Алгоритм оценки фазы

Квантовые вычисления–2023

12 декабря 2023 г.

Outline

Формулировка

Дано

- унитарный оператор U (оракул)
- его собственный вектор $|\psi\rangle$) (состояние).

Поскольку U унитарный,

$$U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle.$$

Найти

n-битовую оценку θ.

Общая идея

Фаза точно записывается в n битов

Если $\theta = \frac{m}{2^n}$:

- записать фазу θ в n разных кубитов в базисе Фурье,
- применить обратное преобразование Фурье
- измерить регистр.

Общая идея

Фаза точно записывается в n битов

Если $\theta = \frac{m}{2^n}$:

- записать фазу θ в n разных кубитов в базисе Фурье,
- применить обратное преобразование Фурье
- измерить регистр.

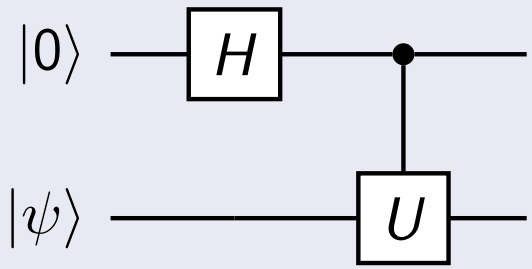
Нужно больше битов

Если фаза не целая:

- получим какую-то оценку (вероятность выше $4/\pi^2 \approx 40\%$).

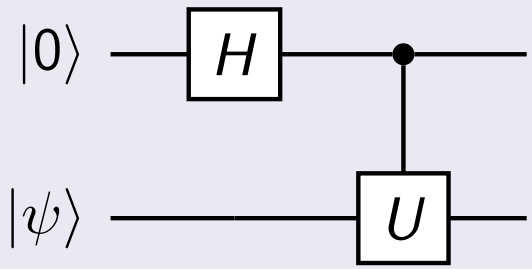
Фазовая отдача (phase kickback)

Схема



Фазовая отдача (phase kickback)

Схема



Результат

$$CU\Big(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\psi\rangle\Big) = \frac{1}{\sqrt{2}}\Big(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes e^{2\pi i\theta}|\psi\rangle\Big) =$$
$$= \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\theta}|1\rangle) \otimes |\psi\rangle.$$

Фаза записалась в управляющий кубит, как будто мы получили отдачу от управляемой операции.

Повышение точности

Разные j и разные степени CU^{2^j}:

$$|\psi'\rangle = \frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i\theta 2^{n-1}}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i\theta 2^{0}}|1\rangle\right) \otimes |\psi\rangle$$

$$= \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i\theta k}|k\rangle \otimes |\psi\rangle.$$

Повышение точности

Разные j и разные степени CU^{2^j}:

$$|\psi'\rangle = \frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i\theta 2^{n-1}}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i\theta 2^{0}}|1\rangle\right) \otimes |\psi\rangle$$
$$= \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i\theta k}|k\rangle \otimes |\psi\rangle.$$

Обратное QFT

$$|\psi''\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{-\frac{2\pi ik}{2^n}(x-2^n\theta)}|x\rangle \otimes |\psi\rangle.$$

Повышение точности

Обратное QFT

$$|\psi''\rangle = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} \sum_{k=0}^{2^n-1} e^{-\frac{2\pi i k}{2^n}(x-2^n\theta)} |x\rangle \otimes |\psi\rangle.$$

Измерение первого регистра

С большой вероятностью получим $x = 2^n\theta$. Для большей точности — больше кубитов в этом регистре.

