

QFT, QPE, Shor

Léo Durand-Kollner

Mars 2021

1 Introduction

Il existe principalement 4 classes d'algorithmes quantiques : les algorithmes *variationnels* (e.g. *VQE*), les algorithmes de *simulation quantique*, les algorithmes de *recherche* (e.g. *Grover*), et enfin les algorithmes basés sur la *transformée de Fourier quantique* (QFT). Dans ces notes de cours, on s'attarde sur cette dernière catégorie. On présente la QFT, et deux de ses plus grandes applications : *Quantum Phase Estimation* (QPE) et *l'algorithme de Shor*.

2 QFT inverse

On se donne le problème suivant : étant donné $x \in \{0, 1\}$ et l'état $|\psi\rangle = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} \omega^{xy}$, avec $\omega = \exp(\frac{2i\pi}{2^n})$, peut-on remonter à x ? Dans le cas $n = 1$ (1 seul qbit), $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^x|1\rangle)$. En appliquant une porte de Hadamard \mathbf{H} , on retrouve donc x . Pour $n = 2$, en posant $x = x_1x_0$ en binaire,

$$\begin{aligned} |\psi\rangle &= \frac{1}{2}(|00\rangle + i^x|01\rangle + (-1)^x|10\rangle + (-i)^x|11\rangle) \\ &= \frac{1}{2}(|0\rangle \otimes (|0\rangle + i^x|1\rangle) + |1\rangle \otimes (-1)^x(|0\rangle + i^x|1\rangle)) \\ &= \frac{1}{2}(|0\rangle + (-1)^x|1\rangle) \otimes (|0\rangle + i^x|1\rangle) \\ &= \frac{1}{2}(|0\rangle + (-1)^{x_0}|1\rangle) \otimes (|0\rangle + (-1)^{x_1}i^{x_0}|1\rangle) \end{aligned}$$

On peut généraliser à n qbits, pour $x = x_{n-1}...x_0$:

$$|\psi\rangle = \frac{1}{2^{n/2}} \bigotimes_{k=1}^n (|0\rangle + (\prod_{l=0}^{k-1} \exp(\frac{2i\pi x_l}{2^{k-l}}))|1\rangle)$$

où le produit est pris de gauche à droite (on commence à gauche par les indices faibles).

L'état obtenu n'est pas intriqué, on peut donc appliquer des portes quantiques à un qbit sans affecter ses voisins. En appliquant \mathbf{H} au dernier qbit (de poids fort), on retrouve x_0 . Ensuite, pour le deuxième qbit de poids fort $\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}i^{x_0}|1\rangle)$, il faudrait arriver à se débarrasser de i^{x_0} . En effet, dans ce cas, on se retrouve juste avec l'état $\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}|1\rangle)$ et il suffit d'appliquer à nouveau une Hadamard \mathbf{H} .

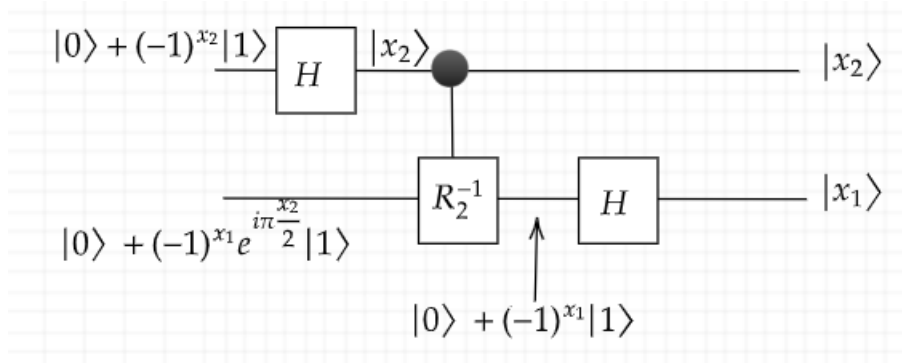


FIGURE 1 – QFT inverse

Pour y arriver, on utilise une phase de $-\frac{\pi}{2}$ contrôlée par le qbit de poids fort. Cette porte peut s'écrire $e^{-i\frac{\pi}{4}}\mathbf{R}_{\mathbf{Z}}(-\frac{\pi}{2})$. En effet :

