Introduction to Quantum Computation

These lecture notes are based on the *Introduction to Quantum Computation* course given at EPFL for the academic year 2024/2025 by Prof. Olivier Lévêque and Prof. Rüdiger Urbanke. The content of the course was originally created by Prof. Nicolas Macris and is based on his lecture notes.

Department of Computer Science - École Polytechnique Fédérale de Lausanne

Author: Arthur Aimone

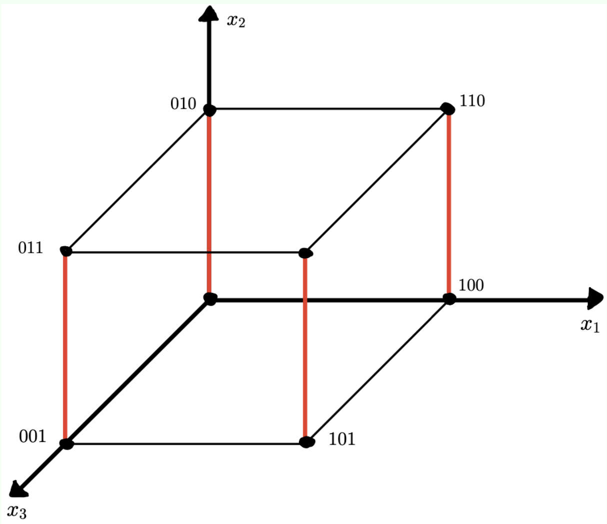
2024/2025

Chapter 4

Simon's Algorithm

Let $f : \{0,1\}^n \longrightarrow \{0,1\}^n$ be a function such that $f(x) = f(y)$ if and only if either $x = y$ or $x \oplus s = y$ for some $s \in \{0,1\}^n$. Note that s is unknown.
The **aim** is to discover the value of $s \neq 0$ by asking as few questions as possible to the oracle f. Classically, we will see that this requires $\mathcal{O}(2^n)$ calls, whereas Simon's quantum algorithm finds the vector a with probability $p \geq 1-\epsilon$ in a runtime of $O(n) \cdot |\log(\epsilon)|$ and with a similar number of calls to the oracle.

Example 4.0.1 $(n = 3)$**.** *Consider* $f(x \oplus s) = f(x)$ *for all* $x \in \{0,1\}^3$*. We consider the vector* s *to be* $s = (0,1,0)$*.*



4.1 Classical Algorithm

The classical algorithm is constructed as follows: draw randomly pairs of points in $\{0,1\}^n$ (with replacement): $\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(q)}, y^{(q)}\right)$. If one such pair (say j), $f\left(x^{(j)}\right) = f\left(y^{(j)}\right)$, compute $s = x^{(j)} \ominus y^{(j)}$ $(= x^{(j)} \oplus y^{(j)})$ and declare success. On the contrary, if $f\left(x^{(j)}\right) \neq f\left(y^{(j)}\right)$ for all $1 \leq j \leq q$, then declare that $s = 0^n$.

Proposition 4.1.1. *It holds that* $\mathbb{P}(success) \leq \frac{q}{2^n - 1}$.

Note that in order to ensure $\mathbb{P}(\text{success}) \geq 1-\epsilon$ we require $q \geq (2^n-1)(1-\epsilon)$ draws.

Proof. We remark that:

$$\mathbb{P}(\text{success}) = \mathbb{P}\left(\exists\, 1 \leq j \leq q : f\left(x^{(j)}\right) = f\left(y^{(j)}\right)\right) \leq \sum_{j=1}^{q} \mathbb{P}\left(f\left(x^{(j)}\right) = f\left(y^{(j)}\right)\right) \leq \frac{q}{2^n-1}. \quad (4.1)$$

We have used the fact that for a given x there is a unique corresponding y, hence:

$$\mathbb{P}\left(f\left(x^{(j)}\right) = f\left(y^{(j)}\right)\right) = \frac{1}{2^n-1}. \quad (4.2)$$

