Алгоритм Саймона

М. В. Волков

Уральский федеральный университет

Институт естественных наук и математики

кафедра алгебры и фундаментальной информатики

2020/2021 учебный год

Алгоритм Саймона: постановка задачи

Рассматривается функция $f : \mathbb{F}^n \to \mathbb{F}^n$, где, как обычно, $\mathbb{F} := \{0, 1\}$ — двухэлементное поле.

Алгоритм Саймона: постановка задачи

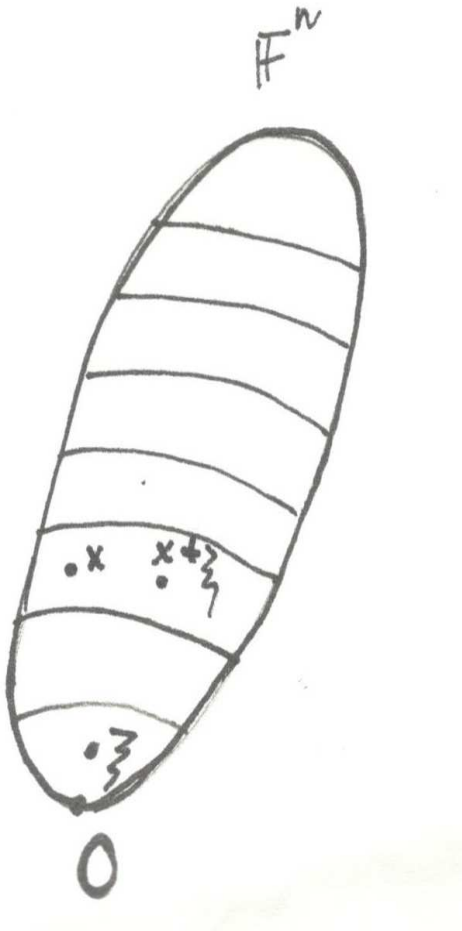
Рассматривается функция $f : \mathbb{F}^n \to \mathbb{F}^n$, где, как обычно, $\mathbb{F} := \{0, 1\}$ — двухэлементное поле. Об этой функции заранее известно, что существует «секретный» вектор $\xi \in \mathbb{F}^n$ такой, что:

1. $f(x) = f(x \oplus \xi)$ для всех $x \in \mathbb{F}^n$,
2. если $f(x) = f(y)$ для каких-то $x, y \in \mathbb{F}^n$, то $y = x \oplus \xi$.

Алгоритм Саймона: постановка задачи

Рассматривается функция $f : \mathbb{F}^n \to \mathbb{F}^n$, где, как обычно, $\mathbb{F} := \{0, 1\}$ — двухэлементное поле. Об этой функции заранее известно, что существует «секретный» вектор $\xi \in \mathbb{F}^n$ такой, что:

1. $f(x) = f(x \oplus \xi)$ для всех $x \in \mathbb{F}^n$,
2. если $f(x) = f(y)$ для каких-то $x, y \in \mathbb{F}^n$, то $y = x \oplus \xi$.

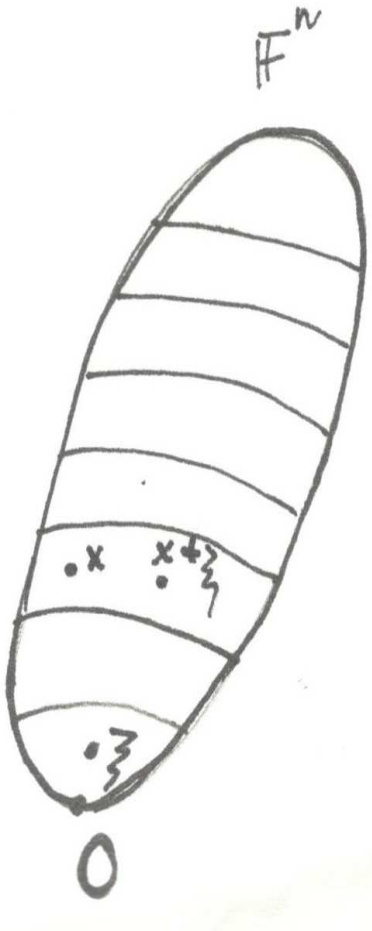


На алгебраическом языке условия 1 и 2 означают, что функция f определена на множестве смежных классов группы $(\mathbb{F}^n, \oplus)$ по подгруппе $\{\xi, 0\}$.

