкт-Петербург, 2011

План

Действие абелевой группы

Действие группы G на множестве X: гомоморфизм $G \to S(X)$ в группу биективных преобразований множества X. Другими словами, это функция $\alpha\colon G \times X \to X$, для которой выполняются условия

- $\alpha(g, \cdot)$ — взаимно однозначное отображение X на X;
- $\alpha(g_1 g_2, x) = \alpha(g_1, \alpha(g_2, x))$.

Часто обозначение функции опускается: $\alpha(g, x)$ обозначается как gx.

Стабилизатор: $S(x_0) = \{g \in G : gx_0 = x_0\}$.

Это подгруппа G.

Утверждение

Любая подгруппа $G < \mathbb{Z}^n$ изоморфна $\mathbb{Z}^k$. Поэтому для G существует базис образующих g_k:

$$G = \left\{ g : g = \sum_k z_k g_k \right\},\ z_k \text{ определены однозначно.}$$

Действие абелевой группы

Действие группы G на множестве X: гомоморфизм $G \to S(X)$ в группу биективных преобразований множества X. Другими словами, это функция $\alpha\colon G \times X \to X$, для которой выполняются условия

- $\alpha(g, \cdot)$ — взаимно однозначное отображение X на X;
- $\alpha(g_1 g_2, x) = \alpha(g_1, \alpha(g_2, x))$.

Часто обозначение функции опускается: $\alpha(g, x)$ обозначается как gx.
Стабилизатор: $S(x_0) = \{g \in G : gx_0 = x_0\}$.
Это подгруппа G.

Утверждение

Любая подгруппа $G < \mathbb{Z}^n$ изоморфна $\mathbb{Z}^k$. Поэтому для G существует базис образующих g_k:

$$G = \left\{ g : g = \sum_k z_k g_k \right\}, \; z_k \text{ определены однозначно.}$$

Действие абелевой группы

Действие группы G на множестве X: гомоморфизм $G \to S(X)$ в группу биективных преобразований множества X. Другими словами, это функция $\alpha\colon G \times X \to X$, для которой выполняются условия

- $\alpha(g, \cdot)$ — взаимно однозначное отображение X на X;
- $\alpha(g_1 g_2, x) = \alpha(g_1, \alpha(g_2, x))$.

Часто обозначение функции опускается: $\alpha(g, x)$ обозначается как gx.
Стабилизатор: $S(x_0) = \{g \in G : gx_0 = x_0\}$.
Это подгруппа G.

Утверждение

Любая подгруппа $G < \mathbb{Z}^n$ изоморфна $\mathbb{Z}^k$. Поэтому для G существует базис образующих g_k:

$$G = \left\{ g : g = \sum_k z_k g_k \right\}, \; z_k \text{ определены однозначно.}$$

Нахождение стабилизатора действия $\mathbb{Z}^n$

Нахождение стабилизатора действия $\mathbb{Z}^n$

Дано: эффективное действие группы $\mathbb{Z}^n$ на множестве $X = \{0,1\}^m$. Это означает, что задан алгоритм вычисления $\alpha(z,x)$, работающий за полиномиальное время. Кроме того, указан элемент $x_0 \in X$.
Найти: базисную систему образующих для $S(x_0)$.

Частный случай: нахождение периода

Действие $\mathbb{Z}^1$: $(n,x) \mapsto (a^n x \bmod q)$. Стабилизатор 1 состоит из чисел, кратных $\mathrm{per}_q(a)$.

Теорема

Существует полиномиальный квантовый алгоритм нахождения стабилизатора.

Нахождение стабилизатора действия $\mathbb{Z}^n$

Нахождение стабилизатора действия $\mathbb{Z}^n$

Дано: эффективное действие группы $\mathbb{Z}^n$ на множестве $X = \{0,1\}^m$. Это означает, что задан алгоритм вычисления $\alpha(z,x)$, работающий за полиномиальное время. Кроме того, указан элемент $x_0 \in X$.
Найти: базисную систему образующих для $S(x_0)$.

Частный случай: нахождение периода

Действие $\mathbb{Z}^1$: $(n,x) \mapsto (a^n x \bmod q)$. Стабилизатор 1 состоит из чисел, кратных $\mathrm{per}_q(a)$.

Теорема

Существует полиномиальный квантовый алгоритм нахождения стабилизатора.

Нахождение стабилизатора действия $\mathbb{Z}^n$

Нахождение стабилизатора действия $\mathbb{Z}^n$

Дано: эффективное действие группы $\mathbb{Z}^n$ на множестве $X = \{0,1\}^m$. Это означает, что задан алгоритм вычисления $\alpha(z,x)$, работающий за полиномиальное время. Кроме того, указан элемент $x_0 \in X$.
Найти: базисную систему образующих для $S(x_0)$.

Частный случай: нахождение периода

Действие $\mathbb{Z}^1$: $(n,x) \mapsto (a^n x \bmod q)$. Стабилизатор 1 состоит из чисел, кратных $\mathrm{per}_q(a)$.

Теорема

Существует полиномиальный квантовый алгоритм нахождения стабилизатора.

Об алгоритме нахождения стабилизатора

- Алгоритм ищет двойственную систему образующих для характеров: гомоморфизмов $S \to \mathbf{U}(1)$. Характер имеет вид

$$(z_1, \ldots, z_n) \overset{\varphi}{\mapsto} \exp\left(2\pi i \sum_j \varphi_j z_j\right),$$

  где $\varphi_j \leqslant 1$ — рациональные числа.
- Чтобы сгенерировать случайный характер, используется оценка собственных чисел операторов сдвига по каждой компоненте.
- Аналогично процедуре вычисления периода проверяется, что достаточно большое количество случайных характеров порождает всю группу.
- Образующие стабилизатора находятся из решения системы диофантовых линейных уравнений.

Об алгоритме нахождения стабилизатора

- Алгоритм ищет двойственную систему образующих для характеров: гомоморфизмов $S \to \mathbf{U}(1)$. Характер имеет вид

$$(z_1, \ldots, z_n) \overset{\varphi}{\mapsto} \exp\left(2\pi i \sum_j \varphi_j z_j\right),$$

  где $\varphi_j \leqslant 1$ — рациональные числа.
- Чтобы сгенерировать случайный характер, используется оценка собственных чисел операторов сдвига по каждой компоненте.
- Аналогично процедуре вычисления периода проверяется, что достаточно большое количество случайных характеров порождает всю группу.
- Образующие стабилизатора находятся из решения системы диофантовых линейных уравнений.

Об алгоритме нахождения стабилизатора

- Алгоритм ищет двойственную систему образующих для характеров: гомоморфизмов $S \to \mathbf{U}(1)$. Характер имеет вид

$$(z_1, \ldots, z_n) \overset{\varphi}{\mapsto} \exp\left(2\pi i \sum_j \varphi_j z_j\right),$$

  где $\varphi_j \leqslant 1$ — рациональные числа.
- Чтобы сгенерировать случайный характер, используется оценка собственных чисел операторов сдвига по каждой компоненте.
- Аналогично процедуре вычисления периода проверяется, что достаточно большое количество случайных характеров порождает всю группу.
- Образующие стабилизатора находятся из решения системы диофантовых линейных уравнений.

Об алгоритме нахождения стабилизатора

- Алгоритм ищет двойственную систему образующих для характеров: гомоморфизмов $S \to \mathbf{U}(1)$. Характер имеет вид

$$(z_1, \ldots, z_n) \overset{\varphi}{\mapsto} \exp\left(2\pi i \sum_j \varphi_j z_j\right),$$

  где $\varphi_j \leqslant 1$ — рациональные числа.
- Чтобы сгенерировать случайный характер, используется оценка собственных чисел операторов сдвига по каждой компоненте.
- Аналогично процедуре вычисления периода проверяется, что достаточно большое количество случайных характеров порождает всю группу.
- Образующие стабилизатора находятся из решения системы диофантовых линейных уравнений.

Дискретный логарифм

Определение

Пусть g — образующая $(\mathbb{Z}/p\mathbb{Z})^*$, p — простое. Дискретный логарифм $\log_g(a)$: наименьшее положительное k такое, что $g^k = a$.

Теорема

Существует полиномиальный квантовый алгоритм вычисления дискретного логарифма.

Доказательство

Задача сводится к нахождению стабилизатора. Действие $\mathbb{Z}^2$:

$$(z_1, z_2)(x) = g^{z_1} a^{z_2} x \pmod{p}.$$

Далее нужно найти в $S(1)$ элементы вида $(k, -1)$ (решением системы диофантовых линейных уравнений).

Дискретный логарифм

Определение

Пусть g — образующая $(\mathbb{Z}/p\mathbb{Z})^*$, p — простое. Дискретный логарифм $\log_g(a)$: наименьшее положительное k такое, что $g^k = a$.

Теорема

Существует полиномиальный квантовый алгоритм вычисления дискретного логарифма.

Доказательство

Задача сводится к нахождению стабилизатора. Действие $\mathbb{Z}^2$:

$$(z_1, z_2)(x) = g^{z_1} a^{z_2} x \pmod{p}.$$

Далее нужно найти в $S(1)$ элементы вида $(k, -1)$ (решением системы диофантовых линейных уравнений).

Дискретный логарифм

Определение

Пусть g — образующая $(\mathbb{Z}/p\mathbb{Z})^*$, p — простое. Дискретный логарифм $\log_g(a)$: наименьшее положительное k такое, что $g^k = a$.

Теорема

Существует полиномиальный квантовый алгоритм вычисления дискретного логарифма.

Доказательство

Задача сводится к нахождению стабилизатора. Действие $\mathbb{Z}^2$:

$$(z_1, z_2)(x) = g^{z_1} a^{z_2} x \pmod{p}.$$

Далее нужно найти в $S(1)$ элементы вида $(k, -1)$ (решением системы диофантовых линейных уравнений).

Открытые вопросы

Обращение перестановки

Дано: перестановка $\sigma\colon \{0,1\}^n \to \{0,1\}^n$ задана схемой вычисления.
Найти: обратную перестановку σ^{-1}.

Квантовый полиномиальный алгоритм неизвестен, существование сомнительно.

Стабилизатор действия S_n

Дано: действие симметрической группы $S_n \times \{0,1\}^n \to \{0,1\}^n$ задано схемой вычисления, элемент $x_0 \in \{0,1\}^n$.
Найти: $S(x_0)$.

К этой задаче сводится изоморфизм графов. Квантовый полиномиальный алгоритм неизвестен.
Уже для группы симметрий многоугольника D_n наилучший квантовый алгоритм поиска стабилизатора (Куперберг, 2003) требует $2^{O(\sqrt{\log n})}$.

Открытые вопросы

Обращение перестановки

Дано: перестановка $\sigma\colon \{0,1\}^n \to \{0,1\}^n$ задана схемой вычисления.
Найти: обратную перестановку σ^{-1}.

Квантовый полиномиальный алгоритм неизвестен, существование сомнительно.

Стабилизатор действия S_n

Дано: действие симметрической группы $S_n \times \{0,1\}^n \to \{0,1\}^n$ задано схемой вычисления, элемент $x_0 \in \{0,1\}^n$.
Найти: $S(x_0)$.

К этой задаче сводится изоморфизм графов. Квантовый полиномиальный алгоритм неизвестен.
Уже для группы симметрий многоугольника D_n наилучший квантовый алгоритм поиска стабилизатора (Куперберг, 2003) требует $2^{O(\sqrt{\log n})}$.

Открытые вопросы

Обращение перестановки

Дано: перестановка $\sigma\colon \{0,1\}^n \to \{0,1\}^n$ задана схемой вычисления.
Найти: обратную перестановку σ^{-1}.

