Quantum algorithms: Exercices 3

Benoît Vermersch (benoit.vermersch@lpmmc.cnrs.fr) -September 27, 2023

1 Implementation of the quantum Fourier transform

Ref: Nielsen and Chuang. The quantum Fourier transform realizes the transformation

$$U|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle, \qquad (1)$$

with j, k = 0, ..., N - 1. Our goal is to implement this transformation for $N = 2^n$, using a circuit of n qubits.

- 1. Write any integer $0 \le j \le N-1$ in terms of a binary representation $j = j_1 \dots j_n$. In our quantum circuit, j_l will represent the state of qubit l.
- 2. We use the notation $0.j_l \dots j_n = j_l/2 + \dots j_n/2^{n-l+1}$. Show that

$$U|j\rangle = \frac{1}{2^{n/2}} (|0\rangle + e^{2i\pi 0.j_n} |1\rangle) (|0\rangle + e^{2i\pi 0.j_{n-1}j_n} |1\rangle) \dots (|0\rangle + e^{2i\pi 0.j_1 \dots j_n} |1\rangle)$$
(2)

3. We show the circuit of the quantum Fourier transform.



The single qubit gate R_k is defined as

$$R_k = \begin{bmatrix} 1 & 0\\ 0 & e^{2i\pi/2^k} \end{bmatrix}.$$
(3)

How is a basis $|j\rangle$ transformed after the first controlled R_2 rotation? After the first R_n rotation?

4. Write down the state at the end of the circuit. Conclusion.

2 Factorizing 21 with Shor's algorithm

We take N = 21.

- 1. Classical part Assume we randomly pick a = 2. Show that the function $f(x) = a^x \mod(N)$ is 6 periodic.
- 2. Find two non-trivial divisors of N.
- 3. The quantum subroutine The quantum subroutine of Shor's algorithm consists in finding the period r = 6 of f(x). How many qubits do we need to implement this algorithm?
- 4. Write the state of the system after modular exponentiation.
- 5. Write the state after inverse quantum Fourier transform and the probability P(y) to observe the bitstring y after measuring the first q qubits.
- 6. Plot the function P(y) and extract the three most likely measured bitstrings.
- 7. The continued fraction algorithm is a classical algorithm that gives the closest fraction p/r from the measured y/Q rational, with a maximum r_{max} value for r. In Python, this is implemented as fractions.Fraction(float).limit_denominator(rmax).

Give the attributed value for each most likely bitstring r. Comment.

8. Repeat the same exercice, aiming at factorizing 35.