Алгоритм Саймона: постановка задачи

Рассматривается функция $f : \mathbb{F}^n \to \mathbb{F}^n$, где, как обычно, $\mathbb{F} := \{0, 1\}$ — двухэлементное поле. Об этой функции заранее известно, что существует «секретный» вектор $\xi \in \mathbb{F}^n$ такой, что:

1. $f(x) = f(x \oplus \xi)$ для всех $x \in \mathbb{F}^n$,
2. если $f(x) = f(y)$ для каких-то $x, y \in \mathbb{F}^n$, то $y = x \oplus \xi$.

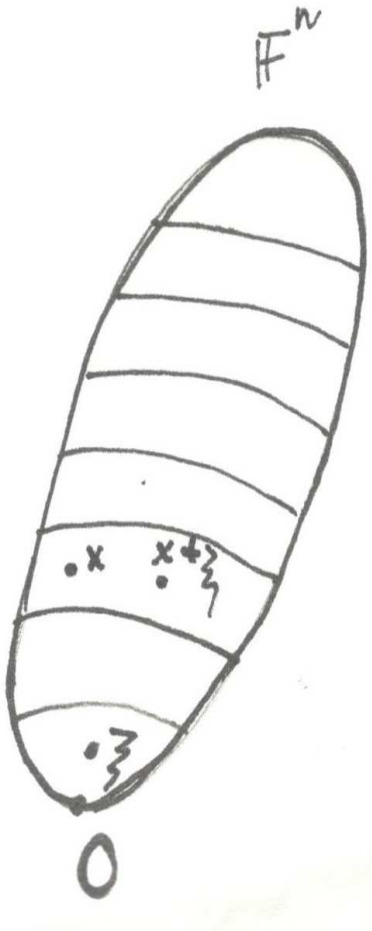


На алгебраическом языке условия 1 и 2 означают, что функция f определена на множестве смежных классов группы $(\mathbb{F}^n, \oplus)$ по подгруппе $\{\xi, 0\}$. Функция f неизвестна, но к ней можно обращаться как к черному ящику: задать аргумент $x \in \mathbb{F}^n$ и получить значение $f(x)$.

Алгоритм Саймона: постановка задачи

Рассматривается функция $f : \mathbb{F}^n \to \mathbb{F}^n$, где, как обычно, $\mathbb{F} := \{0, 1\}$ — двухэлементное поле. Об этой функции заранее известно, что существует «секретный» вектор $\xi \in \mathbb{F}^n$ такой, что:

1. $f(x) = f(x \oplus \xi)$ для всех $x \in \mathbb{F}^n$,
2. если $f(x) = f(y)$ для каких-то $x, y \in \mathbb{F}^n$, то $y = x \oplus \xi$.



На алгебраическом языке условия 1 и 2 означают, что функция f определена на множестве смежных классов группы $(\mathbb{F}^n, \oplus)$ по подгруппе $\{\xi, 0\}$. Функция f неизвестна, но к ней можно обращаться как к черному ящику: задать аргумент $x \in \mathbb{F}^n$ и получить значение $f(x)$. Требуется найти ξ.