Квантовый полиномиальный алгоритм неизвестен, существование сомнительно.

Стабилизатор действия S_n

Дано: действие симметрической группы $S_n \times \{0,1\}^n \to \{0,1\}^n$ задано схемой вычисления, элемент $x_0 \in \{0,1\}^n$.
Найти: $S(x_0)$.

К этой задаче сводится изоморфизм графов. Квантовый полиномиальный алгоритм неизвестен.

Уже для группы симметрий многоугольника D_n наилучший квантовый алгоритм поиска стабилизатора (Куперберг, 2003) требует $2^{O(\sqrt{\log n})}$.

Открытые вопросы

Обращение перестановки

Дано: перестановка $\sigma\colon \{0,1\}^n \to \{0,1\}^n$ задана схемой вычисления.
Найти: обратную перестановку σ^{-1}.

Квантовый полиномиальный алгоритм неизвестен, существование сомнительно.

Стабилизатор действия S_n

Дано: действие симметрической группы $S_n \times \{0,1\}^n \to \{0,1\}^n$ задано схемой вычисления, элемент $x_0 \in \{0,1\}^n$.
Найти: $S(x_0)$.

К этой задаче сводится изоморфизм графов. Квантовый полиномиальный алгоритм неизвестен.
Уже для группы симметрий многоугольника D_n наилучший квантовый алгоритм поиска стабилизатора (Куперберг, 2003) требует $2^{O(\sqrt{\log n})}$.

План

Подход структурной теории сложности

Главная идея

Рассматривать вместо задач классы сложности, т. е. семейства задач, которые решаются при использовании ресурсов указанного вида в указанном количестве.

Технически проще определять не задачи, а языки: подмножества слов в некотором алфавите. С языком L связана естественная алгоритмическая задача: дано слово x, проверить $x \in L$.

Подход структурной теории сложности

Главная идея

Рассматривать вместо задач классы сложности, т. е. семейства задач, которые решаются при использовании ресурсов указанного вида в указанном количестве.

Технически проще определять не задачи, а языки: подмножества слов в некотором алфавите. С языком L связана естественная алгоритмическая задача: дано слово x, проверить $x \in L$.

Примеры классов сложности (эффективные вычисления)

Пример: класс P

$L \in \mathrm{P}$ равносильно тому, что существует алгоритм проверки $x \in L$, работающий за полиномиальное время.
Класс P — наиболее популярная формализация понятия эффективного вычисления.

Пример: класс BPP

$L \in \mathrm{P}$ равносильно тому, что существует вероятностный алгоритм проверки $x \in L$, работающий за полиномиальное время с вероятностью ошибки $< 1/3$.
Класс BPP — наиболее разумная формализация понятия эффективного вычисления.

Утверждение (очевидное)

$\mathrm{P} \subseteq \mathrm{BPP}$.

Примеры классов сложности (эффективные вычисления)

Пример: класс P

$L \in \mathrm{P}$ равносильно тому, что существует алгоритм проверки $x \in L$, работающий за полиномиальное время.
Класс P — наиболее популярная формализация понятия эффективного вычисления.

Пример: класс BPP

$L \in \mathrm{P}$ равносильно тому, что существует вероятностный алгоритм проверки $x \in L$, работающий за полиномиальное время с вероятностью ошибки $< 1/3$.
Класс BPP — наиболее разумная формализация понятия эффективного вычисления.

Утверждение (очевидное)

$\mathrm{P} \subseteq \mathrm{BPP}$.

Примеры классов сложности (эффективные вычисления)

Пример: класс P

$L \in \mathrm{P}$ равносильно тому, что существует алгоритм проверки $x \in L$, работающий за полиномиальное время.
Класс P — наиболее популярная формализация понятия эффективного вычисления.

Пример: класс BPP

$L \in \mathrm{P}$ равносильно тому, что существует вероятностный алгоритм проверки $x \in L$, работающий за полиномиальное время с вероятностью ошибки $< 1/3$.
Класс BPP — наиболее разумная формализация понятия эффективного вычисления.

Утверждение (очевидное)

$\mathrm{P} \subseteq \mathrm{BPP}$.

Примеры классов сложности

Пример: класс NP

$L \in \mathrm{NP}$ равносильно тому, что существует вычислимый за полиномиальное время предикат (фунция в $\{0,1\}$) от двух переменных $R(x,y)$ и полином $p(\cdot)$ такие, что

если $x \in L$, то существует такой y, что $|y| \leqslant p(|x|)$ и $R(x,y)$;

если $x \notin L$, то для любого y из $|y| \leqslant p(|x|)$ следует $\neg R(x,y)$.

Операторы на классах: дополнения

Если $\mathcal{C}$ — класс сложности, то co-$\mathcal{C}$ — это класс, состоящий из языков вида $\bar{L}$, $L \in \mathcal{C}$.

Утверждение (очевидное)

$\mathrm{P} = \text{co-P}$, $\mathrm{BPP} = \text{co-BPP}$.

Гипотеза

$\mathrm{NP} \neq \text{co-NP}$.

Примеры классов сложности

Пример: класс NP

$L \in \mathrm{NP}$ равносильно тому, что существует вычислимый за полиномиальное время предикат (фунция в $\{0,1\}$) от двух переменных $R(x,y)$ и полином $p(\cdot)$ такие, что

если $x \in L$, то существует такой y, что $|y| \leqslant p(|x|)$ и $R(x,y)$;

если $x \notin L$, то для любого y из $|y| \leqslant p(|x|)$ следует $\neg R(x,y)$.

Операторы на классах: дополнения

Если $\mathcal{C}$ — класс сложности, то co-$\mathcal{C}$ — это класс, состоящий из языков вида $\bar{L}$, $L \in \mathcal{C}$.

Утверждение (очевидное)

$\mathrm{P} = \text{co-P}$, $\mathrm{BPP} = \text{co-BPP}$.

Гипотеза

$\mathrm{NP} \neq \text{co-NP}$.

Примеры классов сложности

Пример: класс NP

$L \in \mathrm{NP}$ равносильно тому, что существует вычислимый за полиномиальное время предикат (фунция в $\{0,1\}$) от двух переменных $R(x,y)$ и полином $p(\cdot)$ такие, что

если $x \in L$, то существует такой y, что $|y| \leqslant p(|x|)$ и $R(x,y)$;

если $x \notin L$, то для любого y из $|y| \leqslant p(|x|)$ следует $\neg R(x,y)$.

Операторы на классах: дополнения

Если $\mathcal{C}$ — класс сложности, то co-$\mathcal{C}$ — это класс, состоящий из языков вида $\bar{L}$, $L \in \mathcal{C}$.

Утверждение (очевидное)

$\mathrm{P} = \text{co-P}$, $\mathrm{BPP} = \text{co-BPP}$.

Гипотеза

$\mathrm{NP} \neq \text{co-NP}$.

Примеры классов сложности

Пример: класс NP

$L \in \mathrm{NP}$ равносильно тому, что существует вычислимый за полиномиальное время предикат (фунция в $\{0,1\}$) от двух переменных $R(x,y)$ и полином $p(\cdot)$ такие, что

если $x \in L$, то существует такой y, что $|y| \leqslant p(|x|)$ и $R(x,y)$;

если $x \notin L$, то для любого y из $|y| \leqslant p(|x|)$ следует $\neg R(x,y)$.

Операторы на классах: дополнения

Если $\mathcal{C}$ — класс сложности, то co-$\mathcal{C}$ — это класс, состоящий из языков вида $\bar{L}$, $L \in \mathcal{C}$.

Утверждение (очевидное)

$\mathrm{P} = \text{co-}\mathrm{P}$, $\mathrm{BPP} = \text{co-}\mathrm{BPP}$.

Гипотеза

$\mathrm{NP} \neq \text{co-}\mathrm{NP}$.

Класс суммирования PP

Пример: класс PP

$L \in \mathrm{PP}$ равносильно тому, что существует вероятностный алгоритм проверки $x \in L$, работающий за полиномиальное время и такой, что

если $x \in L$, то вероятность ответа «да» $> 1/2$;

если $x \notin L$, то вероятность ответа «да» $\leqslant 1/2$.

Утверждение

$L \in \mathrm{PP}$ тогда и только тогда, когда существует вычислимый за полиномиальное время предикат (фунция в $\{0,1\}$) от двух переменных $R(x,y)$ и полином $p(\cdot)$ такие, что

если $x \in L$, то количество таких y, что $|y| = p(|x|)$ и $R(x,y)$, больше $2^{p(|x|)}/2$;

если $x \notin L$, то количество таких y, что $|y| = p(|x|)$ и $R(x,y)$, не больше $2^{p(|x|)}/2$.

Класс суммирования PP

Пример: класс PP

$L \in \mathrm{PP}$ равносильно тому, что существует вероятностный алгоритм проверки $x \in L$, работающий за полиномиальное время и такой, что

если $x \in L$, то вероятность ответа «да» $> 1/2$;

если $x \notin L$, то вероятность ответа «да» $\leqslant 1/2$.

Утверждение

$L \in \mathrm{PP}$ тогда и только тогда, когда существует вычислимый за полиномиальное время предикат (фунция в $\{0,1\}$) от двух переменных $R(x,y)$ и полином $p(\cdot)$ такие, что

если $x \in L$, то количество таких y, что $|y| = p(|x|)$ и $R(x,y)$, больше $2^{p(|x|)}/2$;

если $x \notin L$, то количество таких y, что $|y| = p(|x|)$ и $R(x,y)$, не больше $2^{p(|x|)}/2$.

Соотношения между классами

Утверждение

$\mathrm{BPP} \subseteq \mathrm{PP}$; $\mathrm{NP} \subseteq \mathrm{PP}$.

Доказательство

Для BPP: $R(x, y)$ — предикат, который определяет ответ вероятностного алгоритма, если набор случайных битов равен y.

Пусть $L \in \mathrm{NP}$ и R, p – предикат и полином из определения NP.

Тогда L удовлетворяет определению класса PP с полиномом $\tilde{p}(n) = p(n) + 1$ и предикатом

$$\tilde{R}(x, yb) = \begin{cases} 1, & \text{если } b = 0; \\ R(x, y), & \text{если } b = 1. \end{cases}$$

Соотношения между классами

Утверждение

$\mathrm{BPP} \subseteq \mathrm{PP}$; $\mathrm{NP} \subseteq \mathrm{PP}$.

Доказательство

Для BPP: $R(x, y)$ — предикат, который определяет ответ вероятностного алгоритма, если набор случайных битов равен y.

Пусть $L \in \mathrm{NP}$ и R, p – предикат и полином из определения NP.

Тогда L удовлетворяет определению класса PP с полиномом $\tilde{p}(n) = p(n) + 1$ и предикатом

$$\tilde{R}(x, yb) = \begin{cases} 1, & \text{если } b = 0; \\ R(x, y), & \text{если } b = 1. \end{cases}$$

Соотношения между классами

Утверждение

$\mathrm{BPP} \subseteq \mathrm{PP}$; $\mathrm{NP} \subseteq \mathrm{PP}$.

Доказательство

Для BPP: $R(x, y)$ — предикат, который определяет ответ вероятностного алгоритма, если набор случайных битов равен y.
Пусть $L \in \mathrm{NP}$ и R, p – предикат и полином из определения NP.
Тогда L удовлетворяет определению класса PP с полиномом $\tilde{p}(n) = p(n) + 1$ и предикатом

$$\tilde{R}(x, yb) = \begin{cases} 1, & \text{если } b = 0; \\ R(x, y), & \text{если } b = 1. \end{cases}$$

Задачи суммирования лежат в PP

Теорема

Пусть $f_n\colon \{0,1\}^n \mapsto \mathbb{Z}$ — семейство функций, вычислимых за полиномиальное от n время.