Код

```
import cirq
import numpy as np

def make_phase_estimation(n, U, u_bit):
    qubits = cirq.LineQubit.range(n)
    phase_estimator = cirq.Circuit(cirq.H.on_each(*qubits))
    for i, bit in enumerate(qubits):
        phase_estimator.append(
            cirq.ControlledGate(U).on(bit, u_bit) **
            (2**(n - i - 1)))
    phase_estimator.append(cirq.qft(*qubits[::-1],
                                    without_reverse=True,
                                    inverse=True))
    phase_estimator.append(cirq.measure(*qubits,
                                        key='theta'))
    return phase_estimator
```

Пример 1

Если $2^n\theta$ — целое, то получим точное значение:

```
n1 = 3
u1 = cirq.NamedQubit('u')
pe1 = make_phase_estimation(n1, cirq.T, u1)
pe1.insert(0, cirq.X(u1))
print(pe1)

sim = cirq.Simulator()
res1 = sim.run(pe1, repetitions = 10)
print(res1.measurements['theta'])
theta1 = np.sum(2 ** np.arange(n1) *
                res1.measurements['theta'],
                axis=1) / 2**n1
print(theta1)
```

Результат

Здесь мы всегда получаем $\theta = 0.125$ потому что мы находим фазу для оператора $T = Z^{1/4}$ и собственного значения $|1\rangle$ (для того, чтобы начать с этого состояния мы и вставили X в схему):

$$T|1\rangle = e^{2\pi i/8}|1\rangle.$$

Пример 2

Если $2^n\theta$ нельзя представить точно в n битах, получим приближение:

```
n2 = 3
u2 = cirq.NamedQubit('u')
pe2 = make_phase_estimation(n2, cirq.Z**(1/3), u2)
pe2.insert(0, cirq.X(u2))
print(pe2)

res2 = sim.run(pe2, repetitions = 10)
print(res2.measurements['theta'])
theta2 = np.sum(2 ** np.arange(n2) *
                res2.measurements['theta'],
                axis=1) / 2**n2
print(theta2)
print(res2.histogram(key = 'theta'))
```

Пример 3

Чем больше кубитов, тем точнее результат:

Четыре кубита

```
n3 = 4
u3 = cirq.NamedQubit('u')
pe3 = make_phase_estimation(n3, cirq.Z**(1/3), u3)
pe3.insert(0, cirq.X(u3))
print(pe3)

res3 = sim.run(pe3, repetitions = 1000)
print(res3.histogram(key = 'theta'))
```

Результат

- Функция `histogram` собирает биты: $4 = 100_2$, $12 = 1100_2$, $20 = 10100_2$.

Результат

- Функция `histogram` собирает биты: $4 = 100_2$, $12 = 1100_2$, $20 = 10100_2$.
- Приближения: $001, 0011, 00101, \ldots$

Результат

- Функция `histogram` собирает биты: $4 = 100_2$, $12 = 1100_2$, $20 = 10100_2$.
- Приближения: $001, 0011, 00101, \ldots$
- Соответственно: $\theta \approx 0.125, 0.1875, 0.15625, \ldots$

Результат

- Функция `histogram` собирает биты: $4 = 100_2$, $12 = 1100_2$, $20 = 10100_2$.
- Приближения: $001, 0011, 00101, \ldots$
- Соответственно: $\theta \approx 0.125, 0.1875, 0.15625, \ldots$
- Точное значение $\theta = 1/6 = 0.1(6) = 0.1666\ldots = 0.0(01)_2 = 0.00101010101\ldots_2$.

Результат

- Функция `histogram` собирает биты: $4 = 100_2$, $12 = 1100_2$, $20 = 10100_2$.
- Приближения: $001, 0011, 00101, \ldots$
- Соответственно: $\theta \approx 0.125, 0.1875, 0.15625, \ldots$
- Точное значение $\theta = 1/6 = 0.1(6) = 0.1666\ldots = 0.0(01)_2 = 0.00101010101\ldots_2$.

Результат

- Функция `histogram` собирает биты: $4 = 100_2$, $12 = 1100_2$, $20 = 10100_2$.
- Приближения: $001, 0011, 00101, \ldots$
- Соответственно: $\theta \approx 0.125, 0.1875, 0.15625, \ldots$
- Точное значение $\theta = 1/6 = 0.1(6) = 0.1666\ldots = 0.0(01)_2 = 0.00101010101\ldots_2$.

Домашнее задание

Функция получает:

- оракул U
- состояние ψ
- количество битов n
- количество повторений N_{reps}

Возвращает оценку фазы θ