Пример

Вот пример такой функции f для $n = 3$:

x	$f(x)$
000	101
001	010
010	000
011	110
100	000
101	110
110	101
111	010

Пример

Вот пример такой функции f для $n = 3$:

x	$f(x)$
000	101
001	010
010	000
011	110
100	000
101	110
110	101
111	010

«Секретный» вектор ξ здесь равен 110. Это можно увидеть, если рассмотреть все пары $x, y \in \mathbb{F}^3$, для которых $f(x) = f(y)$, и подсчитав $x \oplus y$ для каждой такой пары.

Разница между задачей Дойча–Йожи и задачей Саймона

Напомним, что в задаче Дойча–Йожи про функцию $f : \mathbb{F}^n \to \mathbb{F}$ известно, что она либо константная, либо сбалансированная, и надо узнать, какая из этих альтернатив реализуется.

Разница между задачей Дойча–Йожи и задачей Саймона

Напомним, что в задаче Дойча–Йожи про функцию $f : \mathbb{F}^n \to \mathbb{F}$ известно, что она либо константная, либо сбалансированная, и надо узнать, какая из этих альтернатив реализуется.

Алгоритм Дойча–Йожи решает эту задачу, один раз вызывая оператор U_f, в то время как любой классический детерминированный алгоритм для этой задачи требует $2^{n-1} + 1$ вызовов функции f в наихудшем случае.

Разница между задачей Дойча–Йожи и задачей Саймона

Напомним, что в задаче Дойча–Йожи про функцию $f : \mathbb{F}^n \to \mathbb{F}$ известно, что она либо константная, либо сбалансированная, и надо узнать, какая из этих альтернатив реализуется.

Алгоритм Дойча–Йожи решает эту задачу, один раз вызывая оператор U_f, в то время как любой классический детерминированный алгоритм для этой задачи требует $2^{n-1} + 1$ вызовов функции f в наихудшем случае. Таким образом, алгоритм Дойча–Йожи демонстрирует экспоненциальное превосходство над классическими *детерминированными* алгоритмами.

Разница между задачей Дойча–Йожи и задачей Саймона

Напомним, что в задаче Дойча–Йожи про функцию $f : \mathbb{F}^n \to \mathbb{F}$ известно, что она либо константная, либо сбалансированная, и надо узнать, какая из этих альтернатив реализуется.

Алгоритм Дойча–Йожи решает эту задачу, один раз вызывая оператор U_f, в то время как любой классический детерминированный алгоритм для этой задачи требует $2^{n-1} + 1$ вызовов функции f в наихудшем случае. Таким образом, алгоритм Дойча–Йожи демонстрирует экспоненциальное превосходство над классическими *детерминированными* алгоритмами.

В то же время несложно построить классический *вероятностный* алгоритм для задачи Дойча–Йожи, который дает ответ с вероятностью, сколь угодно близкой к 1, вызывая функцию f линейное от n число раз.

Разница между задачей Дойча–Йожи и задачей Саймона

Напомним, что в задаче Дойча–Йожи про функцию $f : \mathbb{F}^n \to \mathbb{F}$ известно, что она либо константная, либо сбалансированная, и надо узнать, какая из этих альтернатив реализуется.

Алгоритм Дойча–Йожи решает эту задачу, один раз вызывая оператор U_f, в то время как любой классический детерминированный алгоритм для этой задачи требует $2^{n-1} + 1$ вызовов функции f в наихудшем случае. Таким образом, алгоритм Дойча–Йожи демонстрирует экспоненциальное превосходство над классическими *детерминированными* алгоритмами.

В то же время несложно построить классический *вероятностный* алгоритм для задачи Дойча–Йожи, который дает ответ с вероятностью, сколь угодно близкой к 1, вызывая функцию f линейное от n число раз.

Можно доказать, однако, что любой классический вероятностный алгоритм для задачи Саймона требует $\Omega(2^{\sqrt{n}})$ вызовов функции f.

Разница между задачей Дойча–Йожи и задачей Саймона

Напомним, что в задаче Дойча–Йожи про функцию $f : \mathbb{F}^n \to \mathbb{F}$ известно, что она либо константная, либо сбалансированная, и надо узнать, какая из этих альтернатив реализуется.

