9.1 Portes, circuits et algorithmes quantiques

9.1.1 Portes quantiques

Les portes quantiques sont les éléments de base de l'ordinateur quantique.

Définition 9.1. *Une porte est une transformation unitaire qui agit sur au plus trois qbits.*

Bien que les portes quantiques soient agencées en circuits, il faut noter que dans un ordinateur quantique "réel" les portes seraient plutôt comme des instructions assembleur pour un ordinateur classique. Elles sont successivement appliquées à des registres de qbits, sans que le circuit qu'elles constituent ne soit réellement gravé.

9.1.2 Produit tensoriel de portes

Le produit tensoriel de deux portes quantiques est simplement l'application en parallèle de deux portes quantiques.

$$(G_1 \otimes G_2)|\Phi_1\rangle|\Phi_2\rangle = (G_1|\Phi_1\rangle)(G_2|\Phi_2\rangle)$$

On peut étendre une porte à plus de trois entrées en faisant son produit tensoriel avec l'identité.

9.1.3 Circuits et algorithmes

Un circuit est une composition de portes en produit tensoriel avec l'identité. La taille d'un circuit est le nombre de portes qu'il faut pour le réaliser, et la complexité de la fonction F est la taille minimale d'un circuit capable de la calculer. Cette complexité ne dépend pas du choix de portes effectué.

Les portes quantiques opèrent sur un nombre réduit d'entrées, mais un algorithme quantique peut nécessiter beaucoup plus d'entrées. On essaye donc comme en informatique classique, de déterminer avec quels ensembles minimaux de portes on peut construire tout algorithme quantique.

Cette mesure d'approximation permet de définir une famille universelle de portes quantiques.

Définition 9.2. *Une famille de portes est dite universelle si tout opérateur unitaire de n-qbit peut être approché à une précision arbitrairement grande par un circuit composé de portes de cette famille.*

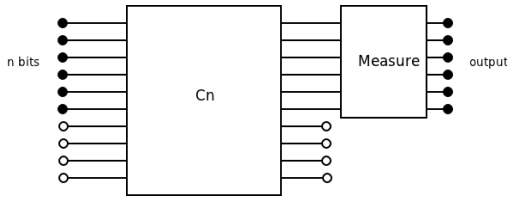
Du fait que l'on doit par exemple être capable de simuler la porte C-NOT, il existe des contraintes sur de telles familles universelles. En particulier, comme C-NOT est intriquante toute famille universelle doit contenir au moins une porte intriquante.

On peut trouver un résultat plus fort :

Théorème 9.3. *Une famille contenant une porte intriquante à 2 qbits, et toutes les portes à 1 qbit, est universelle.*

9.1.4 Algorithm in the circuit model

Définition 9.4 (Circuit model). *A quantum algorithm is a deterministic algorithm that, given n, outputs a description of a quantum circuit C_n for all inputs of size n.*



The time complexity of the quantum algorithm is $\mathcal{T}(n)$ when

1. *the deterministic algorithm has time complexity $\mathcal{T}(n)$,*
2. *and C_n uses at most $\mathcal{T}(n)$ gates.*

9.2 Deutsch-Jozsa problem

Oracle input : $f : \{0,1\}^n \to \{0,1\}$ (n-bit string) with promise :
- either f is constant
- either f is balanced : $\Pr[f(X) = 0] = \Pr[f(X) = 1] = \frac{1}{2}$

Output : decide which case.