- Si $x_0 = 0$, l'avant dernier qbit reste inchangé et se trouve dans l'état $\frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}|1\rangle)$
- Si $x_0 = 1$, on applique la phase $-\frac{\pi}{2}$ devant le $|1\rangle$, et donc on se retrouve dans l'état $\frac{1}{\sqrt{2}}(|0\rangle + e^{-i\frac{\pi}{2}}(-1)^{x_1}i|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{x_1}|1\rangle)$

Ceci se généralise pour le k -ème qbit. On applique des phases contrôlées par les qbits précédents (*i.e.* déjà réduits aux bits de poids fort de x). Les phases à appliquer sont de la forme $e^{-i\frac{\pi}{2^{l+1}}}\mathbf{R}_{\mathbf{Z}}(-\frac{\pi}{2^l})$. Le circuit qu'on vient de voir est la transformée de Fourier quantique inverse. La figure 1 résume les portes à appliquer.

3 QFT

On vient de voir que la QFT inverse était construite à partir d'opérateurs unitaires : la QFT inverse est donc également une transformation unitaire :

$$|\psi\rangle = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} \omega^{xy} \xrightarrow{QFTinv} |x_0\rangle|x_1\rangle\ldots|x_{n-1}\rangle$$

On définit alors la QFT comme étant la réciproque de cette transformation :

$$|x_0\rangle|x_1\rangle\ldots|x_{n-1}\rangle \xrightarrow{QFT} \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} \omega^{xy} |y\rangle$$

On rajoute souvent un opérateur de permutation à la fin, afin d'avoir le bon ordre des qbits.

Pour avoir l'expression de cet opérateur il suffit donc d'inverser tous les opérateurs qui composent la QFT inverse. Par exemple, pour $n = 4$, en notant $\mathbf{U}_{\mathbf{FT}}$ la QFT,

$$\begin{aligned} \mathbf{U}_{\mathbf{FT}} &= (\mathbf{H}_0\mathbf{C}_0^1(e^{-i\frac{\pi}{4}}\mathbf{R}_{\mathbf{Z}}(-\frac{\pi}{2}))\mathbf{C}_0^2(e^{-i\frac{\pi}{8}}\mathbf{R}_{\mathbf{Z}}(-\frac{\pi}{4}))\mathbf{C}_0^3(e^{-i\frac{\pi}{16}}\mathbf{R}_{\mathbf{Z}}(-\frac{\pi}{8})) \\ &\quad \mathbf{H}_1\mathbf{C}_1^2(e^{-i\frac{\pi}{4}}\mathbf{R}_{\mathbf{Z}}(-\frac{\pi}{2}))\mathbf{C}_1^3(e^{-i\frac{\pi}{8}}\mathbf{R}_{\mathbf{Z}}(-\frac{\pi}{4}))\mathbf{H}_2\mathbf{C}_2^3(e^{-i\frac{\pi}{4}}\mathbf{R}_{\mathbf{Z}}(-\frac{\pi}{2}))\mathbf{H}_3)^\dagger \\ &= \mathbf{H}_3\mathbf{C}_2^3(e^{i\frac{\pi}{4}}\mathbf{R}_{\mathbf{Z}}(\frac{\pi}{2}))\mathbf{H}_2\mathbf{C}_1^3(e^{i\frac{\pi}{8}}\mathbf{R}_{\mathbf{Z}}(\frac{\pi}{4}))\mathbf{C}_1^2(e^{i\frac{\pi}{4}}\mathbf{R}_{\mathbf{Z}}(\frac{\pi}{2})) \\ &\quad \mathbf{H}_1\mathbf{C}_0^3(e^{i\frac{\pi}{16}}\mathbf{R}_{\mathbf{Z}}(\frac{\pi}{8}))\mathbf{C}_0^2(e^{i\frac{\pi}{8}}\mathbf{R}_{\mathbf{Z}}(\frac{\pi}{4}))\mathbf{C}_0^1(e^{i\frac{\pi}{4}}\mathbf{R}_{\mathbf{Z}}(\frac{\pi}{2}))\mathbf{H}_0 \end{aligned}$$