Задача определения знака суммы

$$S_n = \sum_{x \in \{0,1\}^n} f_n(x)$$

лежит в классе PP (вход — описание алгоритма вычисления f_n и n в унарной системе).

Равносильная формулировка

Есть два семейства $f_n\colon \{0,1\}^n \mapsto \mathbb{N}$, $g_n\colon \{0,1\}^n \mapsto \mathbb{N}$, а вопрос о знаке разности

$$\sum_{x \in \{0,1\}^n} f_n(x) - \sum_{x \in \{0,1\}^n} g_n(x).$$

Задачи суммирования лежат в PP

Теорема

Пусть $f_n\colon \{0,1\}^n \mapsto \mathbb{Z}$ — семейство функций, вычислимых за полиномиальное от n время.

Задача определения знака суммы

$$S_n = \sum_{x\in\{0,1\}^n} f_n(x)$$

лежит в классе PP (вход — описание алгоритма вычисления f_n и n в унарной системе).

Равносильная формулировка

Есть два семейства $f_n\colon \{0,1\}^n \mapsto \mathbb{N}$, $g_n\colon \{0,1\}^n \mapsto \mathbb{N}$, а вопрос о знаке разности

$$\sum_{x\in\{0,1\}^n} f_n(x) - \sum_{x\in\{0,1\}^n} g_n(x).$$

Задачи суммирования лежат в PP (доказательство)

- Поскольку f_n, g_n вычислимы за полиномиальное время, длина L_n их записи ограничена полиномом от входа (считаем, что $L_n > 0$).
- По f_n, g_n строим предикат $R(x, u, v, b)$, где $x \in \{0,1\}^n$, $0 \leqslant u, v < 2^{L_n}$, $b \in \{0,1\}$.
- Предикат $R(x, u, v, b)$ истинен тогда и только тогда, когда

$$v \geqslant g_n(x) \text{ и } b = 0,\ u = 0, \quad \text{или } v \equiv 0 \pmod 2 \text{ и } b = 0,\ u > 0,$$
$$\text{или } u < f_n(x) \text{ и } b = 1,\ v = 0, \quad \text{или } u \equiv 0 \pmod 2 \text{ и } b = 1,\ v > 0.$$

- Вклад x в
  - ответы «да»: $(2^{L_n} - g_n(x)) + (2^{L_n} - f_n(x))$,
  - ответы «нет»: $(2^{L_n} - f_n(x)) + g_n(x) + (2^{L_n} - 1)$,
  - их разность: $2(v_f(x) - v_g(x))$.

Задачи суммирования лежат в PP (доказательство)

- Поскольку f_n, g_n вычислимы за полиномиальное время, длина L_n их записи ограничена полиномом от входа (считаем, что $L_n > 0$).
- По f_n, g_n строим предикат $R(x, u, v, b)$, где $x \in \{0,1\}^n$, $0 \leqslant u, v < 2^{L_n}$, $b \in \{0,1\}$.
- Предикат $R(x, u, v, b)$ истинен тогда и только тогда, когда

$$v \geqslant g_n(x) \text{ и } b = 0,\ u = 0, \quad \text{или } v \equiv 0 \pmod 2 \text{ и } b = 0,\ u > 0,$$
$$\text{или } u < f_n(x) \text{ и } b = 1,\ v = 0, \quad \text{или } u \equiv 0 \pmod 2 \text{ и } b = 1,\ v > 0.$$

- Вклад x в
  - ответ «да»: $\left(2^{L_n} - g_n(x)\right)\left(2^{L_n} - f_n(x)\right)\left(2^{L_n} - 1\right)$,
  - ответ «нет»: $\left(2^{L_n} - f_n(x)\right)\left(2^{L_n} - g_n(x)\right)\left(2^{L_n} - 1\right)$,
  - их разность: $2U_n(x) - 2^{L_n} f_n(x)$.

Задачи суммирования лежат в PP (доказательство)

- Поскольку f_n, g_n вычислимы за полиномиальное время, длина L_n их записи ограничена полиномом от входа (считаем, что $L_n > 0$).
- По f_n, g_n строим предикат $R(x, u, v, b)$, где $x \in \{0,1\}^n$, $0 \leqslant u, v < 2^{L_n}$, $b \in \{0,1\}$.
- Предикат $R(x, u, v, b)$ истинен тогда и только тогда, когда

$$v \geqslant g_n(x) \text{ и } b = 0,\ u = 0, \quad \text{или } v \equiv 0 \pmod 2 \text{ и } b = 0,\ u > 0,$$
$$\text{или } u < f_n(x) \text{ и } b = 1,\ v = 0, \quad \text{или } u \equiv 0 \pmod 2 \text{ и } b = 1,\ v > 0.$$

- Вклад x в
  - ответы «да» $f_n(x) + (2^{L_n} - g_n(x)) + (2^{L_n} - 1)2^{L_n}$;
  - ответы «нет» $(2^{L_n} - f_n(x)) + 0 + (2^{L_n} - 1)2^{L_n}$;
  - их разность $2(f_n(x) - g_n(x))$.

Задачи суммирования лежат в PP (доказательство)

- Поскольку f_n, g_n вычислимы за полиномиальное время, длина L_n их записи ограничена полиномом от входа (считаем, что $L_n > 0$).
- По f_n, g_n строим предикат $R(x, u, v, b)$, где $x \in \{0,1\}^n$, $0 \leqslant u, v < 2^{L_n}$, $b \in \{0,1\}$.
- Предикат $R(x, u, v, b)$ истинен тогда и только тогда, когда

$$v \geqslant g_n(x) \text{ и } b = 0,\ u = 0, \quad \text{или } v \equiv 0 \pmod 2 \text{ и } b = 0,\ u > 0,$$
$$\text{или } u < f_n(x) \text{ и } b = 1,\ v = 0, \quad \text{или } u \equiv 0 \pmod 2 \text{ и } b = 1,\ v > 0.$$

- Вклад x в
  - ответы «да» $f_n(x) + (2^{L_n} - g_n(x)) + (2^{L_n} - 1)2^{L_n}$;
  - ответы «нет» $(2^{L_n} - f_n(x)) + g_n(x) + (2^{L_n} - 1)2^{L_n}$;
  - разность $2(f_n(x) - g_n(x))$.

Задачи суммирования лежат в PP (доказательство)

- Поскольку f_n, g_n вычислимы за полиномиальное время, длина L_n их записи ограничена полиномом от входа (считаем, что $L_n > 0$).
- По f_n, g_n строим предикат $R(x, u, v, b)$, где $x \in \{0,1\}^n$, $0 \leqslant u, v < 2^{L_n}$, $b \in \{0,1\}$.
- Предикат $R(x, u, v, b)$ истинен тогда и только тогда, когда

$$v \geqslant g_n(x) \text{ и } b = 0,\ u = 0, \quad \text{или } v \equiv 0 \pmod 2 \text{ и } b = 0,\ u > 0,$$
$$\text{или } u < f_n(x) \text{ и } b = 1,\ v = 0, \quad \text{или } u \equiv 0 \pmod 2 \text{ и } b = 1,\ v > 0.$$

- Вклад x в
  - ответы «да» $f_n(x) + (2^{L_n} - g_n(x)) + (2^{L_n} - 1)2^{L_n}$;
  - ответы «нет» $(2^{L_n} - f_n(x)) + g_n(x) + (2^{L_n} - 1)2^{L_n}$;
  - разность $2\big(f_n(x) - g_n(x)\big)$.

Задачи суммирования лежат в PP (доказательство)

- Поскольку f_n, g_n вычислимы за полиномиальное время, длина L_n их записи ограничена полиномом от входа (считаем, что $L_n > 0$).
- По f_n, g_n строим предикат $R(x, u, v, b)$, где $x \in \{0,1\}^n$, $0 \leqslant u, v < 2^{L_n}$, $b \in \{0,1\}$.
- Предикат $R(x, u, v, b)$ истинен тогда и только тогда, когда

$$v \geqslant g_n(x) \text{ и } b = 0,\ u = 0, \quad \text{или } v \equiv 0 \pmod 2 \text{ и } b = 0,\ u > 0,$$
$$\text{или } u < f_n(x) \text{ и } b = 1,\ v = 0, \quad \text{или } u \equiv 0 \pmod 2 \text{ и } b = 1,\ v > 0.$$

- Вклад x в
  - ответы «да» $f_n(x) + (2^{L_n} - g_n(x)) + (2^{L_n} - 1)2^{L_n}$;
  - ответы «нет» $(2^{L_n} - f_n(x)) + g_n(x) + (2^{L_n} - 1)2^{L_n}$;
  - разность $2\big(f_n(x) - g_n(x)\big)$.

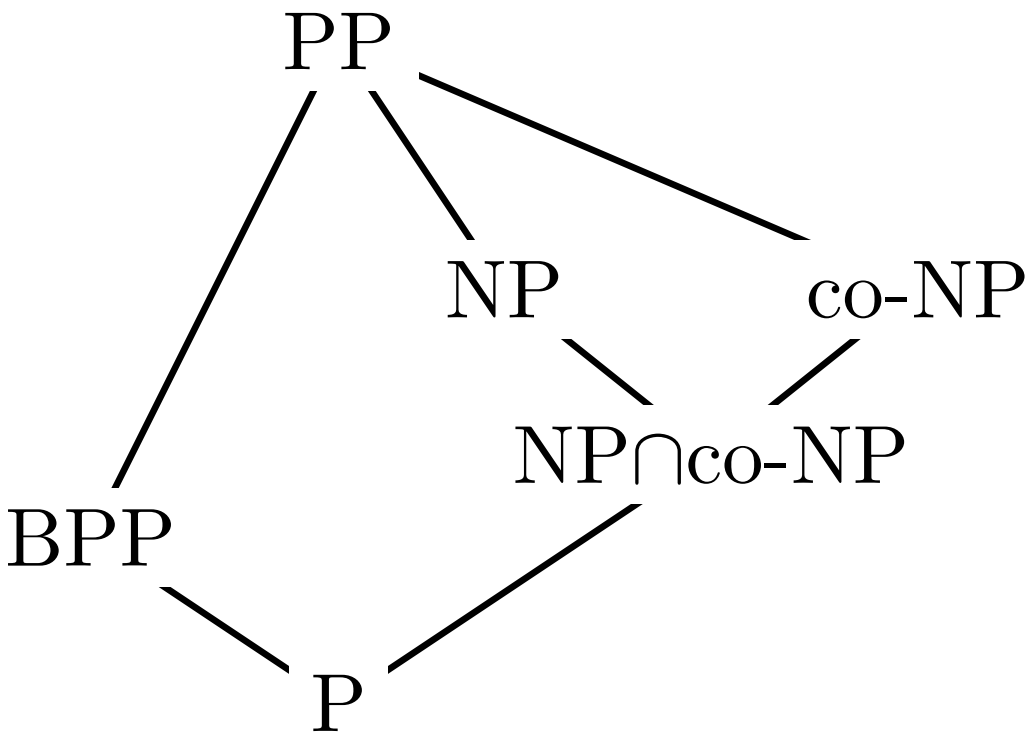
Задачи суммирования лежат в PP (доказательство)

- Поскольку f_n, g_n вычислимы за полиномиальное время, длина L_n их записи ограничена полиномом от входа (считаем, что $L_n > 0$).
- По f_n, g_n строим предикат $R(x, u, v, b)$, где $x \in \{0,1\}^n$, $0 \leqslant u, v < 2^{L_n}$, $b \in \{0,1\}$.
- Предикат $R(x, u, v, b)$ истинен тогда и только тогда, когда

$$v \geqslant g_n(x) \text{ и } b = 0,\ u = 0, \quad \text{или } v \equiv 0 \pmod 2 \text{ и } b = 0,\ u > 0,$$
$$\text{или } u < f_n(x) \text{ и } b = 1,\ v = 0, \quad \text{или } u \equiv 0 \pmod 2 \text{ и } b = 1,\ v > 0.$$

- Вклад x в
  - ответы «да» $f_n(x) + (2^{L_n} - g_n(x)) + (2^{L_n} - 1)2^{L_n}$;
  - ответы «нет» $(2^{L_n} - f_n(x)) + g_n(x) + (2^{L_n} - 1)2^{L_n}$;
  - разность $2\big(f_n(x) - g_n(x)\big)$.