Lemme 9.5. *Any deterministic algorithm for this problem must make at least $1 + 2^{n-1}$ queries to f.*

Théorème 9.6. *There is a quantum algorithm for this problem with only 1 query to f and no error.*

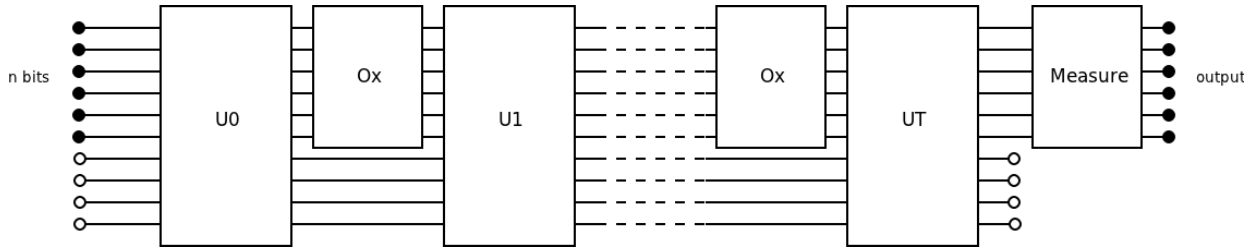
Définition 9.7 (Query model). *In spite of the fact $|i\rangle \mapsto |f(i)\rangle$ is not unitary,*

$$|i\rangle|0\rangle \mapsto |i\rangle|f(i)\rangle$$

can be simulated by a unitary operator :

$$O_f : |i\rangle|b\rangle \mapsto |i\rangle|b \oplus f(i)\rangle$$

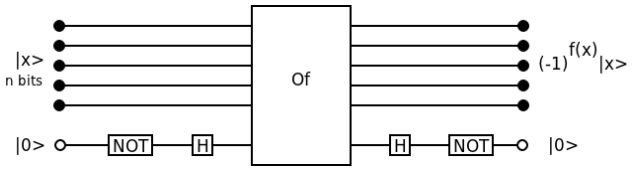
Indeed, $O_f^2 = Id$: this way, we can do queries in the quantum framework.



First, to ease the demonstration, we will need the following lemma :

Lemme 9.8. *One can simulate* $S_f : |x\rangle \mapsto (-1)^{f(x)}|x\rangle$ *with 1 query to* O_f.

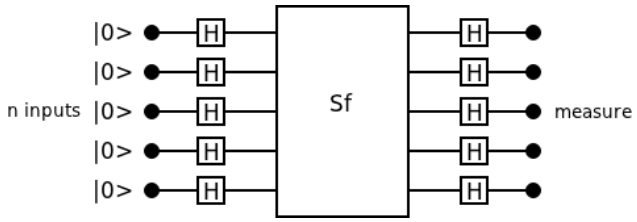
Preuve:



$$\frac{1}{\sqrt{2}}(|x\rangle|0\rangle - |x\rangle|1\rangle) \xrightarrow[\boxed{O_f}]{} \frac{1}{\sqrt{2}}(|x\rangle|f(x)\rangle - |x\rangle|1-f(x)\rangle) = (-1)^{f(x)}|x\rangle\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$
$$\xrightarrow[\boxed{\text{H}}\ \boxed{\text{NOT}}]{} (-1)^{f(x)}|x\rangle|0\rangle$$

□

Then, define the following algorithm :



Lemme 9.9. *If* f *is constant, then the output is* always 0^n. *If* f *is balanced, then the output is* never 0^n.

Preuve: The result requires the use of the Quantum Fourier Transform :

Lemme 9.10.

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

Preuve: $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ so if $b \in \{0,1\}$ we have

$$H|b\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b|1\rangle)$$

This allows us to write the following :

$$H^{\otimes n}|x_1, x_2, \ldots, x_n\rangle = \bigotimes_{i=1}^{n} H|x_i\rangle = \frac{1}{\sqrt{2}^n}\bigotimes_{i=1}^{n}(|0\rangle + (-1)^{x_i}|1\rangle)$$

$$= \frac{1}{\sqrt{2}^n}\bigotimes_{i=1}^{n}((-1)^{x_i\cdot 0}|0\rangle + (-1)^{x_i\cdot 1}|1\rangle)$$

$$= \tfrac{1}{\sqrt{2}^n}\textstyle\sum_{y\in\{0,1\}}\alpha_y|y\rangle \text{ if } \alpha_y = \prod_{i=1}^{n}(-1)^{x_i\cdot y_i} = (-1)^{x\cdot y}$$

□

Thanks to this result, we are now able to express the value of our vector at different steps of our computation.