□

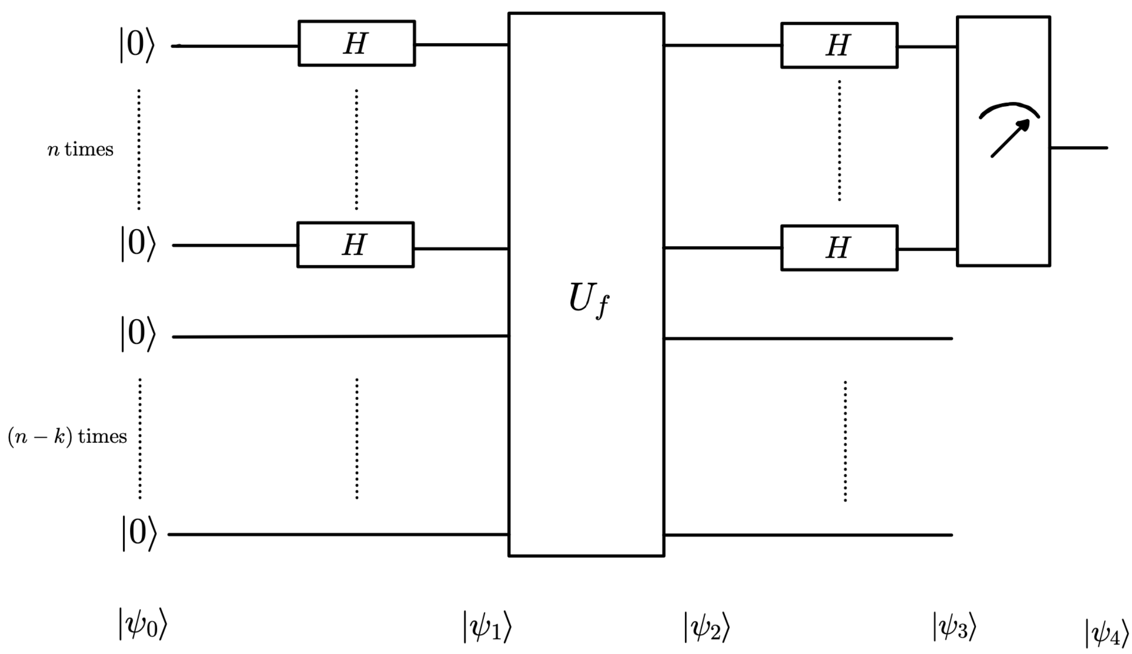
A slightly better classical algorithm can be related to the Birthday Problem. This simply corresponds to the random sampling in a set on N elements. The result is that in general the order of trials until we observe two identical elements is $\sqrt{N}$. As a result $\mathcal{O}\left(2^{\frac{N}{2}}\right)$ draws only are needed, however this is still exponential in N.

4.2 Simon's Quantum Algorithm

We now present the quantum alternative to Simon's algorithm, where we will make use of the Hadamard gate H, the quantum oracle U_f and finally making a measurement. We start with an initial state: (Note that $H^{\otimes n} \otimes \mathbb{I}_n$ is the same as $H^{\otimes n} \otimes \mathbb{I}^{\otimes n}$.)

$$|\psi_0\rangle = \left(\bigotimes^{n \text{ times}} |0\rangle\right) \otimes \left(\bigotimes^{n \text{ times}} |0\rangle\right) := \underbrace{|0\rangle \otimes \ldots \otimes |0\rangle}_{n \text{ times}} \otimes \underbrace{|0\rangle \otimes \ldots \otimes |0\rangle}_{n \text{ times}}. \quad (4.3)$$

(In the figure below, we consider the case where $k = 0$.)



<u>**Stage 1**</u>: Note that contrary to the Deutsch-Josza's algorithm, the n ancilla qubits are left untouched before the passage through the oracle U_f.

$$|\psi_1\rangle = (H^{\otimes n} \otimes \mathbb{I}_n)\,|\psi_0\rangle = H^{\otimes n}\,|0...0\rangle \otimes |0...0\rangle = \frac{1}{2^{\frac{n}{2}}} \sum_{x_1,\dots,x_n \in \{0,1\}} |x_1...x_n\rangle \otimes |0...0\rangle = \frac{1}{2^{\frac{n}{2}}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |0...0\rangle\,.$$