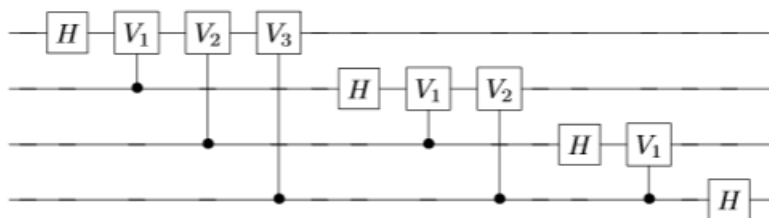


FIGURE 2 – QFT

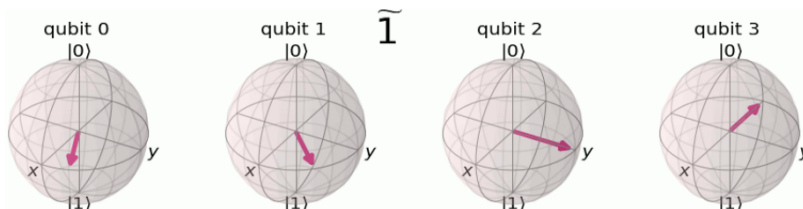


FIGURE 3 – Effet de la QFT sur la base canonique (ici $|1\rangle$)

Comme l'opérateur de phase contrôlée est symétrique en ses qubits target et contrôle, on peut poser $\mathbf{V}_{i,j} = \mathbf{C}_i^j(e^{i\frac{\pi}{2^{|i-j|+1}}}\mathbf{R}_Z(\frac{\pi}{2^{|i-j|}}))$. En faisant commuter les opérateurs qui agissent sur des qubits distincts, on peut donc réécrire :

$$\mathbf{U}_{\text{FT}} = \mathbf{H}_3 \mathbf{V}_{2,3} \mathbf{H}_2 \mathbf{V}_{1,3} \mathbf{V}_{1,2} \mathbf{H}_1 \mathbf{V}_{0,3} \mathbf{V}_{0,2} \mathbf{V}_{0,1} \mathbf{H}_0$$

Intuitivement, que fait cette transformée ? Et bien, elle transforme les états de la base canonique dans la base de Fourier. Par exemple, avec $n = 4$:

$$|x\rangle \xrightarrow{QFT} \frac{1}{4}(|0\rangle + \omega^{8x}|1\rangle) \otimes (|0\rangle + \omega^{4x}|1\rangle) \otimes (|0\rangle + \omega^{2x}|1\rangle) \otimes (|0\rangle + \omega^x|1\rangle)$$

On remarque qu'en faisant varier x de 0 à 15, le bit de poids fort tourne deux fois plus vite que le précédent, qui lui même tourne deux fois plus vite que le précédent, et ainsi de suite. On encode l'information dans des rotations sur la sphère de Bloch. Par exemple, la figure suivante montre l'état $|1\rangle$ après passage dans la QFT : le qbit de poids fort est tourné d'un demi tour ($\omega^8 = e^{i\pi}$), le qbit 2 est tourné d'un quart de tour, etc...

4 QPE

4.1 Présentation du problème

On peut maintenant se servir de la QFT pour résoudre des problèmes. Une des applications les plus courantes est la QPE : *Quantum Phase Estimation*.

Posons le problème : soit \mathbf{U} un opérateur unitaire de vecteur propre $|u\rangle$ associé à la valeur propre λ . Comme \mathbf{U} est unitaire, ses valeurs propres sont nécessairement sur le cercle unité, donc $\lambda = e^{2i\pi\varphi}$ avec $\varphi \in [0, 1[$. Le but du jeu est de trouver φ .