Алгоритм Дойча–Йожи решает эту задачу, один раз вызывая оператор U_f, в то время как любой классический детерминированный алгоритм для этой задачи требует $2^{n-1} + 1$ вызовов функции f в наихудшем случае. Таким образом, алгоритм Дойча–Йожи демонстрирует экспоненциальное превосходство над классическими *детерминированными* алгоритмами.

В то же время несложно построить классический *вероятностный* алгоритм для задачи Дойча–Йожи, который дает ответ с вероятностью, сколь угодно близкой к 1, вызывая функцию f линейное от n число раз.

Можно доказать, однако, что любой классический вероятностный алгоритм для задачи Саймона требует $\Omega(2^{\sqrt{n}})$ вызовов функции f.

Даниель Саймон в 1994 г. предложил для нее квантовый *вероятностный* алгоритм, который вызывает оператор U_f линейное от n число раз.

Разница между задачей Дойча–Йожи и задачей Саймона

Напомним, что в задаче Дойча–Йожи про функцию $f : \mathbb{F}^n \to \mathbb{F}$ известно, что она либо константная, либо сбалансированная, и надо узнать, какая из этих альтернатив реализуется.

Алгоритм Дойча–Йожи решает эту задачу, один раз вызывая оператор U_f, в то время как любой классический детерминированный алгоритм для этой задачи требует $2^{n-1} + 1$ вызовов функции f в наихудшем случае. Таким образом, алгоритм Дойча–Йожи демонстрирует экспоненциальное превосходство над классическими *детерминированными* алгоритмами.

В то же время несложно построить классический *вероятностный* алгоритм для задачи Дойча–Йожи, который дает ответ с вероятностью, сколь угодно близкой к 1, вызывая функцию f линейное от n число раз.

Можно доказать, однако, что любой классический вероятностный алгоритм для задачи Саймона требует $\Omega(2^{\sqrt{n}})$ вызовов функции f.

Даниель Саймон в 1994 г. предложил для нее квантовый *вероятностный* алгоритм, который вызывает оператор U_f линейное от n число раз. Таким образом, алгоритм Саймона демонстрирует экспоненциальное превосходство над классическими *вероятностными* алгоритмами.

Алгоритм Саймона работает с квантовой системой $H := H_{2^n} \otimes H_{2^n}$ (напомним, что $H_{2^n} = \underbrace{H_2 \otimes \cdots \otimes H_2 \otimes H_2}_{n \text{ раз}}$ — это n-я тензорная степень двумерного гильбертова пространства H_2).