<u>**Stage 2**</u>: To find $|\psi_2\rangle$, state $|\psi_1\rangle$ has to go through the quantum oracle. The oracle U_f is defined as:

$$U_f\left(|x\rangle \otimes |y\rangle\right) = |x\rangle \otimes |y \oplus f(x)\rangle\,, \tag{4.4}$$

where we note that $f(x)$ is modulo 2. Hence:

$$U_f U_f\,|x\rangle \otimes |y\rangle = U_f\,|x\rangle \otimes |y \oplus f(x)\rangle = |x\rangle \otimes |y \oplus f(x) \oplus f(x)\rangle = |x\rangle \otimes |y\rangle\,. \tag{4.5}$$

However remark that both y and $f(x)$ are n-dimensional. So:

$$|\psi_2\rangle = U_f\,|\psi_1\rangle = \frac{1}{2^{\frac{n}{2}}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |f(x)\rangle\,. \tag{4.6}$$

<u>**Stage 3**</u>: Following what was done for the Deutsch-Josza's algorithm we have:

$$H^{\otimes n}\,|x\rangle = \frac{1}{2^{\frac{n}{2}}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y}\,|y\rangle\,. \tag{4.7}$$

So, for $|\psi_3\rangle$ we obtain:

$$|\psi_3\rangle = \left(H^{\otimes n} \otimes \mathbb{I}\right)|\psi_2\rangle = \frac{1}{2^n} \sum_{x,y \in \{0,1\}^n} (-1)^{x \cdot y}\,|y\rangle \otimes |f(x)\rangle\,. \tag{4.8}$$

At this stage, we can consider two cases. Either $s = 0^n$, in which case f is a bijection and there is nothing to simplify in this expression. Or $s \neq 0^n$ and we can rewrite $|\psi_3\rangle$ using a set of the representatives for the range of f. For this, label $f(\{0,1\}^n) = \{f_1, \dots, f_K\}$ where $K = 2^{n-1}$. Furthermore, each f_j has two preimages, call them v_j and $v_j \oplus s$. Then,

$$\begin{aligned}
|\psi_3\rangle &= \sum_{x \in \{0,1\}^n} \frac{1}{2^n} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y}\,|y\rangle \otimes |f(x)\rangle \\
&= \frac{1}{2^n} \sum_{y \in \{0,1\}^n} \sum_{j=1}^{K} \left((-1)^{v_j \cdot y} + (-1)^{(v_j \oplus s) \cdot y}\right)|y\rangle \otimes |f_j\rangle \\
&= \frac{1}{2^n} \sum_{y \in \{0,1\}^n} \sum_{j=1}^{K} (-1)^{v_j \cdot y}\left(1 + (-1)^{s \cdot y}\right)|y\rangle \otimes |f_j\rangle\,.
\end{aligned}$$

Here for the second line we grouped each pair of preimages $x = v_j, v_j \oplus s$ to the same $f(x) = f_j$ together, and for the third line we used that $(-1)^{v_j \cdot y}(-1)^{s \cdot y} = (-1)^{(v_j \oplus s) \cdot y}$. Finally, observe that if $s \cdot y = 1$ then $1 + (-1)^{s \cdot y} = 0$ and all such terms vanish. Thus we get

$$|\psi_3\rangle = \frac{1}{2^{n-1}} \sum_{y:\ y \cdot s = 0} \sum_{j=1}^{K} (-1)^{v_j \cdot y}\,|y\rangle \otimes |f_j\rangle\,. \tag{4.9}$$

Stage 4: This last stage is dedicated to the measurement of the first n qubits. Here, the first n qubits are entangled with the last n qubits in state $|\psi_3\rangle$. Hence the partial measurement of the first n qubits is more difficult to describe than in the case of Deutsch-Josza's algorithm. We make an aside to describe general measurements.

In general, a measurement is described in quantum mechanics by a complete collection of orthogonal projectors $\{P_j : 1 \leq j \leq d\}$. In our context, the projectors admit the following properties:

- Applying the projection twice is the same as applying it once: $P_j^2 := P_j \cdot P_j = P_j$
- We consider orthogonal projections, i.e. for all x, y: $\langle P_j x, y - P_j y\rangle = 0$
- The projections are unitary. Recall that for all operators p, $\langle x, p^\dagger y\rangle = \langle px, y\rangle$. Then:

$$\langle P_j x, y - P_j y\rangle = \langle x, P_j^\dagger y - P_j^\dagger P_j y\rangle = \langle x, P_j y - P_j P_j y\rangle = \langle x, (P_j - P_j P_j) y\rangle = 0. \tag{4.10}$$

