

Квантовое преобразование Фурье

Квантовые вычисления–2023

5 декабря 2023 г.

1 Преобразование Фурье

2 Схема для QFT

3 Код для QFT

Fourier Transform

Разложение функции $f(x)$ в частотный спектр:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\xi x} dx.$$

(какая амплитуда $\hat{f}(\xi)$ должна быть у синусоиды с частотой ξ , чтобы в результате сложения получилась $f(x)$)

Discrete Fourier Transform

Входные данные

Если f задана таблицей $\{x_n\} = \{x_0, \dots, x_{N-1}\}$, можно вычислить таблицу значений $\{y_k\} = \{y_0, \dots, y_{N-1}\}$ образа \hat{f} .

DFT

$$y_k = \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} x_n \left(\cos\left(\frac{2\pi}{N}kn\right) - i \sin\left(\frac{2\pi}{N}kn\right) \right).$$

FFT

Для эффективного вычисления используется алгоритм *быстрого преобразования Фурье* (FFT, Fast Fourier Transform).

Квантовое преобразование Фурье (QFT)

- Обозначим $\omega_N = e^{2i\pi/N}$.
- Запишем $\{x_k\}$ и $\{y_k\}$ в амплитудах $|k\rangle$:
- Пусть $|X\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$
- $|Y\rangle = \sum_{k=0}^{N-1} y_k |k\rangle$.
- QFT преобразует:

$$|j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{jk} |k\rangle.$$

Преобразования для QFT

- Пусть $N = 2^n$. Тогда QFT работает так:

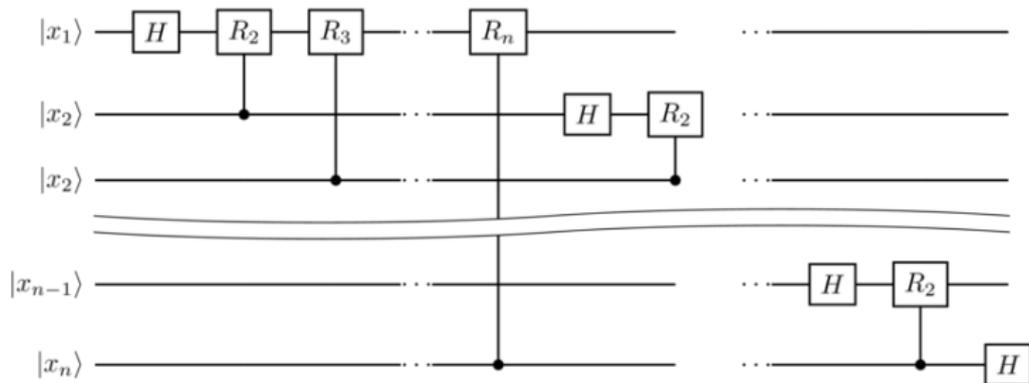
$$QFT_N|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega_N^{xy} |y\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i xy/2^n} |y\rangle.$$

- Пусть y — последовательность битов y_k . Тогда $y/2^n = \sum_{k=1}^n y_k/2^k$.
- Получаем

$$\begin{aligned} QFT_N|x\rangle &= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x \sum_{k=1}^n y_k/2^k} |y_1 y_2 \dots y_n\rangle = \\ &= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \prod_{k=1}^n e^{2\pi i x y_k/2^k} |y_1 y_2 \dots y_n\rangle = \\ &= \frac{1}{\sqrt{N}} \bigotimes_{k=1}^n (|0\rangle + e^{2\pi i x/2^k} |1\rangle). \end{aligned}$$

Схема QFT

Соответствующая схема будет выглядеть так:



Вспомогательные замечания

Нам нужен гейт для CR_j : $CR_j = CZ^{1/2^{j-1}}$.

В Cirq возведение в степень есть: `cirq.CZ ** (1/2**k)`.

Код для QFT

```
import cirq

def make_qft(qubits):
    qreg = list(qubits)
    while len(qreg) > 0:
        q_head = qreg.pop(0)
        yield cirq.H(q_head)
        for i, qubit in enumerate(qreg):
            yield (cirq.CZ ** (1/2**(i+1)))(qubit, q_head)
```

Проверка

Код для запуска

```
qubits = cirq.LineQubit.range(4)
qft = cirq.Circuit(make_qft(qubits))
print(qft)
```

Встроенная реализация

Разумеется, каждый раз писать вручную этот код не нужно. Эта функция уже реализована в `cirq` под названием `cirq.qft`.

Домашнее задание

Реализовать (вручную, не используя встроенную функцию) *обратное* преобразование Фурье