– After the $H^{\otimes n}$:

$$\frac{1}{\sqrt{2}^n}\sum_{x\in\{0,1\}^n}|x\rangle$$

– After S_f :

$$\frac{1}{\sqrt{2}^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle$$

– After the second $H^{\otimes n}$:

$$\frac{1}{\sqrt{2}^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}\frac{1}{\sqrt{2}^n}\sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$$

$$= \frac{1}{2^n}\sum_{y\in\{0,1\}^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)+x\cdot y}|y\rangle$$

Let us call this result $|\psi\rangle$. How to decide which case is f ?

– f is constant
say for instance $f(x) = 0$ then $|\psi\rangle = \frac{1}{2^n}\sum_{y\in\{0,1\}^n}\sum_{x\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle$ which entails that

$$\langle \mathbf{0}|\psi\rangle = \frac{1}{2^n}\sum_{x\in\{0,1\}}1 = 1$$

which gives us $|\psi\rangle = |\mathbf{0}\rangle$. Same argument if $f(x) = 1$ yields $|\psi\rangle = -|\mathbf{0}\rangle$

– f is balanced then

$$\langle \mathbf{0}|\psi\rangle = \frac{1}{2^n}\sum_{x\in\{0,1\}}(-1)^{f(x)} = \frac{1}{2^n}(\#f^{-1}(0) - \#f^{-1}(1)) = 0$$

because f is balanced. Thus $|\psi\rangle$ is orthogonal to $|\mathbf{0}\rangle$.

It is this way very simple to know in which case is f. It is enough to make one measure : the vector we get has to be colinear or orthogonal to $|\mathbf{0}\rangle$ and each corresponds to a possible case for f : being constant or balanced.

□

9.3 Bernstein-Vazirani

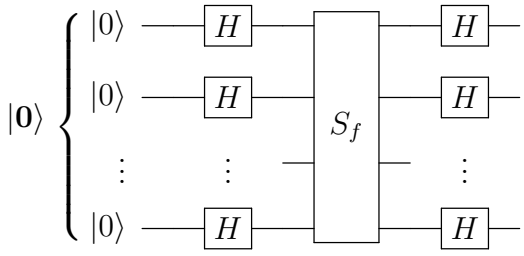
Let $f : \{0,1\}^n \to \{0,1\}$, such that there exists a such that $f(x) = x \cdot a$.

Problem : Find a.

Any randomized algorithm for this problem needs to query f at least $\Omega(n)$ times. Moreover there is a deterministic algorithm with n queries. 2^n possible values for a, we need at least n queries.

However, the quantum complexity is very different is only 1.

To show this, we are going to use a setting very similar to the one previously used to study a balanced function f. Indeed let us consider the same algorithm, and make the same measurement. The only thing that will change will be the expression of f, and so the meaning of the gate S_f.



The steps of the calculus become :

– After the $H^{\otimes n}$:

$$\frac{1}{\sqrt{2}^n} \sum_{x \in \{0,1\}^n} |x\rangle$$

– After S_f :

$$\frac{1}{\sqrt{2}^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle = \frac{1}{\sqrt{2}^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot a} |x\rangle$$

– After the second $H^{\otimes n}$:

$$\frac{1}{\sqrt{2}^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot a} \frac{1}{\sqrt{2}^n} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle$$

$$= \frac{1}{2^n} \sum_{y \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot a + x \cdot y} |y\rangle$$

Let us call $|\psi\rangle$ this last vector. We can write

$$|psi\rangle = \frac{1}{2^n} \sum_{y \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot (a \oplus y)} |y\rangle$$

It is a fact now, that what we measure at the output of our system is actually the vector a used within the definition of the function f. Indeed, we have

$$\langle a|\psi\rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot a + x \cdot a} = 1$$

Since $|\psi\rangle$ is unitary and has one component equal to 1, then all its other components have to be null, which exactly means that the vector we've just computed is a.

9.4 Simon problem

Let $f : \{0,1\}^n \to \{0,1\}^n$.