Диаграмма включений классов



План

Класс BQP

Определение

$L \in \mathrm{BQP}$ равносильно тому, что существует квантовый алгоритм проверки $x \in L$, работающий за полиномиальное время с вероятностью ошибки $< 1/3$.
Этот класс формализует понятие задачи эффективно разрешимой с использованием квантового ресурса.

Утверждение

$\mathrm{BPP} \subseteq \mathrm{BQP}$.

Примеры задач предположительно находящихся в $\mathrm{BQP} \setminus \mathrm{BPP}$:

- нахождение периода;
- факторизация;
- дискретный логарифм.

Класс BQP

Определение

$L \in \mathrm{BQP}$ равносильно тому, что существует квантовый алгоритм проверки $x \in L$, работающий за полиномиальное время с вероятностью ошибки $< 1/3$.
Этот класс формализует понятие задачи эффективно разрешимой с использованием квантового ресурса.

Утверждение

$\mathrm{BPP} \subseteq \mathrm{BQP}$.

Примеры задач предположительно находящихся в $\mathrm{BQP} \setminus \mathrm{BPP}$:

- нахождение периода;
- факторизация;
- дискретный логарифм.

Класс BQP

Определение

$L \in \mathrm{BQP}$ равносильно тому, что существует квантовый алгоритм проверки $x \in L$, работающий за полиномиальное время с вероятностью ошибки $< 1/3$.
Этот класс формализует понятие задачи эффективно разрешимой с использованием квантового ресурса.

Утверждение

$\mathrm{BPP} \subseteq \mathrm{BQP}$.

Примеры задач предположительно находящихся в $\mathrm{BQP} \setminus \mathrm{BPP}$:

- нахождение периода;
- факторизация;
- дискретный логарифм.

В какие классы сложности попадают эти задачи?

Задаче вычисления функции $f : \{0,1\}^* \to \{0,1\}^*$ сопоставим язык $L_f = \{\langle x, j\rangle : j\text{-й бит } f(x) \text{ равен 1 и } |f(x)| \geqslant j\}$.
Если $L_f \in \mathcal{C}$ для некоторого класса сложности $\mathcal{C}$, то будем также говорить, что задача вычисления f лежит в классе $\mathcal{C}$.

Утверждение

Факторизация лежит в классе $\mathrm{NP} \cap \text{co-NP}$.

Идея доказательства

За полиномиальное время можно проверить корректность предлагаемого разложения данного числа на простые.
После этого легко определить как равенство нулю некоторого бита (co-NP), так и равенство единице (NP).

Замечание

Проверка простоты числа принадлежит P.

В какие классы сложности попадают эти задачи?

Задаче вычисления функции $f : \{0,1\}^* \to \{0,1\}^*$ сопоставим язык $L_f = \{\langle x, j\rangle : j\text{-й бит } f(x) \text{ равен 1 и } |f(x)| \geqslant j\}$.
Если $L_f \in \mathcal{C}$ для некоторого класса сложности $\mathcal{C}$, то будем также говорить, что задача вычисления f лежит в классе $\mathcal{C}$.

Утверждение

Факторизация лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

За полиномиальное время можно проверить корректность предлагаемого разложения данного числа на простые.
После этого легко определить как равенство нулю некоторого бита (co-NP), так и равенство единице (NP).

Замечание

Проверка простоты числа принадлежит P.

В какие классы сложности попадают эти задачи?

Задаче вычисления функции $f : \{0,1\}^* \to \{0,1\}^*$ сопоставим язык $L_f = \{\langle x, j\rangle : j\text{-й бит } f(x) \text{ равен 1 и } |f(x)| \geqslant j\}$.
Если $L_f \in \mathcal{C}$ для некоторого класса сложности $\mathcal{C}$, то будем также говорить, что задача вычисления f лежит в классе $\mathcal{C}$.

Утверждение

Факторизация лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

За полиномиальное время можно проверить корректность предлагаемого разложения данного числа на простые.
После этого легко определить как равенство нулю некоторого бита (co-NP), так и равенство единице (NP).

Замечание

Проверка простоты числа принадлежит P.

В какие классы сложности попадают эти задачи?

Задаче вычисления функции $f : \{0,1\}^* \to \{0,1\}^*$ сопоставим язык $L_f = \{\langle x, j\rangle : j\text{-й бит } f(x) \text{ равен 1 и } |f(x)| \geqslant j\}$.
Если $L_f \in \mathcal{C}$ для некоторого класса сложности $\mathcal{C}$, то будем также говорить, что задача вычисления f лежит в классе $\mathcal{C}$.

Утверждение

Факторизация лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

За полиномиальное время можно проверить корректность предлагаемого разложения данного числа на простые.
После этого легко определить как равенство нулю некоторого бита (co-NP), так и равенство единице (NP).

Замечание

Проверка простоты числа принадлежит P.

Период и дискретный логарифм

Утверждение

Вычисление периода лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

Аналогично факторизации. За полиномиальное время можно проверить, что число, представленное в виде разложения на простые множители, является периодом: $t = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ — период тогда и только тогда, когда $a^t \equiv 1 \pmod{q}$ и $a^{t/p_j} \not\equiv 1 \pmod{q}$.

Утверждение

Вычисление дискретного логарифма лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

Аналогично предыдущим случаям. $\log_g(a)$ — единственное положительное число, меньшее q и такое, что $g^s \equiv a \pmod{q}$.

Период и дискретный логарифм

Утверждение

Вычисление периода лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

Аналогично факторизации. За полиномиальное время можно проверить, что число, представленное в виде разложения на простые множители, является периодом: $t = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ — период тогда и только тогда, когда $a^t \equiv 1 \pmod{q}$ и $a^{t/p_j} \not\equiv 1 \pmod{q}$.

Утверждение

Вычисление дискретного логарифма лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

Аналогично предыдущим случаям. $\log_g(a)$ — единственное положительное число, меньшее q и такое, что $g^s \equiv a \pmod{q}$.

Период и дискретный логарифм

Утверждение

Вычисление периода лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

Аналогично факторизации. За полиномиальное время можно проверить, что число, представленное в виде разложения на простые множители, является периодом: $t = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ — период тогда и только тогда, когда $a^t \equiv 1 \pmod{q}$ и $a^{t/p_j} \not\equiv 1 \pmod{q}$.

Утверждение

Вычисление дискретного логарифма лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

Аналогично предыдущим случаям. $\log_g(a)$ — единственное положительное число, меньшее q и такое, что $g^s \equiv a \pmod{q}$.

Период и дискретный логарифм

Утверждение

Вычисление периода лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

Аналогично факторизации. За полиномиальное время можно проверить, что число, представленное в виде разложения на простые множители, является периодом: $t = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ — период тогда и только тогда, когда $a^t \equiv 1 \pmod{q}$ и $a^{t/p_j} \not\equiv 1 \pmod{q}$.

Утверждение

Вычисление дискретного логарифма лежит в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Идея доказательства

Аналогично предыдущим случаям. $\log_g(a)$ — единственное положительное число, меньшее q и такое, что $g^s \equiv a \pmod{q}$.

Самые трудные задачи в классе BQP

Определение

Задачу A будем называть трудной для класса $\mathcal{C}$, если любую задачу из этого класса можно свести к A, и полной для класса $\mathcal{C}$, если дополнительно к тому A лежит в $\mathcal{C}$.

Замечание

Чтобы говорить о BQP-полных задачах, языков недостаточно. Нужно рассматривать задачи с априорными условиями (promise problems). Такие задачи задаются двумя непересекающимися множествами слов: L_1 (ответ положительный) и L_0 (ответ отрицательный). Для остальных слов ответ не определен.

Самые трудные задачи в классе BQP

Определение

Задачу A будем называть трудной для класса $\mathcal{C}$, если любую задачу из этого класса можно свести к A, и полной для класса $\mathcal{C}$, если дополнительно к тому A лежит в $\mathcal{C}$.

Замечание

Чтобы говорить о BQP-полных задачах, языков недостаточно. Нужно рассматривать задачи с априорными условиями (promise problems). Такие задачи задаются двумя непересекающимися множествами слов: L_1 (ответ положительный) и L_0 (ответ отрицательный). Для остальных слов ответ не определен.

Пример полной задачи с априорной информацией

Неизвестно, существуют ли языки, полные в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Полная задача для класса $\mathrm{NP} \cap \text{co-}\mathrm{NP}$

Даны две КНФ C_0 и C_1. Известно (априорная информация), что ровно одна из них выполнима. Определить, какая именно.

Здесь использован хорошо известный факт.

Теорема

Задача выполнимости КНФ полна в классе NP.

Пример полной задачи с априорной информацией

Неизвестно, существуют ли языки, полные в классе $\mathrm{NP} \cap \text{co-}\mathrm{NP}$.

Полная задача для класса $\mathrm{NP} \cap \text{co-}\mathrm{NP}$

Даны две КНФ C_0 и C_1. Известно (априорная информация), что ровно одна из них выполнима. Определить, какая именно.

Здесь использован хорошо известный факт.

Теорема

Задача выполнимости КНФ полна в классе NP.

Пример BQP-полной задачи

Задача

Дано: описание квантовой схемы в стандартном базисе $\{\text{c-NOT}, H, K(\pi/4)\}$.
Известно: вероятность наблюдения 1 в первом кубите либо больше $2/3$, либо меньше $1/3$.
Выяснить: какой из случаев имеет место.

BQP-полнота этой задачи следует непосредственно из определений.

Пример BQP-полной задачи

Задача

Дано: описание квантовой схемы в стандартном базисе $\{\text{c-NOT}, H, K(\pi/4)\}$.
Известно: вероятность наблюдения 1 в первом кубите либо больше $2/3$, либо меньше $1/3$.
Выяснить: какой из случаев имеет место.

BQP-полнота этой задачи следует непосредственно из определений.

Оценка матричного элемента

Задача

Дано: описание квантовой схемы в стандартном базисе $\{\text{c-NOT}, H, K(\pi/4)\}$, реализующей оператор U.
Известно: действительная часть матричного элемента $\langle 0^n|U|0^n\rangle$ больше $1/3$, либо меньше $-1/3$.
Выяснить: какой из случаев имеет место.

Утверждение

Задача оценки матричного элемента BQP-полна.

Оценка матричного элемента

Задача

Дано: описание квантовой схемы в стандартном базисе $\{\text{c-NOT}, H, K(\pi/4)\}$, реализующей оператор U.
Известно: действительная часть матричного элемента $\langle 0^n|U|0^n\rangle$ больше $1/3$, либо меньше $-1/3$.
Выяснить: какой из случаев имеет место.

Утверждение

Задача оценки матричного элемента BQP-полна.

Оценка матричного элемента (продолжение)

Доказательство BQP-трудности

Пусть U — оператор, реализуемый схемой, для которой нужно оценить вероятность p наблюдения 1 в первом кубите.