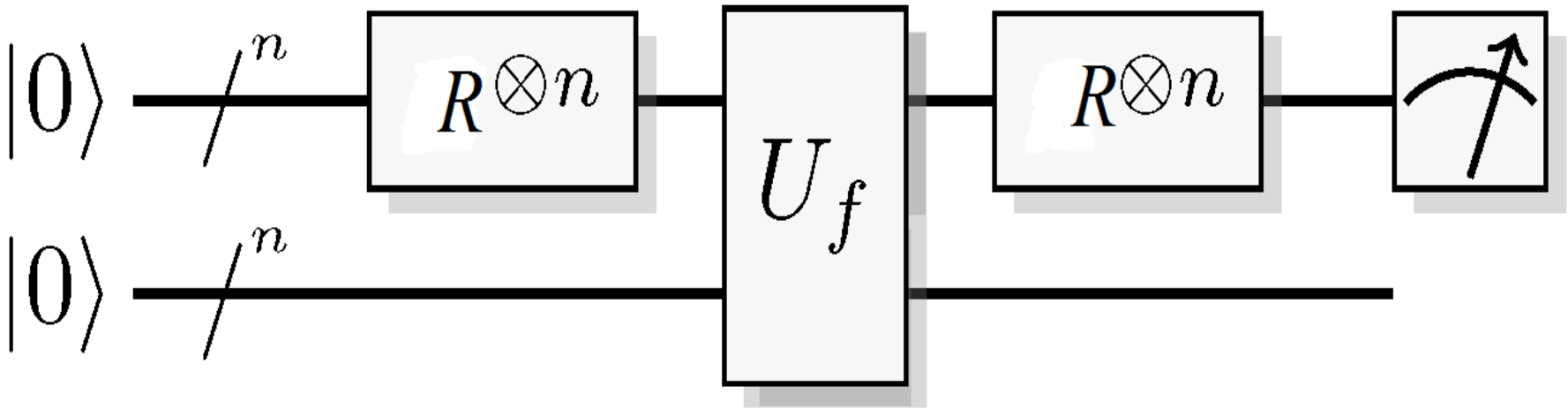
Алгоритм Саймона работает с квантовой системой $H := H_{2^n} \otimes H_{2^n}$ (напомним, что $H_{2^n} = \underbrace{H_2 \otimes \cdots \otimes H_2 \otimes H_2}_{n \text{ раз}}$ — это n-я тензорная степень двумерного гильбертова пространства H_2).
Вместо $|a\rangle \otimes |b\rangle$ пишем $|a\rangle|b\rangle$ и ссылаемся на $|a\rangle$ и соответственно $|b\rangle$ как на *содержимое* первого и соответственно второго регистра.

Алгоритм Саймона работает с квантовой системой $H := H_{2^n} \otimes H_{2^n}$ (напомним, что $H_{2^n} = \underbrace{H_2 \otimes \cdots \otimes H_2 \otimes H_2}_{n \text{ раз}}$ — это n-я тензорная степень двумерного гильбертова пространства H_2).
Вместо $|a\rangle \otimes |b\rangle$ пишем $|a\rangle|b\rangle$ и ссылаемся на $|a\rangle$ и соответственно $|b\rangle$ как на *содержимое* первого и соответственно второго регистра.

Алгоритм Саймона

1. Инициализируем H в состояние $|0\rangle|0\rangle$.
2. Применяем преобразование Адамара R_{2^n} к первому регистру.
3. Применяем оператор $U_f : |a\rangle|b\rangle \mapsto |a\rangle|f(a) \oplus b\rangle$, где $\oplus$ — сумма mod 2.
4. Применяем преобразование Адамара R_{2^n} к первому регистру.
5. Измеряя первый регистр, находим вектор, ортогональный вектору ξ.
6. Повторяем шаги 1–5, пока не найдем $n-1$ линейно независимых векторов $u_1, \ldots, u_{n-1}$, ортогональных вектору ξ.
7. Классически решаем систему линейных уравнений $\begin{cases} u_1 \cdot \xi = 0 \\ \ldots\ldots\ldots\ldots \\ u_{n-1} \cdot \xi = 0 \end{cases}$.

Алгоритм Саймона: схема



Корректность алгоритма Саймона

Вычислим состояние системы на каждом шаге алгоритма Саймона.

Корректность алгоритма Саймона

Вычислим состояние системы на каждом шаге алгоритма Саймона.
После шага 1 состояние $|0\rangle|0\rangle$.

Корректность алгоритма Саймона

Вычислим состояние системы на каждом шаге алгоритма Саймона.
После шага 1 состояние $|0\rangle|0\rangle$.

На шаге 2 к первому регистру применяется преобразование Адамара.
Получится $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle|0\rangle$.

Корректность алгоритма Саймона

Вычислим состояние системы на каждом шаге алгоритма Саймона.
После шага 1 состояние $|0\rangle|0\rangle$.

На шаге 2 к первому регистру применяется преобразование Адамара.
Получится $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle|0\rangle$.

На шаге 3 применяется оператор U_f. Получим $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle|f(k)\rangle$.

Корректность алгоритма Саймона

Вычислим состояние системы на каждом шаге алгоритма Саймона.
После шага 1 состояние $|0\rangle|0\rangle$.