We assume that f is **s-periodic** and **injective up to the period**, i.e.

$\exists s \in \{0,1\}^n \, \forall x, y. f(x) = f(y) \Leftrightarrow x = y \vee x = y \oplus s$.

Problem : Find s.

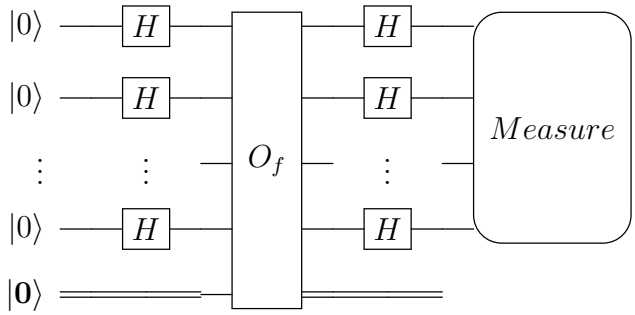
Notes :

If $s = \mathbf{0}$, then the function is bijective.

If $s \neq \mathbf{0}$, then the solution is easy to check, by verifying if $f(\mathbf{0}) = f(s)$.

Any randomized algorithm for this problem requires $\Omega(2^{n/2})$ queries. This complexity can be achieved by a randomized algorithm based on birthday paradox.

9.4.1 The quantum algorithm for Simon's problem



Initially we will assume $s \neq \mathbf{0}$, and later see what happens when we run the algorithm for $s = \mathbf{0}$.

Initially : $|0^n\rangle|0^n\rangle$.

After $H^{\otimes n}$: $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle|\mathbf{0}\rangle$

Query : $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle|f(x)\rangle$

We measure now the value of $|f(x)\rangle$, call the result v

The state becomes :

$\frac{1}{\sqrt{2}} \sum_{x \in \{0,1\}^n, f(x)=v} |x\rangle|v\rangle = \frac{1}{\sqrt{2}} (|x\rangle + |x \oplus s\rangle) |v\rangle$, where $f(x) = v$, and x is chosen uniformly at random.

Now, we use Hadamard gates $H^{\otimes n}$ again :

$\frac{1}{\sqrt{2^{n-1}}} \sum_y \langle (-1)^{x \cdot y} + (-1)^{(x \oplus s) \cdot y}, y\rangle = \frac{1}{\sqrt{2^{n-1}}} \left((-1)^{x+y} \left(1 + (-1)^{s \cdot y}\right)\right)$.

Here, the amplitude of y is $\begin{cases} 0 & s \cdot y = 1 \\ \frac{1}{\sqrt{2^{n-1}}} & s \cdot y = 0 \end{cases}$. So as a result we get uniform y such that $s \cdot y = 0$.

If $s = \mathbf{0}$, then the same reasoning gives us any y, uniformly at random.

We will now repeat the experiment many times, getting a sequence y^i, whose elements satisfy $s \cdot y^i = 0$. This will give us a system of linear equations over $\mathbb{Z}_2$. If the system is rank n, then we can be sure that $s = \mathbf{0}$. Otherwise, with large probability, there will be a unique nonzero solution, and we output it. The soundness of this approach follows from

Lemme 9.11. $\Pr\left[\mathrm{rk}\left\{x^1, x^2, \ldots, x^{d+k}\right\} < d\right] \leqslant \frac{1}{2^k}$, *where x^i are vectors from $\mathbb{Z}_2^d$.*

Corollaire 9.12. – *If* $s = \mathbf{0}$*, then* $\Pr\left[\mathrm{rk}\left\{y^1, \ldots, y^{d+k}\right\} < d\right] \leqslant \frac{1}{2^k}$
– *If* $s \neq \mathbf{0}$*, then* $\Pr_{sy=0}\left[\mathrm{rk}\left\{y^1, \ldots, y^{d+k}\right\} < d-1\right] \leqslant \frac{1}{2^{k+1}}$