Докажем, что для $V = U^\dagger \sigma_z[1] U$ выполняется равенство

$$\langle 0^n | V | 0^n \rangle = 1 - 2p.$$

Пусть $U|0^n\rangle = |0\rangle \otimes |\psi_0\rangle + |1\rangle \otimes |\psi_1\rangle$.
Тогда

$$\langle 0^n | V | 0^n \rangle = \langle 0^n | U^\dagger \sigma_z[1] U | 0^n \rangle = \langle \psi_0 | \psi_0 \rangle - \langle \psi_1 | \psi_1 \rangle = 1 - 2p.$$

Осталось заметить, что $p < 1/3$ влечет $\langle 0^n | V | 0^n \rangle > 1/3$, а $p > 2/3$ влечет $\langle 0^n | V | 0^n \rangle < -1/3$.

Оценка матричного элемента (продолжение)

Доказательство BQP-трудности

Пусть U — оператор, реализуемый схемой, для которой нужно оценить вероятность p наблюдения 1 в первом кубите.
Докажем, что для $V = U^\dagger \sigma_z[1] U$ выполняется равенство

$$\langle 0^n | V | 0^n \rangle = 1 - 2p.$$

Пусть $U|0^n\rangle = |0\rangle \otimes |\psi_0\rangle + |1\rangle \otimes |\psi_1\rangle$.
Тогда

$$\langle 0^n | V | 0^n \rangle = \langle 0^n | U^\dagger \sigma_z[1] U | 0^n \rangle = \langle \psi_0 | \psi_0 \rangle - \langle \psi_1 | \psi_1 \rangle = 1 - 2p.$$

Осталось заметить, что $p < 1/3$ влечет $\langle 0^n | V | 0^n \rangle > 1/3$, а $p > 2/3$ влечет $\langle 0^n | V | 0^n \rangle < -1/3$.

Оценка матричного элемента (продолжение)

Доказательство BQP-трудности

Пусть U — оператор, реализуемый схемой, для которой нужно оценить вероятность p наблюдения 1 в первом кубите.
Докажем, что для $V = U^\dagger \sigma_z[1] U$ выполняется равенство

$$\langle 0^n | V | 0^n \rangle = 1 - 2p.$$

Пусть $U|0^n\rangle = |0\rangle \otimes |\psi_0\rangle + |1\rangle \otimes |\psi_1\rangle$.
Тогда

$$\langle 0^n | V | 0^n \rangle = \langle 0^n | U^\dagger \sigma_z[1] U | 0^n \rangle = \langle \psi_0 | \psi_0 \rangle - \langle \psi_1 | \psi_1 \rangle = 1 - 2p.$$

Осталось заметить, что $p < 1/3$ влечет $\langle 0^n | V | 0^n \rangle > 1/3$, а $p > 2/3$ влечет $\langle 0^n | V | 0^n \rangle < -1/3$.

Оценка матричного элемента (продолжение)

Доказательство BQP-трудности

Пусть U — оператор, реализуемый схемой, для которой нужно оценить вероятность p наблюдения 1 в первом кубите.
Докажем, что для $V = U^\dagger \sigma_z[1] U$ выполняется равенство

$$\langle 0^n | V | 0^n \rangle = 1 - 2p.$$

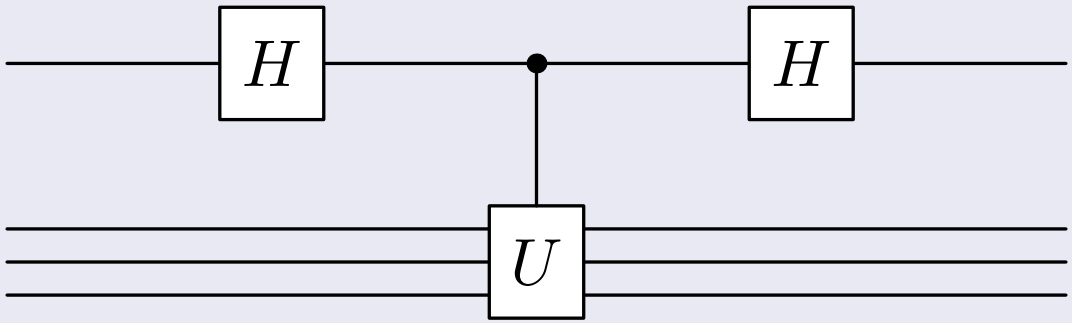
Пусть $U|0^n\rangle = |0\rangle \otimes |\psi_0\rangle + |1\rangle \otimes |\psi_1\rangle$.
Тогда

$$\langle 0^n | V | 0^n \rangle = \langle 0^n | U^\dagger \sigma_z[1] U | 0^n \rangle = \langle \psi_0 | \psi_0 \rangle - \langle \psi_1 | \psi_1 \rangle = 1 - 2p.$$

Осталось заметить, что $p < 1/3$ влечет $\langle 0^n | V | 0^n \rangle > 1/3$, а $p > 2/3$ влечет $\langle 0^n | V | 0^n \rangle < -1/3$.

Оценка матричного элемента (окончание)

Доказательство принадлежности BQP



Собственные числа и векторы U: λ_k, $|\psi_k\rangle$; $|0^n\rangle = \sum_k c_k |\psi_k\rangle$.

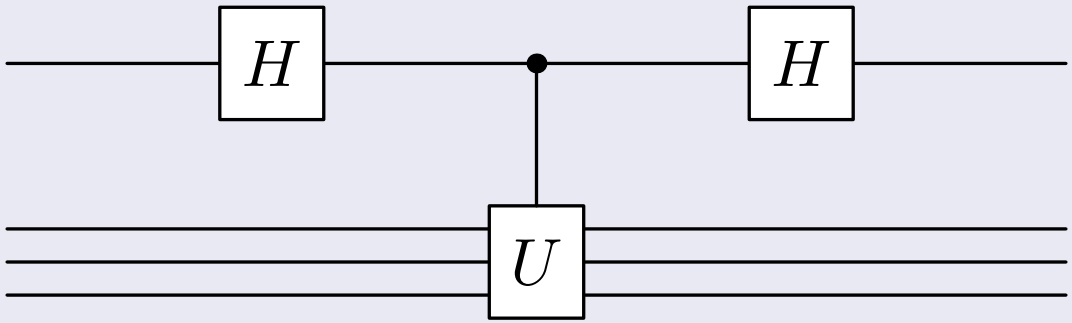
$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \sum_k |c_k|^2 \frac{1 - \mathrm{Re}\,\lambda_k}{2}$$

$$\langle 0^n|U|0^n\rangle = \sum_{j,k} c_j^* c_k \lambda_k \langle \psi_j|\psi_k\rangle = \sum_k |c_k|^2 \lambda_k$$

$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \frac{1}{2}(1 - \mathrm{Re}(\langle 0^n|U|0^n\rangle))$$

Оценка матричного элемента (окончание)

Доказательство принадлежности BQP



Собственные числа и векторы U: λ_k, $|\psi_k\rangle$; $|0^n\rangle = \sum_k c_k |\psi_k\rangle$.

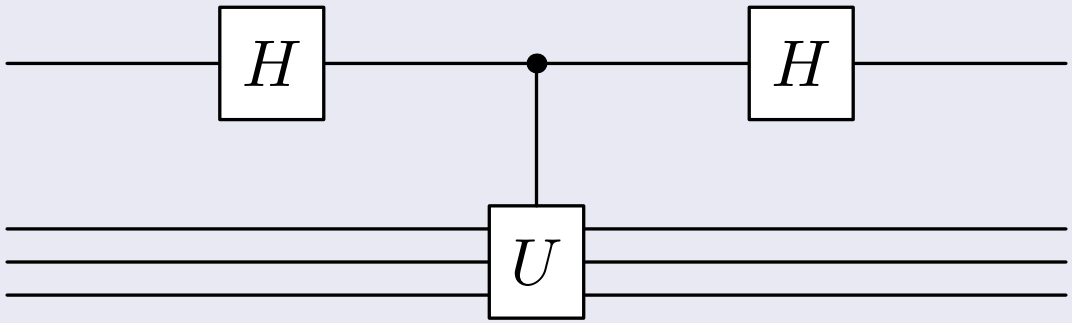
$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \sum_k |c_k|^2 \frac{1 - \mathrm{Re}\,\lambda_k}{2}$$

$$\langle 0^n|U|0^n\rangle = \sum_{j,k} c_j^* c_k \lambda_k \langle \psi_j|\psi_k\rangle = \sum_k |c_k|^2 \lambda_k$$

$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \frac{1}{2}(1 - \mathrm{Re}(\langle 0^n|U|0^n\rangle))$$

Оценка матричного элемента (окончание)

Доказательство принадлежности BQP



Собственные числа и векторы U: λ_k, $|\psi_k\rangle$; $|0^n\rangle = \sum_k c_k |\psi_k\rangle$.

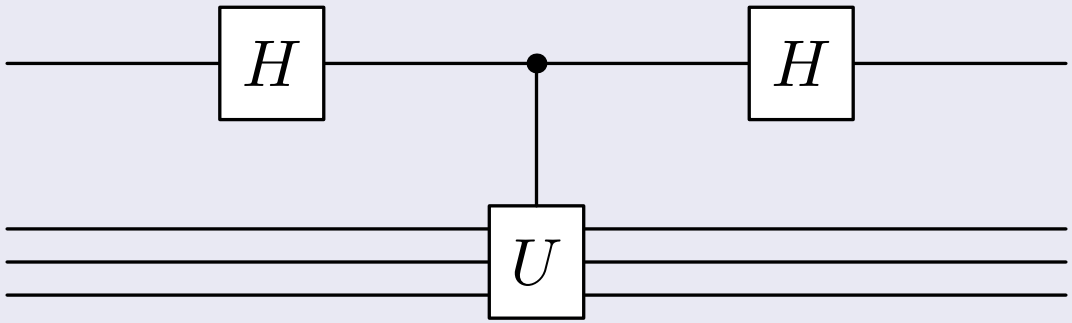
$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \sum_k |c_k|^2 \frac{1 - \mathrm{Re}\,\lambda_k}{2}$$

$$\langle 0^n|U|0^n\rangle = \sum_{j,k} c_j^* c_k \lambda_k \langle \psi_j|\psi_k\rangle = \sum_k |c_k|^2 \lambda_k$$

$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \frac{1}{2}(1 - \mathrm{Re}(\langle 0^n|U|0^n\rangle))$$

Оценка матричного элемента (окончание)

Доказательство принадлежности BQP



Собственные числа и векторы U: λ_k, $|\psi_k\rangle$; $|0^n\rangle = \sum_k c_k |\psi_k\rangle$.

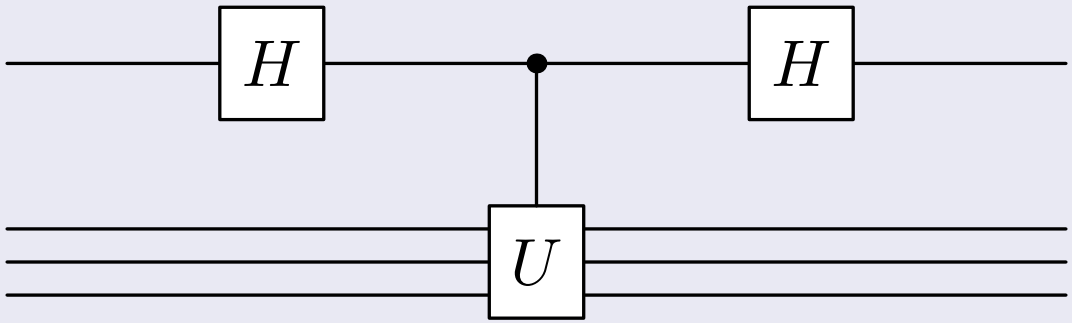
$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \sum_k |c_k|^2 \frac{1 - \mathrm{Re}\,\lambda_k}{2}$$

$$\langle 0^n|U|0^n\rangle = \sum_{j,k} c_j^* c_k \lambda_k \langle \psi_j|\psi_k\rangle = \sum_k |c_k|^2 \lambda_k$$

$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \frac{1}{2}(1 - \mathrm{Re}(\langle 0^n|U|0^n\rangle))$$

Оценка матричного элемента (окончание)

Доказательство принадлежности BQP



Собственные числа и векторы U: λ_k, $|\psi_k\rangle$; $|0^n\rangle = \sum_k c_k |\psi_k\rangle$.