Pour cela, on suppose disposer de l'attirail suivant :

- On sait implémenter l'état $|u\rangle$.
- On sait implémenter les portes U^{2^k} pour n'importe quel k .

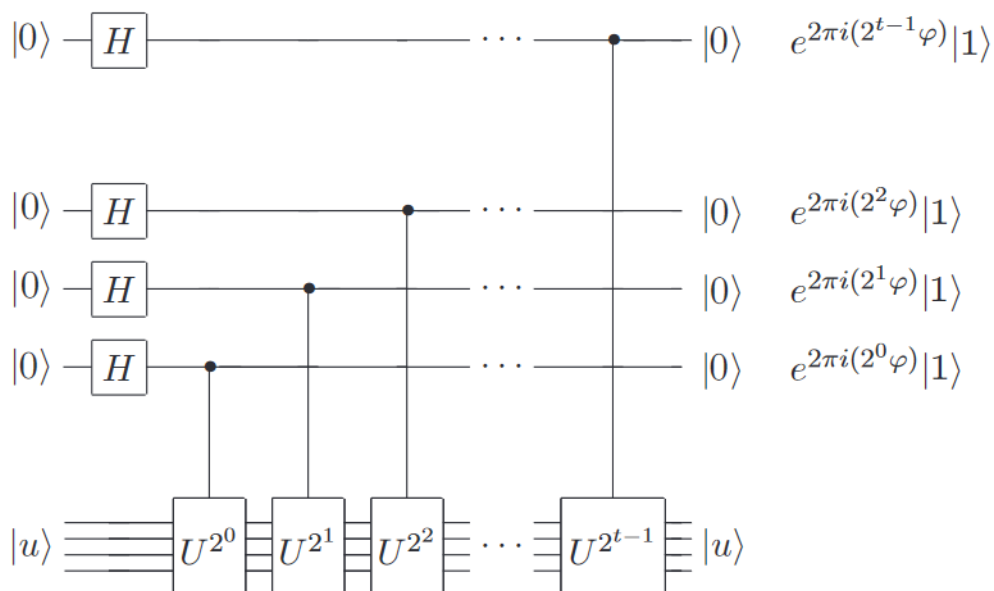


FIGURE 4 – Première partie du circuit

4.2 Cas idéal

On se fixe un entier t , qui correspond à la précision souhaitée pour φ , et on suppose dans un premier temps que φ s'écrit de manière exacte en binaire sur t bits : $\varphi = \frac{1}{2}\varphi_1 + \dots + \frac{1}{2^t}\varphi_t$. On note d le nombre de qbits sur lesquels agit \mathbf{U} . On dispose de deux registres : le premier est initialisé à $|0\rangle_t$, et le deuxième est initialisé à $|u\rangle_d$. On commence par appliquer des Hadamards au premier registre. Ensuite, on applique successivement des portes contrôlées \mathbf{U}^{2^k} (cf figure 1).

Comme $|u\rangle$ est un vecteur propre, à chaque application de \mathbf{U} , on fait sortir un facteur $e^{2i\varphi\pi}$. Par conséquent l'état final du premier registre s'écrit :

$$|\psi\rangle = \frac{1}{2^{t/2}}(|0\rangle + e^{2i\pi 2^{t-1}\varphi}|1\rangle) \otimes \dots \otimes (|0\rangle + e^{2i\pi 2^0\varphi}|1\rangle)$$

En utilisant l'écriture binaire de φ :

$$\begin{aligned} |\psi\rangle &= \frac{1}{2^{t/2}}(|0\rangle + e^{\frac{2i\pi}{2^t}2^{t-1}2^t\varphi}|1\rangle) \otimes \dots \otimes (|0\rangle + e^{\frac{2i\pi}{2^t}2^02^t\varphi}|1\rangle) \\ &= \frac{1}{2^{t/2}} \bigotimes_{k=0}^{t-1} (|0\rangle + \omega^{2^k(2^t\varphi)}|1\rangle) \end{aligned}$$

avec $\omega = e^{\frac{2i\pi}{2^t}}$, ce qui peut encore se réécrire, comme on l'a vu précédemment, comme :

$$|\psi\rangle = \frac{1}{2^{t/2}} \sum_{y=0}^{2^t-1} \omega^{(2^t\varphi)y} |y\rangle$$

Par conséquent, si on applique ensuite la QFT inverse à notre premier registre, on tombe sur $2^t\varphi = 2^{t-1}\varphi_1 + \dots + 2^0\varphi_t$. On mesure directement l'écriture binaire de φ .