As a result and by definition it holds that $\langle P_j x, y - P_j y\rangle = 0 = \langle x - P_j x, P_j y\rangle$. Hence:

$$\langle x, P_j^\dagger y\rangle = \langle P_j x, y\rangle = \langle P_j x, P_j y\rangle = \langle x, P_j y\rangle \Longrightarrow P_j = P_j^\dagger, \quad \forall 1 \leq j \leq d. \tag{4.11}$$

- The set of projectors is complete: $\sum_{j=1}^d P_j = \mathbb{I}$.

Example 4.2.1. *$P_j = |\phi_j\rangle\langle\phi_j|$, where $\{|\phi_j\rangle, 1 \leq j \leq d\}$, is an orthonormal basis of the Hilbert space $\mathcal{H}$.*

Anyways, back to our measurement! Now, if the system is in state $|\psi\rangle$ before the measurement, the outcome state is:

$$|\psi'\rangle = \frac{P_j|\psi\rangle}{\|P_j|\psi\rangle\|} \quad \text{with probability} \quad \|P_j|\psi\rangle\|^2 = \langle\psi|P_j^\dagger P_j|\psi\rangle = \langle\psi|P_j|\psi\rangle.$$

In our case, the measurement of the first n qubits is described by the following complete collection of projectors:

$$\{P_y = |y\rangle\langle y| \otimes \mathbb{I}_{n-k}, y \in \{0,1\}^n\}. \tag{4.12}$$

For a given $y_0 \in \{0,1\}^n$, let us compute the outcome probability $\langle\psi_3|P_{y_0}|\psi_3\rangle$ of the state $\frac{P_{y_0}|\psi_3\rangle}{\|P_{y_0}|\psi_3\rangle\|} = |y_0\rangle \otimes$ (some state we do not care about). We then obtain:

$$\begin{aligned}
\langle\psi_3|P_{y_0}|\psi_3\rangle &= \Big(\sum_{y:\ y\cdot s=0} \frac{1}{2^{n-1}} \sum_{j=1}^K (-1)^{v_j\cdot y} \langle y| \otimes \langle f_j|\Big)\big(|y_0\rangle\langle y_0| \otimes \mathbb{I}_n\big)\Big(\sum_{y':\ y'\cdot s=0} \frac{1}{2^{n-1}} \sum_{j'=1}^K (-1)^{v_{j'}\cdot y'} |y'\rangle \otimes |f_{j'}\rangle\Big) \\
&= \sum_{y,y':\ y\cdot s=y'\cdot s=0} \frac{1}{2^{2(n-1)}} \sum_{j,j'=1}^K (-1)^{v_j\cdot y + v_{j'}\cdot y'} \langle y|y_0\rangle\langle y_0|y'\rangle\langle f_j|f_{j'}\rangle \\
&= \sum_{y,y':\ y\cdot s=y'\cdot s=0} \frac{1}{2^{2(n-1)}} \sum_{j,j'=1}^K (-1)^{v_j\cdot y + v_{j'}\cdot y'} \delta_{yy_0}\delta_{y_0y'}\delta_{jj'}.
\end{aligned}$$

The above quadruple sum simplifies to two different results depending on the status of y_0:

- If $y_0 \cdot s \neq 0$ then it is equal to 0 .

• If $y_0 \cdot s = 0$ then we obtain:

$$\frac{1}{2^{2(n-1)}}\sum_{j=1}^{K}(-1)^{v_j\cdot y_0+v_j\cdot y_0} = \frac{1}{2^{2(n-1)}}\sum_{j=1}^{K}(-1)^{2v_j\cdot y_0} = \frac{1}{2^{2(n-1)}}\sum_{j=1}^{K}1 = \frac{1}{2^{n-1}}. \tag{4.13}$$

Hence, in the case of $y_0 \cdot s = 0$, the outcome probabilities are uniform over valid equations in s.

In conclusion, Simon's algorithm is then the following:

• Run $(n-1)$ times the above circuit. This will output $y^{(1)},...,y^{(n-1)}$ uniformly and independently distributed, conditioned on $y^{(j)} \cdot s = 0$.