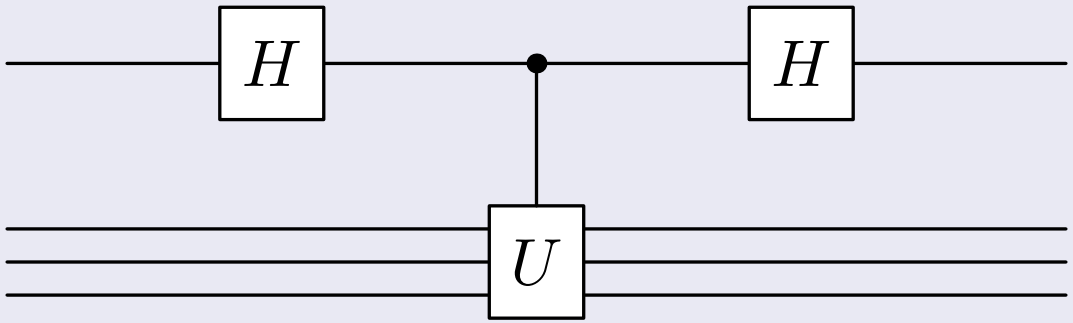
$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \sum_k |c_k|^2 \frac{1 - \mathrm{Re}\,\lambda_k}{2}$$

$$\langle 0^n|U|0^n\rangle = \sum_{j,k} c_j^* c_k \lambda_k \langle \psi_j|\psi_k\rangle = \sum_k |c_k|^2 \lambda_k$$

$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \frac{1}{2}(1 - \mathrm{Re}(\langle 0^n|U|0^n\rangle))$$

Оценка матричного элемента (окончание)

Доказательство принадлежности BQP



Собственные числа и векторы U: λ_k, $|\psi_k\rangle$; $|0^n\rangle = \sum_k c_k |\psi_k\rangle$.

$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \sum_k |c_k|^2 \frac{1 - \mathrm{Re}\,\lambda_k}{2}$$

$$\langle 0^n|U|0^n\rangle = \sum_{j,k} c_j^* c_k \lambda_k \langle \psi_j|\psi_k\rangle = \sum_k |c_k|^2 \lambda_k$$

$$\mathbf{Pr}(V|0^{n+1}\rangle, 1) = \frac{1}{2}(1 - \mathrm{Re}(\langle 0^n|U|0^n\rangle))$$

Задача Нилла – Лафламма

Формулировка задачи Нилла – Лафламма

Знаковая весовая функция

$$S(A, B, x, y) = \sum_{b:Ab=0} (-1)^{b^T Bb} x^{|b|} y^{n-|b|},$$

где A, B — матрицы над полем $\mathbb{F}_2$, b — вектор, $|b|$ — количество единиц (вес Хэмминга).
Дано: A — матрица с единицами на диагонали, k, ℓ — числа.
Известно: $|S(A, lwtr(A), k, \ell)| \geqslant (k^2 + \ell^2)^{n/2}/2$. Здесь $lwtr(A)$ получается из A заменой всех элементов на и над главной диагональю на нули.
Найти: знак $S(A, lwtr(A), k, \ell)$.

Теорема (Knill, Laflamme, 1999)

Задача Нилла – Лафламма полна в классе BQP.

Задача Нилла – Лафламма

Формулировка задачи Нилла – Лафламма

Знаковая весовая функция

$$S(A,B,x,y)=\sum_{b:Ab=0}(-1)^{b^TBb}x^{|b|}y^{n-|b|},$$

где A,B — матрицы над полем $\mathbb{F}_2$, b — вектор, $|b|$ — количество единиц (вес Хэмминга).
Дано: A — матрица с единицами на диагонали, k, ℓ — числа.
Известно: $|S(A,lwtr(A),k,\ell)|\geqslant(k^2+\ell^2)^{n/2}/2$. Здесь $lwtr(A)$ получается из A заменой всех элементов на и над главной диагональю на нули.
Найти: знак $S(A,lwtr(A),k,\ell)$.

Теорема (Knill, Laflamme, 1999)

Задача Нилла – Лафламма полна в классе BQP.

Задача Воцяна – Янцинга

Разреженная матрица — матрица, элементы которой заданы схемой вычисления, при этом известно, что количество ненулевых элементов в каждой строке и каждом столбце ограничено полиномом от логарифма размера матрицы.

Формулировка задачи оценки диагонального элемента

Дано: A — матрица размера N, заданная схемой вычисления, числа b, $m = \mathrm{polylog}(N)$, j, $\varepsilon = 1/\mathrm{polylog}(N)$, g.
Известно: A — разреженная, симметричная; $\|A\| \leqslant b$;
$|(A^m)_{jj} - g| \geqslant \varepsilon b^m$.
Найти: знак разности $(A^m)_{jj} - g$.

Теорема (Janzing, Wocjan, 2007)

Задача оценки диагонального элемента полна в классе BQP.

Задача Воцяна – Янцинга

Разреженная матрица — матрица, элементы которой заданы схемой вычисления, при этом известно, что количество ненулевых элементов в каждой строке и каждом столбце ограничено полиномом от логарифма размера матрицы.

Формулировка задачи оценки диагонального элемента

Дано: A — матрица размера N, заданная схемой вычисления, числа b, $m = \mathrm{polylog}(N)$, j, $\varepsilon = 1/\mathrm{polylog}(N)$, g.
Известно: A — разреженная, симметричная; $\|A\| \leqslant b$; $|(A^m)_{jj} - g| \geqslant \varepsilon b^m$.
Найти: знак разности $(A^m)_{jj} - g$.

Теорема (Janzing, Wocjan, 2007)

Задача оценки диагонального элемента полна в классе BQP.

Задача Воцяна – Янцинга

Разреженная матрица — матрица, элементы которой заданы схемой вычисления, при этом известно, что количество ненулевых элементов в каждой строке и каждом столбце ограничено полиномом от логарифма размера матрицы.

Формулировка задачи оценки диагонального элемента

Дано: A — матрица размера N, заданная схемой вычисления, числа b, $m = \mathrm{polylog}(N)$, j, $\varepsilon = 1/\mathrm{polylog}(N)$, g.
Известно: A — разреженная, симметричная; $\|A\| \leqslant b$; $|(A^m)_{jj} - g| \geqslant \varepsilon b^m$.
Найти: знак разности $(A^m)_{jj} - g$.

Теорема (Janzing, Wocjan, 2007)

Задача оценки диагонального элемента полна в классе BQP.

План

(Частичных) измерений достаточно

Nielsen, 2001; Fenner, Zhang, 2001. Результат Феннера – Жанга дает следующую модель универсального квантового вычисления.

- Приготовление кубитов в состоянии $|0\rangle$.
- Конечный набор измерительных приборов.
- Измерения производятся над не более чем тремя кубитами с двумя возможными исходами, т. е.
  1. Пространство (не более чем трех) кубитов разбивается в ортогональную сумму двух подпространств (определяется типом прибора измерения).
  2. Измерение состоит в том, что выбирается прибор, и вектор состояния проектируется на одно из подпространств с вероятностью, равной квадрату длины проекции вектора состояния на это подпространство.
  3. Проекция нормируется.
- Квантовая память (хранение многокубитных состояний, возникающих в процессе измерений).

(Частичных) измерений достаточно

Nielsen, 2001; Fenner, Zhang, 2001. Результат Феннера – Жанга дает следующую модель универсального квантового вычисления.

- Приготовление кубитов в состоянии $|0\rangle$.
- Конечный набор измерительных приборов.
- Измерения производятся над не более чем тремя кубитами с двумя возможными исходами, т. е.
  1. Пространство (не более чем трех) кубитов разбивается в ортогональную сумму двух подпространств (определяется выбором прибора измерения).
  2. Измерение состоит в том, что выбирается определенным образом одно из подпространств с вероятностью, равной квадрату длины проекции вектора состояния на это подпространство.
  3. Проекция нормируется.
- Квантовая память (хранение многокубитных состояний, возникающих в процессе измерений).

(Частичных) измерений достаточно

Nielsen, 2001; Fenner, Zhang, 2001. Результат Феннера – Жанга дает следующую модель универсального квантового вычисления.

- Приготовление кубитов в состоянии $|0\rangle$.
- Конечный набор измерительных приборов.
- Измерения производятся над не более чем тремя кубитами с двумя возможными исходами, т. е.
  1. Пространство (не более чем трех) кубитов разбивается в ортогональную сумму двух подпространств (определяется выбором прибора измерения).
  2. Измерение состоит в том, что выбирается проекция на одно из подпространств с вероятностью, равной квадрату длине проекции вектора состояния на это подпространство.
  3. Проекция нормируется.
- Квантовая память (хранение многокубитных состояний, возникающих в процессе измерений).

(Частичных) измерений достаточно

Nielsen, 2001; Fenner, Zhang, 2001. Результат Феннера – Жанга дает следующую модель универсального квантового вычисления.

- Приготовление кубитов в состоянии $|0\rangle$.
- Конечный набор измерительных приборов.
- Измерения производятся над не более чем тремя кубитами с двумя возможными исходами, т. е.
  1. Пространство (не более чем трех) кубитов разбивается в ортогональную сумму двух подпространств (определяется выбором прибора измерения).
  2. Измерение состоит в том, что выбирается проекция на одно из подпространств с вероятностью, равной квадрату длине проекции вектора состояния на это подпространство.
  3. Проекция нормируется.
- Квантовая память (хранение многокубитных состояний, возникающих в процессе измерений).

(Частичных) измерений достаточно

Nielsen, 2001; Fenner, Zhang, 2001. Результат Феннера – Жанга дает следующую модель универсального квантового вычисления.

- Приготовление кубитов в состоянии $|0\rangle$.
- Конечный набор измерительных приборов.
- Измерения производятся над не более чем тремя кубитами с двумя возможными исходами, т. е.
  1. Пространство (не более чем трех) кубитов разбивается в ортогональную сумму двух подпространств (определяется выбором прибора измерения).
  2. Измерение состоит в том, что выбирается проекция на одно из подпространств с вероятностью, равной квадрату длине проекции вектора состояния на это подпространство.
  3. Проекция нормируется.
- Квантовая память (хранение многокубитных состояний, возникающих в процессе измерений).

(Частичных) измерений достаточно

Nielsen, 2001; Fenner, Zhang, 2001. Результат Феннера – Жанга дает следующую модель универсального квантового вычисления.

- Приготовление кубитов в состоянии $|0\rangle$.
- Конечный набор измерительных приборов.
- Измерения производятся над не более чем тремя кубитами с двумя возможными исходами, т. е.
  1. Пространство (не более чем трех) кубитов разбивается в ортогональную сумму двух подпространств (определяется выбором прибора измерения).
  2. Измерение состоит в том, что выбирается проекция на одно из подпространств с вероятностью, равной квадрату длине проекции вектора состояния на это подпространство.
  3. Проекция нормируется.
- Квантовая память (хранение многокубитных состояний, возникающих в процессе измерений).