4.3 Cas général

Mais que se passe-t-il si φ ne s'écrit pas de manière exacte en binaire ? Est-on certain de trouver une bonne approximation de φ ? Il se trouve que QPE fonctionne toujours bien, mais avec une certaine probabilité. On a deux paramètres à prendre en compte :

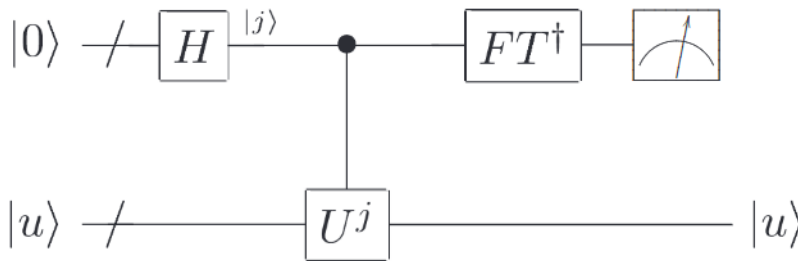


FIGURE 5 – Circuit complet

- ϵ , notre tolérance à l'erreur
- m , la précision souhaitée

Si on choisit la taille t du premier registre comme

$$t = m + \lceil \log_2(2 + \frac{1}{2\epsilon}) \rceil$$

alors QPE nous donne une approximation φ' de φ telle que $|\varphi - \varphi'| \leq \frac{1}{2^m}$ avec une probabilité d'au moins $1 - \epsilon$.

5 L'algorithme de Shor

L'algorithme de Shor est une des applications les plus fascinantes et remarquables de la QFT. Cet algorithme permet de factoriser un nombre composé en temps polynomial, contrairement aux meilleurs algorithmes classiques qui s'exécutent en temps exponentiel. Si des ordinateurs quantiques suffisamment performants et résistants au bruit venaient à voir le jour, des cryptosystèmes couramment utilisés comme *RSA* ou *ElGamal* deviendraient inefficaces. En effet, la sécurité de ces algorithmes repose sur la présumée difficulté de factoriser un nombre composé, ou, plus généralement, de calculer le logarithme discret d'un élément dans un groupe.

5.1 Rappels sur RSA

Rappels succints de RSA :

- $n = pq$ avec p et q premiers
- $e \in \llbracket 1, \phi(n) \rrbracket$ avec $e \wedge \phi(n) = 1$
- $\exists! d \in \llbracket 1, \phi(n) \rrbracket$ tel que $ed \equiv 1[\phi(n)]$

Message clair : $a \in \llbracket 1, n \rrbracket$. Message chiffré : $b \equiv a^e[n]$. La question qu'on peut se poser est : Comment, à partir de b , n et e , retrouver a ?

Considérons r l'ordre de \bar{b} dans $\mathbb{Z}/n\mathbb{Z}$: il peut être trouvé en déterminant la période de $f : x \mapsto \bar{b}^x$. Le sous groupe engendré par b est égal à celui engendré par a . En effet, $\forall x \in \langle \bar{b} \rangle, \exists k \in \mathbb{Z}, x = \bar{b}^k = \bar{a}^{ek} \in \langle \bar{a} \rangle$, et réciproquement $\forall x \in \langle \bar{a} \rangle, \exists k \in \mathbb{Z}, x = \bar{a}^k = \bar{b}^{-ke} \in \langle \bar{b} \rangle$. Donc \bar{a} est également d'ordre r .