Preuve: Fix a hyperplane H of dimension $d-1$. Call the vector space G.
$\Pr\left[y^1, y^2, \ldots, y^{d+k} \in H\right] =$
[independence]
$(\Pr[y \in H])^{d+k} = (|H| / |G|)^{d+k} \leqslant \frac{1}{2^{d+k}}$.
We now prove the bound by summing over all hyperplanes H.
$\Pr\left[\mathrm{rk}\left\{x^1, x^2, \ldots, x^{d+k}\right\} < d\right] = \Pr\left[\exists \text{hyperplane } H. y^1, \ldots, y^{d+k} \in H\right] \leqslant$
[union bound]
$\sum_H \Pr\left[y^1, \ldots, y^{d+k} \in H\right] = \#\text{hyperplanes} \cdot \frac{1}{2^{d+k}} =$
[Every hyperplane H corresponds to a vector v by duality, i.e. it can be represented as $v^\perp = \{w : v \cdot w = 0\}$]
$= |G| \cdot \frac{1}{2^{d+k}} = \frac{1}{2^k}$.

□

9.4.2 Generalization of Simon's problem

We now consider a function $f : \{0,1\}^n \to \{0,1\}^n$ such that there is a subspace H such that $\forall x, y. f(x) = f(y) \Leftrightarrow x \oplus y \in H$.

In original problem, $H = \{\mathbf{0}, s\}$.

We can generalize the previous algorithm to find H (equivalently, its generator set) ; this is left as an exercise.

9.5 Factorization Algorithm (Shor)

9.5.1 Period over $\mathbb{Z}$

We now consider a different generalization, by going from $\mathbb{Z}_2$ to $\mathbb{Z}$.

We are given a function $f : \mathbb{Z} \to S$ and an integer N which is r-periodic and injective up to r, with $r \leq N$, i.e.

$\forall x, y. f(x) = f(y) \Leftrightarrow x \equiv y \pmod r$

To avoid annoying and not used special case we assume $r \geqslant 2$, i.e. f is not injective.

Problem : Find r.

Théorème 9.13. *There is a quantum algorithm that finds r with $\Theta(\log N)$ queries to f, and time $O\left(\log^3 N\right)$. Moreover this algorithm fails with constant probability, and never errs.*

9.5.2 Order finding

Input : Integer $N \geqslant 1$ and $a \in \{1, 2, \ldots, N-1\}$ coprime to N.

Output : Smallest $r \geqslant 1$ such that $a^r \equiv 1 \bmod N$

We define $f : \mathbb{Z} \to \{0, 1, \ldots, N-1\}$ by $f(x) = a^x \bmod N$.

It is easily observed that first r values of f are pairwise distinct (if $a^i \equiv a^j$ for $0 \leqslant i < j < r$ then $a^{j-i} \equiv 1$, which contradicts definition of r)

Therefore, f is r-periodic and injective up to the period.
What more, f can be computed fast, in logarithmic time (exponentiation by squaring).
Therefore one can solve order finding with quantum polynomial time.

9.6 Factorization

Input : N - integer, not a prime
Output : d such that $d|N$ and $1 < d < N$

Théorème 9.14. *There is a polynomial time randomized reduction from factorization to order finding.*

Algorithm :

1. Deal with the case when N is a prime power.
   If $N = p^k$, then $2 \leqslant k \leqslant \log_2 N$, so we can check all possible k, and find p using binary search. This takes $O\left(\log^c N\right)$ time.
   From now on, we can assume N is not a prime power.
2. Select $a \in \{2, \ldots, N-1\}$ uniformly at random. If $\gcd(a, N) \neq 1$ then we are lucky and found a factor of N, and we can stop.
3. Compute the order r of a modulo N, using the procedure.
4. If r is odd, or $a^{r/2} \equiv -1 \pmod N$, then report failure.
   Otherwise, we compute $\gcd\left(N, a^{r/2} - 1\right)$, and report it as a prime factor.

Lemme 9.15. *The probability of failure in step 4 is at most 1/2.*

Therefore one can factorize within quantum polynomial time.