(Частичных) измерений достаточно

Nielsen, 2001; Fenner, Zhang, 2001. Результат Феннера – Жанга дает следующую модель универсального квантового вычисления.

- Приготовление кубитов в состоянии $|0\rangle$.
- Конечный набор измерительных приборов.
- Измерения производятся над не более чем тремя кубитами с двумя возможными исходами, т. е.
  1. Пространство (не более чем трех) кубитов разбивается в ортогональную сумму двух подпространств (определяется выбором прибора измерения).
  2. Измерение состоит в том, что выбирается проекция на одно из подпространств с вероятностью, равной квадрату длине проекции вектора состояния на это подпространство.
  3. Проекция нормируется.
- Квантовая память (хранение многокубитных состояний, возникающих в процессе измерений).

Адиабатическое квантовое вычисление

Предложено Farhi, Goldstone, Gutmann, Sipser, 2000 как алгоритм решения SAT.
Эквивалентность стандартной модели доказана Aharonov, van Dam, Kempe, Landau, Lloyd, Regev, 2004.

- Гамильтониан — эрмитов оператор. k-Локальный гамильтониан — сумма операторов, действующих на k кубитах, $k = O(1)$.
- Основное состояние гамильтониана — собственный вектор, отвечающий наименьшему собственному числу.
- Вычисление задается двумя локальными гамильтонианами
  1. начальным гамильтонианом H_0, основное состояние которого предполагается разложимым (легко подготовить);
  2. конечным гамильтонианом H_1, измерение которого дает ответ.
- Время работы пропорционально полиному от обратной точности $1/\varepsilon$, $\|H_1 - H_0\|$ и $\min \Delta(s)$, где $\Delta(s)$ — зазор между наименьшим и следующим по величине собственным числом гамильтониана $(1-s)H_0 + sH_1$.

Адиабатическое квантовое вычисление

Предложено Farhi, Goldstone, Gutmann, Sipser, 2000 как алгоритм решения SAT.
Эквивалентность стандартной модели доказана Aharonov, van Dam, Kempe, Landau, Lloyd, Regev, 2004.

- Гамильтониан — эрмитов оператор. k-Локальный гамильтониан — сумма операторов, действующих на k кубитах, $k = O(1)$.
- Основное состояние гамильтониана — собственный вектор, отвечающий наименьшему собственному числу.
- Вычисление задается двумя локальными гамильтонианами
  1. начальным гамильтонианом H_0, основное состояние которого предполагается разложимым (легко изготовить);
  2. конечным гамильтонианом H_1, основное состояние которого кодирует ответ.
- Время работы пропорционально полиному от обратной точности $1/\varepsilon$, $\|H_1 - H_0\|$ и $\min \Delta(s)$, где $\Delta(s)$ — зазор между наименьшим и следующим по величине собственным числом гамильтониана $(1-s)H_0 + sH_1$.

Адиабатическое квантовое вычисление

Предложено Farhi, Goldstone, Gutmann, Sipser, 2000 как алгоритм решения SAT.
Эквивалентность стандартной модели доказана Aharonov, van Dam, Kempe, Landau, Lloyd, Regev, 2004.

- Гамильтониан — эрмитов оператор. k-Локальный гамильтониан — сумма операторов, действующих на k кубитах, $k = O(1)$.
- Основное состояние гамильтониана — собственный вектор, отвечающий наименьшему собственному числу.
- Вычисление задается двумя локальными гамильтонианами
  1. начальным гамильтонианом H_0, основное состояние которого предполагается разложимым (легко изготовить);
  2. конечным гамильтонианом H_1, измерение основного состояния которого дает ответ.
- Время работы пропорционально полиному от обратной точности $1/\varepsilon$, $\|H_1 - H_0\|$ и $\min \Delta(s)$, где $\Delta(s)$ — зазор между наименьшим и следующим по величине собственным числом гамильтониана $(1-s)H_0 + sH_1$.

Адиабатическое квантовое вычисление

Предложено Farhi, Goldstone, Gutmann, Sipser, 2000 как алгоритм решения SAT.
Эквивалентность стандартной модели доказана Aharonov, van Dam, Kempe, Landau, Lloyd, Regev, 2004.

- Гамильтониан — эрмитов оператор. k-Локальный гамильтониан — сумма операторов, действующих на k кубитах, $k = O(1)$.
- Основное состояние гамильтониана — собственный вектор, отвечающий наименьшему собственному числу.
- Вычисление задается двумя локальными гамильтонианами
  1. начальным гамильтонианом H_0, основное состояние которого предполагается разложимым (легко изготовить);
  2. конечным гамильтонианом H_1, измерение основного состояния которого дает ответ.
- Время работы пропорционально полиному от обратной точности $1/\varepsilon$, $\|H_1 - H_0\|$ и $\min \Delta(s)$, где $\Delta(s)$ — зазор между наименьшим и следующим по величине собственным числом гамильтониана $(1-s)H_0 + sH_1$.

Адиабатическое квантовое вычисление

Предложено Farhi, Goldstone, Gutmann, Sipser, 2000 как алгоритм решения SAT.
Эквивалентность стандартной модели доказана Aharonov, van Dam, Kempe, Landau, Lloyd, Regev, 2004.

- Гамильтониан — эрмитов оператор. k-Локальный гамильтониан — сумма операторов, действующих на k кубитах, $k = O(1)$.
- Основное состояние гамильтониана — собственный вектор, отвечающий наименьшему собственному числу.
- Вычисление задается двумя локальными гамильтонианами
  1. начальным гамильтонианом H_0, основное состояние которого предполагается разложимым (легко изготовить);
  2. конечным гамильтонианом H_1, измерение основного состояния которого дает ответ.
- Время работы пропорционально полиному от обратной точности $1/\varepsilon$, $\|H_1 - H_0\|$ и $\min \Delta(s)$, где $\Delta(s)$ — зазор между наименьшим и следующим по величине собственным числом гамильтониана $(1-s)H_0 + sH_1$.

Адиабатическое квантовое вычисление

Предложено Farhi, Goldstone, Gutmann, Sipser, 2000 как алгоритм решения SAT.
Эквивалентность стандартной модели доказана Aharonov, van Dam, Kempe, Landau, Lloyd, Regev, 2004.

- Гамильтониан — эрмитов оператор. k-Локальный гамильтониан — сумма операторов, действующих на k кубитах, $k = O(1)$.
- Основное состояние гамильтониана — собственный вектор, отвечающий наименьшему собственному числу.
- Вычисление задается двумя локальными гамильтонианами
  1. начальным гамильтонианом H_0, основное состояние которого предполагается разложимым (легко изготовить);
  2. конечным гамильтонианом H_1, измерение основного состояния которого дает ответ.
- Время работы пропорционально полиному от обратной точности $1/\varepsilon$, $\|H_1 - H_0\|$ и $\min \Delta(s)$, где $\Delta(s)$ — зазор между наименьшим и следующим по величине собственным числом гамильтониана $(1-s)H_0 + sH_1$.

Топологическая квантовая теория поля

Унитарный топологический модулярный функтор: функтор из категории (поверхности, диффеоморфизмы) в категорию (унитарные пространства, унитарные операторы).

Freedman, Kitaev, Wang, 2000 показали, что значения функтора можно эффективно приближать квантовыми схемами (вводя подходящим образом кубитовую структуру).

Freedman, Larsen, Wang, 2000 нашли функтор, который универсален для квантового вычисления. Это означает, что квантовые схемы могут быть представлены как значения функтора от некоторого (явно заданного в некотором смысле) диффеоморфизма.

Aharonov, Jones, Landau, 2005: построили алгоритм приближенного (в аддитивном смысле) вычисления полинома Джонса (NP-трудная задача для точного вычисления).

Топологическая квантовая теория поля

Унитарный топологический модулярный функтор: функтор из категории (поверхности, диффеоморфизмы) в категорию (унитарные пространства, унитарные операторы).
Freedman, Kitaev, Wang, 2000 показали, что значения функтора можно эффективно приближать квантовыми схемами (вводя подходящим образом кубитовую структуру).
Freedman, Larsen, Wang, 2000 нашли функтор, который универсален для квантового вычисления. Это означает, что квантовые схемы могут быть представлены как значения функтора от некоторого (явно заданного в некотором смысле) диффеоморфизма.
Aharonov, Jones, Landau, 2005: построили алгоритм приближенного (в аддитивном смысле) вычисления полинома Джонса (NP-трудная задача для точного вычисления).

Топологическая квантовая теория поля

Унитарный топологический модулярный функтор: функтор из категории (поверхности, диффеоморфизмы) в категорию (унитарные пространства, унитарные операторы).
Freedman, Kitaev, Wang, 2000 показали, что значения функтора можно эффективно приближать квантовыми схемами (вводя подходящим образом кубитовую структуру).
Freedman, Larsen, Wang, 2000 нашли функтор, который универсален для квантового вычисления. Это означает, что квантовые схемы могут быть представлены как значения функтора от некоторого (явно заданного в некотором смысле) диффеоморфизма.
Aharonov, Jones, Landau, 2005: построили алгоритм приближенного (в аддитивном смысле) вычисления полинома Джонса (NP-трудная задача для точного вычисления).

Топологическая квантовая теория поля

Унитарный топологический модулярный функтор: функтор из категории (поверхности, диффеоморфизмы) в категорию (унитарные пространства, унитарные операторы).
Freedman, Kitaev, Wang, 2000 показали, что значения функтора можно эффективно приближать квантовыми схемами (вводя подходящим образом кубитовую структуру).
Freedman, Larsen, Wang, 2000 нашли функтор, который универсален для квантового вычисления. Это означает, что квантовые схемы могут быть представлены как значения функтора от некоторого (явно заданного в некотором смысле) диффеоморфизма.
Aharonov, Jones, Landau, 2005: построили алгоритм приближенного (в аддитивном смысле) вычисления полинома Джонса (NP-трудная задача для точного вычисления).

План

Верхняя оценка

Теорема

$\mathrm{BQP} \subseteq \mathrm{PP}$.

Идея доказательства

Матричный элемент оператора U, задаваемого схемой, выражается в виде суммы слагаемых, каждое из которых эффективно вычисляется.

Если $U = U_\ell \ldots U_1$, то

$$\langle 0^n | U | 0^n \rangle = \sum_{x_0,\ldots,x_\ell} \prod_{j=1}^{\ell} \langle x_{j-1} | U_j | x_j \rangle,$$

где $x_0 = x_\ell = 0^n$, $x_k \in \{0,1\}^n$.

Достаточно эффективной вычислимости матричных элементов базисных операторов. Наиболее прост случай базиса Ши $\{c\text{-NOT}, H\}$.

Все матричные элементы в этом случае можно представить рациональными числами со знаменателем 5.

Верхняя оценка

Теорема

$\mathrm{BQP} \subseteq \mathrm{PP}$.

Идея доказательства

Матричный элемент оператора U, задаваемого схемой, выражается в виде суммы слагаемых, каждое из которых эффективно вычисляется. Если $U = U_\ell \dots U_1$, то

$$\langle 0^n | U | 0^n \rangle = \sum_{x_0,\dots,x_\ell} \prod_{j=1}^{\ell} \langle x_{j-1} | U_j | x_j \rangle,$$

где $x_0 = x_\ell = 0^n$, $x_k \in \{0,1\}^n$.

Достаточно эффективной вычислимости матричных элементов базисных операторов. Наиболее прост случай базиса Ши $\{\text{c-NOT}, F\}$. Все матричные элементы в этом случае можно представить рациональными числами со знаменателем 5.