Soit maintenant $e' \in \llbracket 1, r \rrbracket$ tel que $e \equiv e'[r]$. On a $e \wedge r = 1$ car $r|\phi(n)$ et $e \equiv e'[r]$. Donc $\bar{e} = \bar{e}'$ est inversible dans $\mathbb{Z}/r\mathbb{Z}$, ie $\exists d' \in \llbracket 1, r \rrbracket, e'd' \equiv 1[r]$, ie $\exists m \in \mathbb{Z}, ed' = 1 + mr$. On a alors $b^{d'} = a^{ed'} = aa^{mr} = a[n]$. d' peut facilement être calculé avec l'algorithme d'Euclide étendu.

On peut même faire encore plus fort que cela. On peut retrouver N , en suivant les étapes suivantes :

1. On choisit $a \in \llbracket 1, n-1 \rrbracket$ au hasard.
2. Si $a \wedge n \neq 1$, alors $a \wedge n$ est p ou q , on a donc gagné (mais n'arrive jamais, très très improbable, autant chercher à la main). Sinon on continue.
3. On calcule son ordre r avec l'algorithme de Shor (sections suivantes). Avec un peu de chance, $2|r$ et $a^{r/2} \not\equiv -1[n]$ (un peu plus de 50% de chances d'avoir un a qui convient). Sinon, goto 1.
4. Si on a obtenu un a qui convient, alors $n|(a^{r/2} - 1)(a^{r/2} + 1)$, mais on a aussi $n \nmid a^{r/2} + 1$ par hypothèse, et $n \nmid a^{r/2} - 1$ car l'ordre de a est r . Donc nécessairement on a (à l'ordre de p et q près) $p|a^{r/2} + 1$ et $q|a^{r/2} - 1$. Alors on peut retrouver p et q avec $p = (a^{r/2} + 1) \wedge n$ et $q = (a^{r/2} - 1) \wedge n$. We won ! :)

5.2 Trouver l'ordre dans un groupe

Le but est donc de trouver r , ordre de a dans $\mathbb{Z}/n\mathbb{Z}$. De manière totalement inattendue et presque miraculeuse, *QPE* permet de faire cela ! Pour voir comment, on pose l'opérateur \mathbf{U} qui agit de la manière suivante pour $x \in [0, n-1]$: $\mathbf{U}|x\rangle = |ax \pmod n\rangle$. On considère également les états suivants, indexés par un nombre $s \in [0, r-1]$:

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2i\pi sk}{r}} |a^k \pmod n\rangle$$

On peut remarquer qu'il s'agit de vecteurs propres de \mathbf{U} :

$$\begin{aligned} \mathbf{U}|u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2i\pi sk}{r}} |a^{k+1} \pmod n\rangle \\ &= \frac{1}{\sqrt{r}} e^{\frac{2i\pi s}{r}} \sum_{k=0}^{r-1} e^{-\frac{2i\pi sk}{r}} |a^k \pmod n\rangle \\ &= e^{\frac{2i\pi s}{r}} |u_s\rangle \end{aligned}$$

On peut donc utiliser QPE pour retrouver les valeurs propres associées, $\frac{2i\pi s}{r}$. On peut alors remonter à r ! Le deuxième registre doit donc être de taille $d = \lceil \log_2(n) \rceil$, pour stocker le vecteur propre. Si on avait un des états $|u_s\rangle$, on pourrait déterminer $\frac{s}{r}$. On sait que QPE nous donne un résultat approché φ de $\frac{s}{r}$ avec une erreur de $\frac{1}{2^t}$ au plus :

$$\left| \frac{s}{r} - \varphi \right| \leq \frac{1}{2^t}$$

En fixant la taille du premier registre à $t = 2\lceil \log_2(n) \rceil + 1$,

$$2^t \geq 2^{2\log_2(n)+1} = 2n^2 \geq 2r^2$$

d'où :

$$\left| \frac{s}{r} - \varphi \right| \leq \frac{1}{2r^2}$$

Un théorème des fractions continues nous permet d'affirmer que dans ce cas, on peut trouver la valeur exacte de $\frac{s}{r}$ dans le développement en fractions continues de φ . On trouve alors s' et r' avec $s' \wedge r' = 1$ et $\frac{s}{r} = \frac{s'}{r'}$. Si s et r sont premiers entre eux, ce qui arrive très souvent, alors on connaît r . Sinon,

on connaît un diviseur de r , généralement assez grand. In fine, on a réussi à casser RSA avec une probabilité assez haute :)