На шаге 2 к первому регистру применяется преобразование Адамара.
Получится $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle|0\rangle$.

На шаге 3 применяется оператор U_f. Получим $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle|f(k)\rangle$.

На шаге 4 снова «проадамарим» первый регистр. Напомним, что $R_{2^n}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{k\cdot u}|u\rangle$. Получим

$$\frac{1}{2^n} \sum_{u=0}^{2^n-1} |u\rangle \sum_{k=0}^{2^n-1} (-1)^{k\cdot u}|f(k)\rangle = \frac{1}{2^n} \sum_{u=0}^{2^n-1} |u\rangle \sum_{m} |f(m)\rangle (-1)^{m\cdot u} \left(1 + (-1)^{\xi\cdot u}\right).$$

Здесь m — представитель одного из 2^{n-1} смежных классов, тогда второй элемент того же класса равен $m \oplus \xi$. Учтем, что $f(m) = f(m \oplus \xi)$, а

$$(-1)^{m\cdot u} + (-1)^{(m\oplus\xi)\cdot u} = (-1)^{m\cdot u} + (-1)^{m\cdot u}(-1)^{\xi\cdot u} = (-1)^{m\cdot u}\left(1 + (-1)^{\xi\cdot u}\right).$$

Корректность алгоритма Саймона

Вычислим состояние системы на каждом шаге алгоритма Саймона. После шага 1 состояние $|0\rangle|0\rangle$.

На шаге 2 к первому регистру применяется преобразование Адамара. Получится $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle|0\rangle$.

На шаге 3 применяется оператор U_f. Получим $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |k\rangle|f(k)\rangle$.

На шаге 4 снова «проадамарим» первый регистр. Напомним, что $R_{2^n}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{k\cdot u}|u\rangle$. Получим

$$\frac{1}{2^n} \sum_{u=0}^{2^n-1} |u\rangle \sum_{k=0}^{2^n-1} (-1)^{k\cdot u}|f(k)\rangle = \frac{1}{2^n} \sum_{u=0}^{2^n-1} |u\rangle \sum_{m} |f(m)\rangle (-1)^{m\cdot u} \left(1 + (-1)^{\xi\cdot u}\right).$$

Здесь m — представитель одного из 2^{n-1} смежных классов, тогда второй элемент того же класса равен $m \oplus \xi$. Учтем, что $f(m) = f(m \oplus \xi)$, а

$$(-1)^{m\cdot u} + (-1)^{(m\oplus\xi)\cdot u} = (-1)^{m\cdot u} + (-1)^{m\cdot u}(-1)^{\xi\cdot u} = (-1)^{m\cdot u}\left(1 + (-1)^{\xi\cdot u}\right).$$

Видно, что если $\xi \cdot u \neq 0$, то $1 + (-1)^{\xi\cdot u} = 0$, откуда коэффициент при соответствующем $|u\rangle$ нулевой. Остаются только $|v\rangle$ такие, что $\xi \cdot v = 0$.

Корректность алгоритма Саймона (2)

На шаге 5 мы замеряем первый регистр. Вероятность того, что при этом получится какой-то вектор v, равна, как мы знаем, $\langle x|P_v|x\rangle$, где P_v — самосопряженный оператор ортопроектирования на ось вектора v, а

$$|x\rangle := \frac{1}{2^n}\sum_{u=0}^{2^n-1}|u\rangle\sum_m|f(m)\rangle(-1)^{m\cdot u}\left(1+(-1)^{\xi\cdot u}\right).$$

Корректность алгоритма Саймона (2)

На шаге 5 мы замеряем первый регистр. Вероятность того, что при этом получится какой-то вектор v, равна, как мы знаем, $\langle x|P_v|x\rangle$, где P_v — самосопряженный оператор ортопроектирования на ось вектора v, а

$$|x\rangle := \frac{1}{2^n}\sum_{u=0}^{2^n-1}|u\rangle\sum_m|f(m)\rangle(-1)^{m\cdot u}\left(1+(-1)^{\xi\cdot u}\right).$$

