Freie Universität Berlin
Tutorials on Quantum Information Theory
Winter term 2020/21

Problem Sheet 10
Aspects of quantum algorithms and circuits

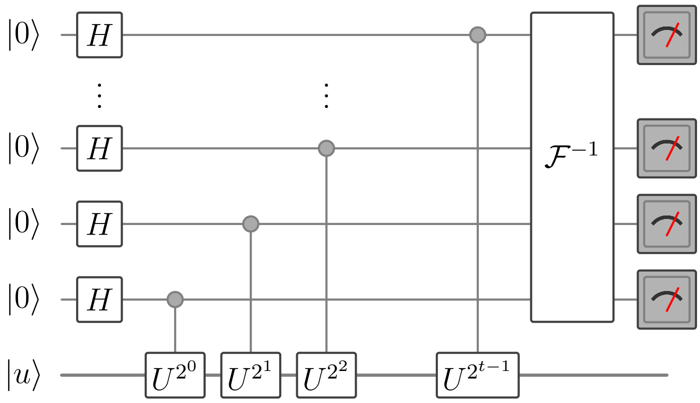
J. Eisert, J. Haferkamp, J. C. Magdalena De La Fuente

---

1. **Phase estimation.** Perhaps at the heart of the majority of modern quantum algorithms lies the *phase estimation algorithm*. The problem of phase estimation is the following: Given a unitary operator U and one of its eigenvectors $|u\rangle$ with eigenvalue $e^{2\pi i\phi}$, the phase estimation problem is to output the phase ϕ.

   a) On the last sheet the definition and the circuit of the quantum Fourier transform was given. Show that the quantum Fourier tranform is a unitary operator and draw the circuit implementing the inverse of the Fourier transform.

   The phase estimation algorithm is implemented via the following quantum circuit:



   The circuit constists of H, the Hadamard gate, controlled-U^{2^k}-gates, that apply the unitary operator U for 2^k times if the control qubit is $|1\rangle$, the inverse of the quantum Fourier transform $\mathcal{F}^{-1}$ and a measurement in the computational basis at the very end. At the beginning, the first register comprising t qubits is initialised as $|0\rangle^{\otimes t}$ and the second register is prepared in the state $|u\rangle$. For simplicity we assume that ϕ can be written with exactly t bits, i.e. $\phi = \sum_{k=1}^{t} \phi_k 2^{-k}$ with $\phi_k \in \{0, 1\}$.

   b) Show that the algorithm works.

   Solution: Before applying the inverse Fourier transform, the first register will be

$$\frac{1}{\sqrt{2^t}}\left(|0\rangle + e^{2\pi i 2^{t-1}\phi}|1\rangle\right)\left(|0\rangle + e^{2\pi i 2^{t-2}\phi}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i t^0\phi}|1\rangle\right) \tag{1}$$

$$= \frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^{t-1}} e^{2\pi i\phi k}|k\rangle\,. \tag{2}$$

   Thus, after applying the inverse Fourier transform the measurement will report the fractional binary expression $\{\phi_k\}$ for ϕ.

   c) How many calls of the unitary operator are required in the algorithms?

   Solution: We need $1+2+4+\ldots+2^t = \sum_{i=k}^{t} 2^k = \frac{1-2^t}{1-2} \in \mathcal{O}(2^t)$ calls of the unitary.

   d) What is the computational complexity of a classical solution to the phase estimation problem?

Solution: Classically it would suffice to just apply the unitary U only once to u and read of the phases from the resulting eigenvalue. Nevertheless, depending on the unitary U this might be very costly.

The benefit of quantum phase estimation comes from being a subroutine in another quantum algorithm such as Shor's algorithm and being able to read out the phase of a quantum state deterministically.

e) Sketch why phase estimation constitutes the core of Shor's algorithm.

Solution: By the black magic of number theory, one can establish the equivalence of prime factoring and order finding. For positive integers x and N , $x < N$, with no common factors, the order of $x \mod N$ is defined to be the least positive integer, r, such that $x^r = 1 \mod N$. Calculating r given x and N is the order-finding problem.

Order finding can be formulated as a phase estimation problem in the following way: Assume you have access to unitary implementing

$$U_{x,N} |y\rangle = |xy \mod N\rangle . \tag{3}$$

(This can be done using a classical logical implementation of the corresponding circuit and rendering it revesible by standard techniques.)

Now $U_{x,N}$ has eigenstates

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left\{\frac{-2\pi \mathrm{i} sk}{r}\right\} \left|x^k \mod N\right\rangle \tag{4}$$

with eigenvalues $\mathrm{e}^{2\pi \mathrm{i} s/r}$. They fulfil

$$\sum_{s=0}^{r-1} |u_s\rangle = |1\rangle . \tag{5}$$

So we can easily prepare their superposition $|1\rangle$ and use this as the input vector on the second register of the standard phase estimation algorithm.

Now, measuring the output of the will yield one of the phases $\{s/r\}_{s=0}^{r-1}$ with equal probability. From which we can infer r.