Maintenant, pour réussir à implémenter les états $|u_s\rangle$, il suffit de remarquer que $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle$. Ainsi en initialisant le deuxième registre à $|1\rangle$, on a en sortie une superposition de valeurs propres qui contiennent l'information sur r . En sortie QPE nous donne donc une valeur approchée d'un $\frac{s}{r}$ pour un s quelconque.

5.3 Un exemple : factorisation de 15

Dans cette section, on applique l'algorithme de Shor à la décomposition de 15. On détaille chaque étape.

5.3.1 Calcul de \mathbf{CU}^{2^k}

On a besoin de calculer \mathbf{CU}^{2^k} pour n'importe quel a premier avec 15. Dans la suite, on note $\mathbf{CS}_{i,j}$ le *SWAP* contrôlé. On liste les cas possibles :

1. Cas $a = 1$: cela revient à multiplier par 1, donc à ne rien faire (identité)
2. Cas $a = 2$:
 - Cas $k = 0$: on fait une permutation circulaire.

$$|2 \times x_3 x_2 x_1 x_0 \pmod{15}\rangle = |x_2 x_1 x_0 x_3\rangle$$

L'opération unitaire associée est donc $\mathbf{CS}_{1,0} \mathbf{CS}_{2,1} \mathbf{CS}_{3,2}$.

- Cas $k = 1$: il faut multiplier par 4 (mod 15).

$$|4 \times x_3 x_2 x_1 x_0 \pmod{15}\rangle = |x_1 x_0 x_3 x_2\rangle$$

Autrement dit, on applique $\mathbf{CS}_{3,1} \mathbf{CS}_{2,0}$.

- Cas $k \geq 2$: $4^2 = 1 \pmod{15}$, donc en posant la division euclidienne de k par 2, $k = 2p + r$, on a

$$|2^{2^k} \times x \pmod{15}\rangle = |4^{2^{2(p-1)+r}} x \pmod{15}\rangle = |x\rangle$$

car $2(p-1) + r \geq 1$. On ne fait rien.

3. Cas $a = 4$: traité dans le cas $a = 2$
4. Cas $a = 7$:
 - Cas $k = 0$: on peut remarquer que la multiplication modulo 7 revient à appliquer un NON à une permutation circulaire :

$$|7 \times x_3 x_2 x_1 x_0 \pmod{15}\rangle = |\overline{x_0 x_3 x_2 x_1}\rangle$$

Autrement dit, on applique $\mathbf{CX}_3 \mathbf{CX}_2 \mathbf{CX}_1 \mathbf{CX}_0 \mathbf{CS}_{3,2} \mathbf{CS}_{2,1} \mathbf{CS}_{1,0}$

- Cas $k = 1$: comme $7^2 = 4 \pmod{15}$, on se ramène au cas $a = 4$
- Cas $k \geq 2$: idem $a = 2$, $k \geq 2$, on ne fait rien

5. Cas $a = 8$:
 - Cas $k = 0$: permutation circulaire vers la droite.

$$|8 \times x_3 x_2 x_1 x_0 \pmod{15}\rangle = |x_0 x_3 x_2 x_1\rangle$$

On applique donc $\mathbf{CS}_{3,2} \mathbf{CS}_{2,1} \mathbf{CS}_{1,0}$.

- Cas $k = 1$: $8^2 = 4 \pmod{15}$, donc on se ramène au cas $k = 0$, $a = 4$.
- Cas $k \geq 2$: toujours la même chose, on ne fait rien.