Верхняя оценка

Теорема

$\mathrm{BQP} \subseteq \mathrm{PP}$.

Идея доказательства

Матричный элемент оператора U, задаваемого схемой, выражается в виде суммы слагаемых, каждое из которых эффективно вычисляется. Если $U = U_\ell \dots U_1$, то

$$\langle 0^n | U | 0^n \rangle = \sum_{x_0,\dots,x_\ell} \prod_{j=1}^{\ell} \langle x_{j-1} | U_j | x_j \rangle,$$

где $x_0 = x_\ell = 0^n$, $x_k \in \{0,1\}^n$.

Достаточно эффективной вычислимости матричных элементов базисных операторов. Наиболее прост случай базиса Ши $\{\text{c-NOT}, F\}$. Все матричные элементы в этом случае можно представить рациональными числами со знаменателем 5.

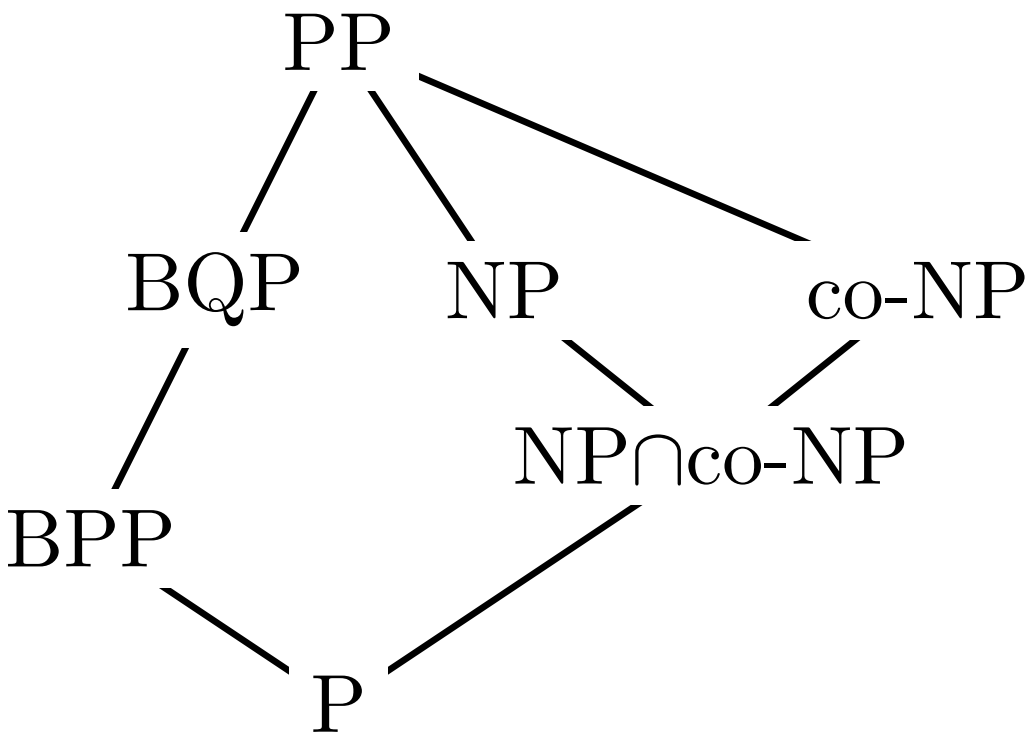
Случай стандартного базиса

Задача

Завершите доказательство включения $\mathrm{BQP} \subseteq \mathrm{PP}$, используя схемы в стандартном базисе $\{\text{c-NOT}, H, K(\pi/4)\}$.

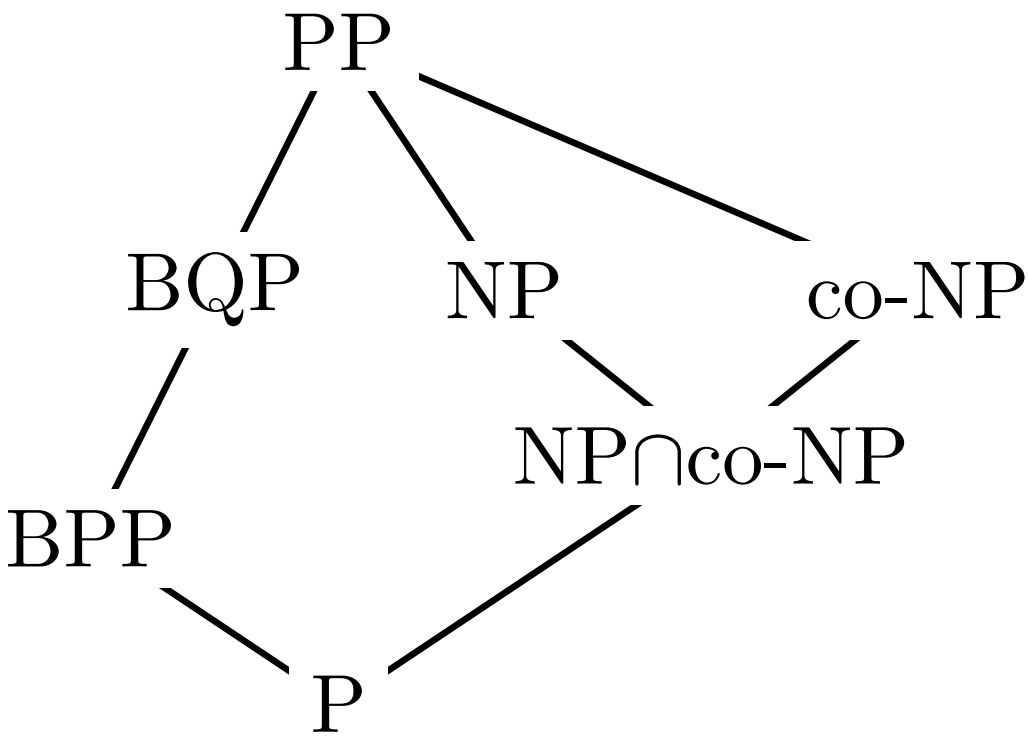
Указание: докажите, что за полиномиальное от n время можно построить рациональное число $q = r/10^k$, r, k — целые, для которого $|q - \sqrt{2}| < 2^{-n}$.

Уточненная диаграмма включений классов



Предполагается, что других включений в диаграмме нет (и тем самым, все включения строгие).

Уточненная диаграмма включений классов



Предполагается, что других включений в диаграмме нет (и тем самым, все включения строгие).

Оракулы

Релятивизация

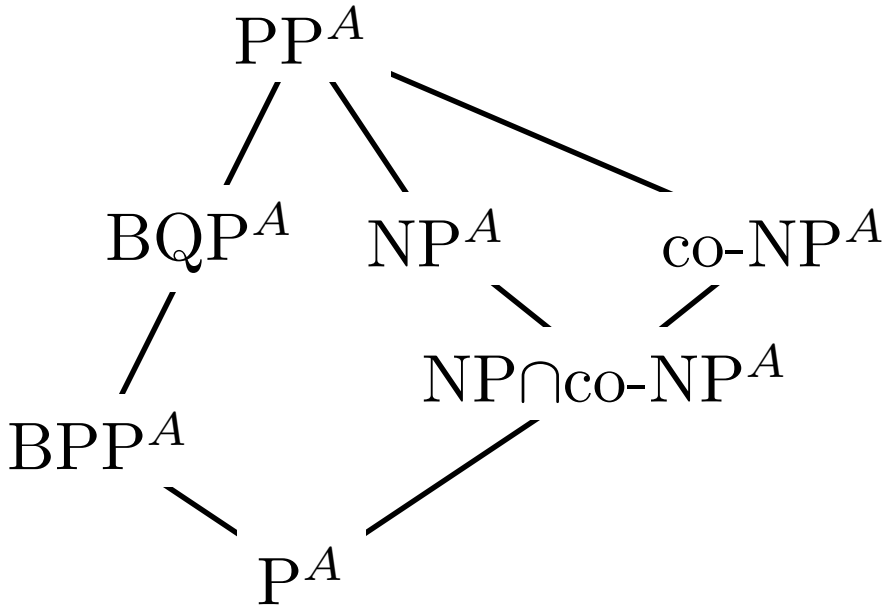
Если A — задача (язык, класс), а $\mathcal{C}$ — класс сложности, то $\mathcal{C}^A$ — класс сложности, состоящий из тех задач, которые решаются алгоритмами из класса $\mathcal{C}$, которые имеют доступ к оракулу (подпрограмме), решающему задачу A.

Оракулы

Релятивизация

Если A — задача (язык, класс), а $\mathcal{C}$ — класс сложности, то $\mathcal{C}^A$ — класс сложности, состоящий из тех задач, которые решаются алгоритмами из класса $\mathcal{C}$, которые имеют доступ к оракулу (подпрограмме), решающему задачу A.

Для любого оракула наша диаграмма классов сохраняется:



Оракульное разделение

Задача

Докажите, что существует такой оракул, относительно которого совпадают все указанные на диаграмме классы.

Теорема (Benett, Bernstein, Brassard, Vazirani, 1996)

Существует такой оракул A, что $\mathrm{BQP}^A \not\supseteq (\mathrm{NP} \cap \text{co-NP})^A$.

Доказательство теоремы о разделении основано на нижних оценках для задачи поиска Гровера. Оракул не строится явно, а неравенство доказывается для «почти всех» оракулов некоторого специального вида.

Теорема (Bernstein, Vazirani, 1993)

Существует такой оракул A, что $\mathrm{BQP}^A \supset \mathrm{BPP}^A$.

Оракульное разделение

Задача

Докажите, что существует такой оракул, относительно которого совпадают все указанные на диаграмме классы.

Теорема (Benett, Bernstein, Brassard, Vazirani, 1996)

Существует такой оракул A, что $\mathrm{BQP}^A \not\supseteq (\mathrm{NP} \cap \text{co-}\mathrm{NP})^A$.

Доказательство теоремы о разделении основано на нижних оценках для задачи поиска Гровера. Оракул не строится явно, а неравенство доказывается для «почти всех» оракулов некоторого специального вида.

Теорема (Bernstein, Vazirani, 1993)

Существует такой оракул A, что $\mathrm{BQP}^A \supset \mathrm{BPP}^A$.

Оракульное разделение

Задача

Докажите, что существует такой оракул, относительно которого совпадают все указанные на диаграмме классы.

Теорема (Benett, Bernstein, Brassard, Vazirani, 1996)

Существует такой оракул A, что $\mathrm{BQP}^A \not\supseteq (\mathrm{NP} \cap \text{co-}\mathrm{NP})^A$.

Доказательство теоремы о разделении основано на нижних оценках для задачи поиска Гровера. Оракул не строится явно, а неравенство доказывается для «почти всех» оракулов некоторого специального вида.

Теорема (Bernstein, Vazirani, 1993)

Существует такой оракул A, что $\mathrm{BQP}^A \supset \mathrm{BPP}^A$.

Оракульное разделение

Задача

Докажите, что существует такой оракул, относительно которого совпадают все указанные на диаграмме классы.

Теорема (Benett, Bernstein, Brassard, Vazirani, 1996)

Существует такой оракул A, что $\mathrm{BQP}^A \not\supseteq (\mathrm{NP} \cap \text{co-}\mathrm{NP})^A$.

Доказательство теоремы о разделении основано на нижних оценках для задачи поиска Гровера. Оракул не строится явно, а неравенство доказывается для «почти всех» оракулов некоторого специального вида.

Теорема (Bernstein, Vazirani, 1993)

Существует такой оракул A, что $\mathrm{BQP}^A \supset \mathrm{BPP}^A$.