AQAlg: Advanced Quantum Algorithms **2023 - 2024**

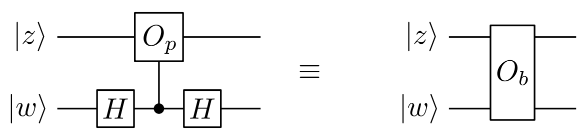
Exercises 1: QFT, phase estimation and Shor's algorithm

Lecturer: Simon Apers (apers@irif.fr)

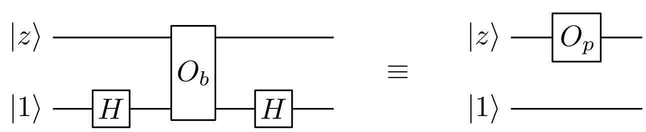
Exercise 1 (Oracles)**.** For accessing a function $f : \{0,1\}^n \to \{0,1\}$ with a quantum circuit, we use a *bit oracle* O_b or a *phase oracle* O_p. For $z \in \{0,1\}^n$ and $w \in \{0,1\}$, these are defined as follows:

We can show that both oracles are equivalent in a sense.

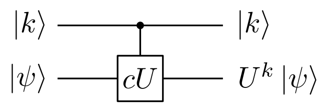
- Show that the phase oracle can simulate the bit oracle:



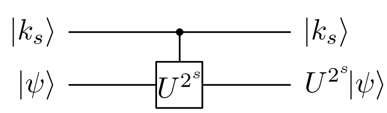
- Show that the bit oracle can simulate the phase oracle:



Exercise 2 (Controlled unitary)**.** Recall the controlled unitary gate:

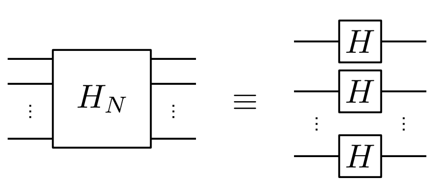


where $k = k_1 \ldots k_n$ is an n-bit integer. Expand this gate into more elementary gates of the form



for $k_s \in \{0,1\}$ and $s \in \{0,1,\ldots,n-1\}$.

Exercise 3 (Hadamard transform)**.** A variation on the quantum Fourier transform is the Hadamard transform H_N for $N = 2^n$. It is defined by $H_N = H^{\otimes n}$, which corresponds to the circuit

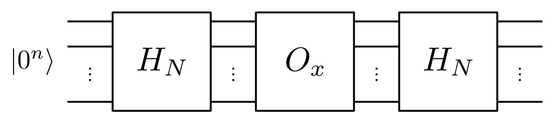


- What is $H_N |0^n\rangle$ equal to?
- What is $H_N |k\rangle = H_N |k_1 \dots k_n\rangle$ equal to? Use the inner product $x \cdot k = \sum_\ell x_\ell k_\ell$.[1]

Exercise 4 (Bernstein-Vazirani algorithm)**.** Let $N = 2^n$. Consider a function $f : \{0,1\}^n \to \{0,1\}$ that is determined by some hidden string $a \in \{0,1\}^n$ in the following way:

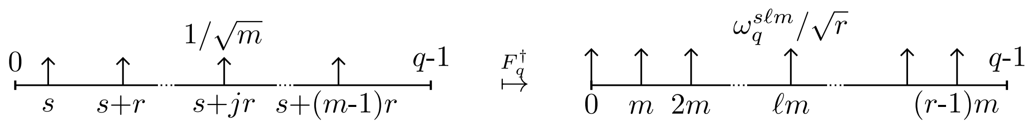
$$f(x) = (x \cdot a) \pmod 2.$$

We can access the function through the phase oracle $O_x |x\rangle = (-1)^{f(x)} |x\rangle$. What is the output of the following circuit?



Exercise 5 (Fourier analysis)**.** Consider natural numbers q, m, r such that $q = mr$. Prove the following critical identity in Shor's algorithm for period finding:

$$\frac{1}{\sqrt{m}} \sum_{j=0}^{m-1} |s + jr\rangle \quad \overset{F_q^\dagger}{\mapsto} \quad \frac{1}{\sqrt{r}} \sum_{\ell=0}^{r-1} \omega_q^{s\ell m} |\ell m\rangle \,,$$



Exercise 6 (Factoring reduction (optional))**.** Here we walk through Shor's reduction from factoring to period finding. Recall that we are given an n-bit integer N such that $2^{n-1} \le N < 2^n$, and we wish to find a (nontrivial) factor of N. Argue that, without loss of generality, we can assume that N is odd and not a prime power.[2]

Now pick $x \in \{2, \dots, N-1\}$ uniformly at random. If $\gcd(N, x) > 1$ then we can run Euclid's algorithm to find a factor. Hence, assume that N and x are coprime, and consider the series

$$x^0 = 1 \pmod N, \qquad x \pmod N, \qquad x^2 \pmod N, \qquad \dots$$

Since N and x are coprime, there does not exist s such that $x^s = 0 \pmod N$. Show that this implies that the series must have a period $r \le N$ for which $x^r = 1 \pmod N$. This r is called the *multiplicative order* of x modulo N, and it is precisely this factor that is calculated using quantum period finding.

One can show (not in this exercise!) that, with probability at least $1/2$ over the choice of x, the period r will be even and both $x^{r/2} + 1$ and $x^{r/2} - 1$ are not multiples of N. Use $x^r = 1 \pmod N$ to show that this implies that both $x^{r/2} + 1$ and $x^{r/2} - 1$ must share a (nontrivial) factor with N. Once we computed r, we can then find these factors by computing $\gcd(x^{r/2} \pm 1, N)$.

[1] Hint: use that $H |k_\ell\rangle = \frac{1}{\sqrt{2}} \sum_{x_\ell=0}^{1} (-1)^{x_\ell k_\ell} |x_\ell\rangle$.

[2] Hint: if $N = p^k$ for some prime $p \ge 2$ then necessarily $k \le n$.