6. Cas $a = 11$:

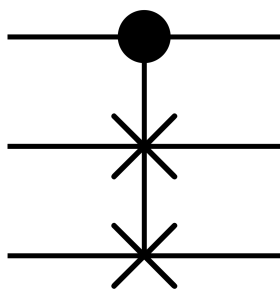


FIGURE 6 – Porte de Fredkin

— Cas $k = 0$: on permute puis on fait une négation.

$$|7 \times x_3 x_2 x_1 x_0 \pmod{15}\rangle = |\overline{x_1 x_0 x_3 x_2}\rangle$$

On fait agir $\mathbf{CX}_3 \mathbf{CX}_2 \mathbf{CX}_1 \mathbf{CX}_0 \mathbf{CS}_{3,1} \mathbf{CS}_{2,0}$.

— Cas $k \geq 1$: $11^2 \pmod{15} = 1$, donc on ne fait rien.

7. Cas $a = 13$:

— Cas $k = 0$: on permute puis on fait une négation.

$$|7 \times x_3 x_2 x_1 x_0 \pmod{15}\rangle = |\overline{x_2 x_1 x_0 x_3}\rangle$$

On fait agir $\mathbf{CX}_3 \mathbf{CX}_2 \mathbf{CX}_1 \mathbf{CX}_0 \mathbf{CS}_{1,0} \mathbf{CS}_{2,1} \mathbf{CS}_{3,2}$.

— Cas $k \geq 1$: comme $13^2 \pmod{15} = 4$, on se ramène au cas $a = 4$.

8. Cas $a = 14$:

— Cas $k = 0$: il s'agit simplement d'un NON.

$$|14 \times x_3 x_2 x_1 x_0 \pmod{15}\rangle = |\overline{x_3 x_2 x_1 x_0}\rangle$$

On applique $\mathbf{CX}_3 \mathbf{CX}_2 \mathbf{CX}_1 \mathbf{CX}_0$.

— Cas $k \geq 1$: comme $14^2 = 1 \pmod{15}$, on ne fait rien.

5.3.2 Implémentation d'une porte de Fredkin

Ouf! On a toutes nos opérations possibles. Le problème implicite, c'est qu'on n'a pas explicité l'expression des SWAP contrôlés, aussi appelés *portes de Fredkin* (figure 6). Pour cela, on peut décomposer un SWAP en CNOTs, pour se ramener à des portes de Toffoli. En effet,

$$\mathbf{S} = (\mathbf{C}_0^1 \mathbf{X})(\mathbf{C}_1^0 \mathbf{X})(\mathbf{C}_0^1 \mathbf{X})$$

Or, toute porte de Toffoli peut s'implémenter de la manière suivante, à l'aide de la racine carrée $\mathbf{V} = e^{i\frac{\pi}{4}} \mathbf{R}_{\mathbf{X}}(\frac{\pi}{2})$ de \mathbf{X} :

$$\mathbf{C}_0^{2,1} \mathbf{X} = (\mathbf{C}_0^2 \mathbf{V})(\mathbf{C}_1^2 \mathbf{X})(\mathbf{C}_0^1 \mathbf{V}^\dagger)(\mathbf{C}_1^2 \mathbf{X})(\mathbf{C}_0^1 \mathbf{V})$$

ce qui est représenté figure 7. Donc au final :

$$\mathbf{C}_{1,0}^2 \mathbf{S} = (\mathbf{C}_0^{2,1} \mathbf{X})(\mathbf{C}_1^{2,0} \mathbf{X})(\mathbf{C}_0^{2,1} \mathbf{X})$$

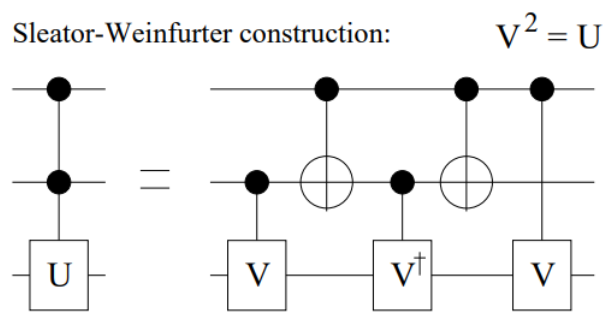


FIGURE 7 – Construction d'une Toffoli