Вычисляя, получим

$$\langle x|P_v|x\rangle = \frac{1}{2^{2n}}\sum_m\langle f(m)\,|\,f(m)\rangle\left|1+(-1)^{\xi\cdot v}\right|^2 = \begin{cases}\frac{1}{2^{n-1}}, & \text{если } v\perp\xi,\\ 0, & \text{если } v\not\perp\xi.\end{cases}$$

Корректность алгоритма Саймона (2)

На шаге 5 мы замеряем первый регистр. Вероятность того, что при этом получится какой-то вектор v, равна, как мы знаем, $\langle x|P_v|x\rangle$, где P_v — самосопряженный оператор ортопроектирования на ось вектора v, а

$$|x\rangle := \frac{1}{2^n}\sum_{u=0}^{2^n-1}|u\rangle\sum_m|f(m)\rangle(-1)^{m\cdot u}\left(1+(-1)^{\xi\cdot u}\right).$$

Вычисляя, получим

$$\langle x|P_v|x\rangle = \frac{1}{2^{2n}}\sum_m\langle f(m)\,|\,f(m)\rangle\left|1+(-1)^{\xi\cdot v}\right|^2 = \begin{cases}\frac{1}{2^{n-1}}, & \text{если } v\perp\xi,\\ 0, & \text{если } v\not\perp\xi.\end{cases}$$

Следовательно, будут наблюдаться только вектора, ортогональные ξ, причем каждый из таких векторов будет возникать с вероятностью $\frac{1}{2^{n-1}}$.

Корректность алгоритма Саймона (2)

На шаге 5 мы замеряем первый регистр. Вероятность того, что при этом получится какой-то вектор v, равна, как мы знаем, $\langle x|P_v|x\rangle$, где P_v — самосопряженный оператор ортопроектирования на ось вектора v, а

$$|x\rangle := \frac{1}{2^n}\sum_{u=0}^{2^n-1}|u\rangle\sum_m|f(m)\rangle(-1)^{m\cdot u}\left(1+(-1)^{\xi\cdot u}\right).$$

Вычисляя, получим

$$\langle x|P_v|x\rangle = \frac{1}{2^{2n}}\sum_m\langle f(m)\,|\,f(m)\rangle\left|1+(-1)^{\xi\cdot v}\right|^2 = \begin{cases}\frac{1}{2^{n-1}}, & \text{если } v\perp\xi,\\ 0, & \text{если } v\not\perp\xi.\end{cases}$$

Следовательно, будут наблюдаться только вектора, ортогональные ξ, причем каждый из таких векторов будет возникать с вероятностью $\frac{1}{2^{n-1}}$.

Повторяя шаги 1–5, находим вектора, ортогональные ξ, пока не наберется $n-1$ линейно независимых.

Корректность алгоритма Саймона (2)

На шаге 5 мы замеряем первый регистр. Вероятность того, что при этом получится какой-то вектор v, равна, как мы знаем, $\langle x|P_v|x\rangle$, где P_v — самосопряженный оператор ортопроектирования на ось вектора v, а

$$|x\rangle := \frac{1}{2^n}\sum_{u=0}^{2^n-1}|u\rangle\sum_m|f(m)\rangle(-1)^{m\cdot u}\left(1+(-1)^{\xi\cdot u}\right).$$

Вычисляя, получим

$$\langle x|P_v|x\rangle = \frac{1}{2^{2n}}\sum_m\langle f(m)\,|\,f(m)\rangle\left|1+(-1)^{\xi\cdot v}\right|^2 = \begin{cases}\frac{1}{2^{n-1}}, & \text{если } v\perp\xi,\\ 0, & \text{если } v\not\perp\xi.\end{cases}$$

Следовательно, будут наблюдаться только вектора, ортогональные ξ, причем каждый из таких векторов будет возникать с вероятностью $\frac{1}{2^{n-1}}$.