2. **Control gates.**

a) Show that the control-Z gate is invariant under swapping the two inputs with each other and the two outputs.

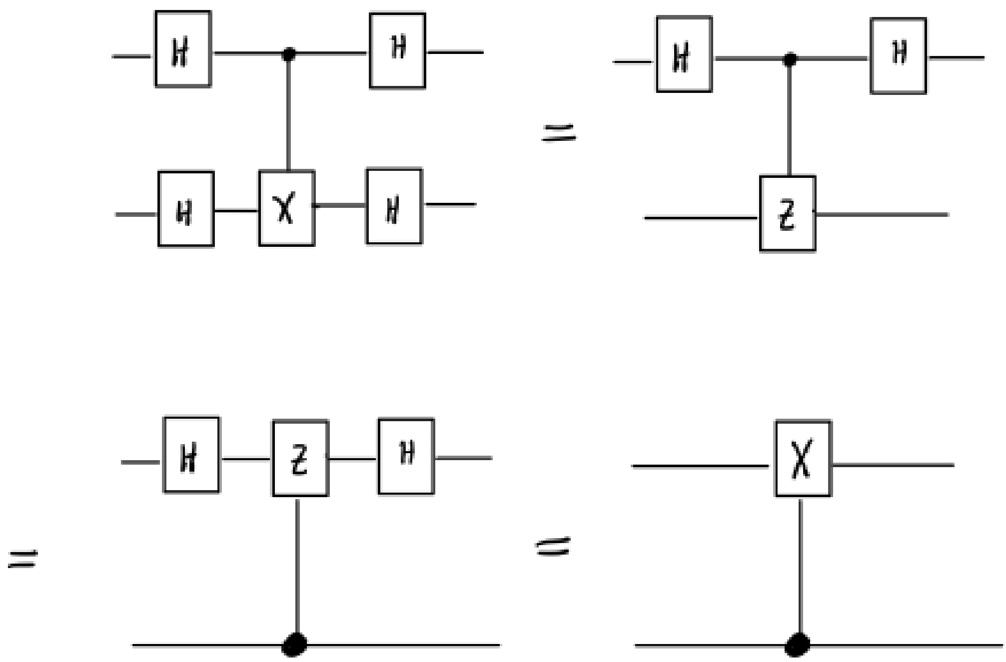
Solution: One way to show this is to calculate the matrix representation of both cZ-gates in the computational basis. We denote a unitary acting on the i-th register controlled by the logical qubit in the j-th register by $c_i U_j$. For $c_1 Z_2$ we have

$$
\begin{aligned}
c_1 Z_2 &= |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes Z && (6)\\
&= \begin{pmatrix} 1 & \\ & \end{pmatrix} \otimes \begin{pmatrix} 1 & \\ & 1 \end{pmatrix} + \begin{pmatrix} & \\ & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} && (7)\\
&= \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix} && (8)\\
&= \begin{pmatrix} 1 & & & \\ & & & \\ & & 1 & \\ & & & \end{pmatrix} + \begin{pmatrix} & & & \\ & 1 & & \\ & & & \\ & & & -1 \end{pmatrix} && (9)\\
&= \mathbb{1} \otimes |0\rangle\langle 0| + Z \otimes |1\rangle\langle 1| = c_2 Z_1, && (10)
\end{aligned}
$$

where it might be more straight-forward to start the second part of the calculation at the end and meet in the middle.

b) The roles of the two inputs to the cNOT gate can be exchanged by applying the gate in another basis than the computational basis. Find a local unitary that applied to all inputs and outputs and turns a cNOT gate controlled by the first register into one controlled by the second register.

Solution: We now that $c_1 Z_2 = c_2 Z_1$. Furthermore, we can rotate from the Z to X eigenbasis and vice versa with the Hadamard gate H. Thus, we have $HXH = Z$. Thus, the idea is to rotate $c_1 X_1$ to Z basis use the result of (a) and rotate back. Indeed, we have



3. **Probabilistic algorithm for Deutsch-Josza.**

The Deutsch-Josza algorithm can determine whether a function $f : \{0,1\}^n \to \{0,1\}$ is balanced or constant by invoking the function (or more precisely a quantum implementation of the function) only a single time. In contrast, a deterministic classical algorithm needs to invoke the function exponentially $\mathcal{O}(2^n)$ often (at least in a worst-case scenario).

Assume instead that the goal is not to distinghuish these two cases with certainty, but only with a probability $p > 1/2$. How does the best classical algorithm for this problem perform?

Solution: A probabilistic classical algorithm with high success probability can be easily found. You simply query the function f m times and if all outputs agree, you write "constant" and else you write "balanced". In the latter case the error probability is 0 as f cannot be constant. In the former case, the error probability is supressed as 2^{-m+1}: First you draw a string x_0 and memorize $f(x_0)$. If f is balanced, the probability is $1/2$ to draw next an x_1 such that $f(x_1) = f(x_0)$ and so on...