• If $y^{(1)},...,y^{(n-1)}$ are linearly independent, then these form a basis of $H = \{y : y\cdot = 0\}$ which is of dimension $(n-1)$. From this basis, we compute the basis of the dual space H via a classical algorithm (Gauss elimination - runtime $\mathcal{O}(n^3)$). In this case, we declare success and return the basis (which is just s).

• If $y^{(1)},...,y^{(n-1)}$ are not linearly independent the declare failure and restart the algorithm.

Proposition 4.2.2. *It holds that* $\mathbb{P}(\textit{success}) \geq \frac{1}{4}$.

Proof. We have to consider the following probabilities:

$$\mathbb{P}\left(y^{(1)} \neq 0\right) = 1 - \frac{1}{2^{n-1}}. \tag{4.14}$$

$$\mathbb{P}\left(y^{(2)} \notin \text{span}\left(y^{(1)}\right)\Big|y^{(1)} \neq 0\right) = \mathbb{P}\left(y^{(2)} \notin \left\{0, y^{(1)}\right\}\Big|y^{(1)} \neq 0\right) = 1 - \frac{2}{2^{n-1}} = 1 - \frac{1}{2^{n-2}}. \tag{4.15}$$

$$\mathbb{P}\left(y^{(3)} \notin \text{span}\left(y^{(1)}, y^{(2)}\right)\Big|y^{(1)}, y^{(2)} \text{ lin. indep.}\right) = 1 - \frac{4}{2^{n-1}} = 1 - \frac{1}{2^{n-3}}. \tag{4.16}$$

This process continues until the step $(n-1)$ which is given by:

$$\mathbb{P}\left(y^{(n-k)} \notin \text{span}\left(y^{(1)}, ..., y^{(n-2)}\right)\Big|y^{(1)}, ..., y^{(n-2)} \text{ lin. indep.}\right) = 1 - \frac{2^{n-2}}{2^{n-1}} = 1 - \frac{1}{2} = \frac{1}{2}.$$

Now, $\mathbb{P}(\text{success}) = \mathbb{P}\left(y^{(1)}, ..., y^{(n-1)} \text{ are lin. indep.}\right)$ and this can be represented by:

$$\mathbb{P}(\text{success}) = \prod_{i=1}^{n-1}\left(1 - \frac{2^{i-1}}{2^{n-1}}\right) = \prod_{i=0}^{n-2}\left(1 - \frac{1}{2^{n-1-i}}\right) = \prod_{i=1}^{n-1}\left(1 - \frac{1}{2^i}\right) = \exp\left(\sum_{i=1}^{n-1}\log\left(1 - \frac{1}{2^i}\right)\right).$$

One can plot the function $\log(1-x)$ and find a linear function $g(x)$ such that $\log(1-x) \geq g(x)$ on the interval $0 \leq x \leq \frac{1}{2}$. Given that the function $\log(1-x)$ is concave, with a simple analysis it is not difficult to show that the required function is $g(x) = -(2\log(2))x$. Therefore:

$$\mathbb{P}(\text{success}) \geq \exp\left(-2\log(2)\sum_{i=1}^{n-1}\frac{1}{2^i}\right) \geq \exp\left(-2\log(2)\right) = 2^{-2} = \frac{1}{4}, \tag{4.17}$$

where we have used the fact that $\sum_{i=1}^{n-1}\frac{1}{2^i} \leq 1$.

□

Of course, a success probability of only $\frac{1}{4}$ is not satisfactory; we would like a success probability

$\mathbb{P} \geq 1-\epsilon$. Let us therefore repeat independently the whole algorithm T times:

$$\mathbb{P}(\text{failure after } T \text{ attempts}) = \mathbb{P}(\text{failure})^T \leq \left(\frac{3}{4}\right)^T \leq \epsilon \quad \text{if} \quad T \geq \frac{|\ln(\epsilon)|}{|\ln(3/4)|}. \tag{4.18}$$

In the end, we obtain a success probability $\mathbb{P} \geq 1-\epsilon$ after $\mathcal{O}(n \cdot |\ln(\epsilon)|)$ calls to the quantum oracle U_f (and a polynomial runtime dominated by the $\mathcal{O}(n^3)$ computation of the dual basis). This is to be compared to the $\Omega(2^n)$ calls to the oracle f of any classical algorithm.