Повторяя шаги 1–5, находим вектора, ортогональные ξ, пока не наберется $n-1$ линейно независимых. Легко подсчитать, что хватит $O(n)$ повторов.

Корректность алгоритма Саймона (2)

На шаге 5 мы замеряем первый регистр. Вероятность того, что при этом получится какой-то вектор v, равна, как мы знаем, $\langle x|P_v|x\rangle$, где P_v — самосопряженный оператор ортопроектирования на ось вектора v, а

$$|x\rangle := \frac{1}{2^n}\sum_{u=0}^{2^n-1}|u\rangle\sum_m|f(m)\rangle(-1)^{m\cdot u}\left(1+(-1)^{\xi\cdot u}\right).$$

Вычисляя, получим

$$\langle x|P_v|x\rangle = \frac{1}{2^{2n}}\sum_m\langle f(m)\,|\,f(m)\rangle\left|1+(-1)^{\xi\cdot v}\right|^2 = \begin{cases}\frac{1}{2^{n-1}}, & \text{если } v\perp\xi,\\ 0, & \text{если } v\not\perp\xi.\end{cases}$$

Следовательно, будут наблюдаться только вектора, ортогональные ξ, причем каждый из таких векторов будет возникать с вероятностью $\frac{1}{2^{n-1}}$.

Повторяя шаги 1–5, находим вектора, ортогональные ξ, пока не наберется $n-1$ линейно независимых. Легко подсчитать, что хватит $O(n)$ повторов. Так, первые $n-1$ векторов линейно независимы с вероятностью не меньше

$$\prod_{k=1}^{\infty}\left(1-\frac{1}{2^k}\right) = 0.288788\cdots > \frac{1}{4}.$$

Обсуждение

Новые моменты, проявившиеся в алгоритме Саймона: *вероятностный характер* и *классический постпроцессинг*.

Обсуждение

Новые моменты, проявившиеся в алгоритме Саймона: *вероятностный характер* и *классический постпроцессинг*.

В действительности, это типично: практически все квантовые алгоритмы вероятностные, и в большинстве нетривиальных квантовых алгоритмов используется нужен классический постпроцессинг, работающий за полиномиальное от размера задачи время.

Новые моменты, проявившиеся в алгоритме Саймона: *вероятностный характер* и *классический постпроцессинг*.

В действительности, это типично: практически все квантовые алгоритмы вероятностные, и в большинстве нетривиальных квантовых алгоритмов используется нужен классический постпроцессинг, работающий за полиномиальное от размера задачи время. В алгоритме Саймона применяется метод Гаусса, работающий, как известно, за $O(n^3)$.

Обсуждение

Новые моменты, проявившиеся в алгоритме Саймона: *вероятностный характер* и *классический постпроцессинг*.

В действительности, это типично: практически все квантовые алгоритмы вероятностные, и в большинстве нетривиальных квантовых алгоритмов используется нужен классический постпроцессинг, работающий за полиномиальное от размера задачи время. В алгоритме Саймона применяется метод Гаусса, работающий, как известно, за $O(n^3)$.

Задача Саймона — простейший представитель класса задач *о скрытой подгруппе в конечной группе*.

Новые моменты, проявившиеся в алгоритме Саймона: *вероятностный характер* и *классический постпроцессинг*.

В действительности, это типично: практически все квантовые алгоритмы вероятностные, и в большинстве нетривиальных квантовых алгоритмов используется нужен классический постпроцессинг, работающий за полиномиальное от размера задачи время. В алгоритме Саймона применяется метод Гаусса, работающий, как известно, за $O(n^3)$.

Задача Саймона — простейший представитель класса задач *о скрытой подгруппе в конечной группе*. В общем случае, известен квантовый алгоритм для случая абелевых групп.