Chapter 6

Modèle des Circuits et Algorithmes Quantique

Ce chapitre est une première introduction au calcul quantique. Après une brève introduction historique et une discussion de la notion de circuit classique, nous introduisons le modèle de Deutsch des "circuits quantiques" (1995). Comme nous allons le voir ce modèle sert à définir ce qui sera pour nous un algorithme quantique, tout en fournissant une représentation très concrète de ces algorithmes. Enfin, nous illustrons ces notions grâce à un algorithme quantique simple mais important, l'algorithme de Deutsch et Josza. Cet algorithme quantique contient déjà la plupart des ingrédients d'une classe plus large, qui traite le problème plus général de la recherche de symétries cachées. Le célèbre algorithme de Shor (1997), permettant la factorisation d'un nombre entier en temps polynomial (par rapport aux nombres de bits de l'entier) appartient à cette classe. Celui-ci fera l'objet des chapitres suivants. Il est suspecté, mais pas démontré, que cette famille d'algorithmes quantiques permet une accélération exponentielle du temps de calcul, par rapport aux algorithmes classiques.

6.1 Brève introduction historique

Un calcul est de facon ultime réalisé par un dispositif physique. Il est donc naturel de se demander quelles sont les limites fondamentales que les lois de la physique imposent au calcul. Un travail précurseur fut celui de Landauer (dans les années 60-70) qui montra que l'effacement d'un bit - un processus irréversible - est toujours accompagné d'une dissipation de chaleur. Essentiellement, ceci provient de l'augmentation de l'entropie du système due a l'effacement du bit (perte d'information) et des lois de la thermodynamique

reliant la variation d'entropie d'un système au flux de chaleur entre le système et son environnement. En conséquence, tout calcul utilisant des portes logiques *irréversibles* (par exemple AND, OR) dissipe de la chaleur. Mais existe-il un principe fondamental qui nécessite absolument une dissipation minimale de chaleur lors d'un calcul ? Ou bien pourrait-on, en théorie du moins, éliminer la dissipation de chaleur lors du calcul ? Une *réponse positive* à cette deuxième question a été présentée par Bennett, Benioff et d'autres. Plus précisément, *tout calcul irréversible peut être rendu réversible*, grâce à des portes élémentaires appropriées. Néanmoins il y a un cout: l'espace de stockage doit être augmenté pour éliminer les effacements.

En l'absence de la chaleur générée par un calcul réversible, lorsque le support physique des bits atteint les dimensions moléculaires ou atomiques et que la température du système est maintenue très basse, le comportement quantique de la matière et la cohérence des états quantiques (le principe de superposition) deviennent importants. Il est naturel de se poser la question suivante: quels sont les effets du comportement quantique de la matière sur le calcul ? Est ce que les effets quantiques aident, ou bien au contraire apportent-ils de nouvelles limites ?

Ces questions ont été soulevées et discutées par Feynman, Benioff et Manin au début des années 1980. En principe la MQ n'apporte pas de nouvelles limitations. Au contraire ! Le principe de superposition appliqué à des systèmes à plusieures particules (plusieurs qubits) nous permet d'effectuer des "calculs parallèles". Ce "paralélisme quantique" accélère les calculs classiques, et parfois même de facon exponentielle. Cela a été reconnu notamment par Feynman qui a fait valoir que les ordinateurs classiques ne peuvent simuler des processus quantiques "efficacement" (du point de vue du temps de calcul). Feynman a suggérer que nous devrions construire des "machines quantiques" pour simuler efficacement les processus quantiques.

La raison fondamentale pour laquelle le calcul classique ne peut pas simuler efficacement les processus quantiques est la suivante. Un état général quantique de n bits quantiques contient une superposition de 2^n "états classiques":

$$|\Psi\rangle = \sum_{b_1 \ldots b_N \in \{\{0,1\}\}^N} C(b_1, \ldots, b_N)|b_1, \ldots, b_N\rangle$$

Ici $|b_1, \ldots, b_N\rangle = |b_1\rangle \otimes \ldots \otimes |b_N\rangle$ (avec $b_i = 0, 1$) sont les états de la base dite "computationnelle". Une simulation classique de l'évolution du ket $|\Psi\rangle$ doit effectuer essentiellement 2^n calculs pour l'évolution de suite binaire classique $(b_1 \ldots b_N)$. Au contraire, la dynamique quantique unitaire U agit sur $|\Psi\rangle$ dans son ensemble (ou en parallèle sur chaque ket $|b_1 \ldots b_N\rangle$). Un dispositif physique réalisant la dynamique unitaire U sur $|\Psi\rangle$ peut être considéré comme

un ordinateur quantique.

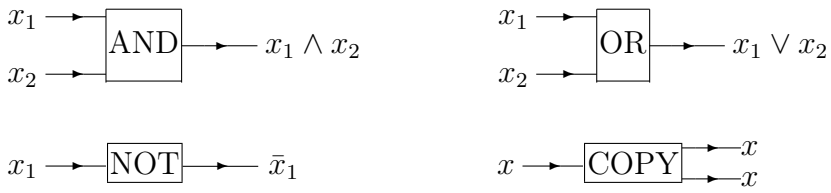
Un calcul classique peut être représenté par un modèle de circuit construit à partir d'un ensemble donné de portes élémentaires universelles agissant de manière récursive sur l'entrée du calcul. En 1985 David Deutsch a montré que la même chose est valable dans le cas quantique. Notamment, toute évolution unitaire peut être assez bien approchée par un ensemble universel de portes quantiques universelles.

Il existe aussi d'autres modèles d'ordinateur quantique mais le modèle de Deutsch - un modèle de circuit quantique - est le plus populaire aujourd'hui. Un des buts de ce chapitre est d'expliquer ce modèle. Une des raisons de sa popularité est qu'il s'agit d'un modèle universel: en principe, tout calcul quantique peut être représenté comme un circuit quantique construit à partir d'un ensemble restreint de porte logiques quantiques. De plus cette repré sentation est relativement concrète.

Il existe aussi une notion de machine de Turing quantique (analogue aux machines de Turing classiques) qui est le cadre naturel pour discuter des classes de complexité quantiques. Il a été démontré que le modèle de machine de Turing quantique est équivalent au modèle des circuits quantiques. Cet aspect ne sera pas abordé ici.

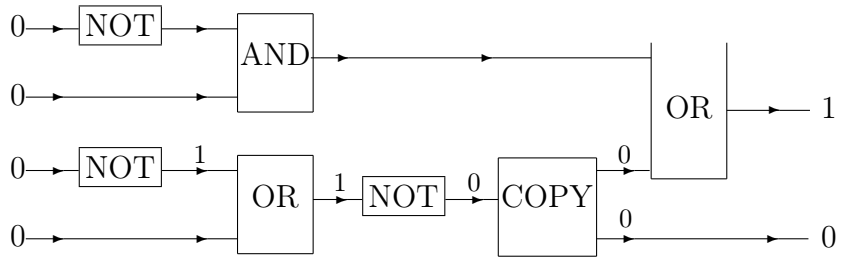
6.2 Modèle des circuits pour le calcul classique

Nous discutons brièvement le calcul classique basé sur les circuits booléens. Considérons les portes logiques classiques de base $x_i \in \mathbf{F}_2 = \{0, 1\}$.



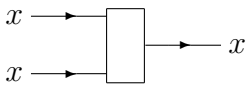
L'opération COPY s'appelle aussi parfois FANOUT. Cet ensemble de portes peut être utilisé pour définir les circuits Booléens.

Un circuit Booléen est un graphe acyclique (sans cycles) dirigé (les liens ont une direction) avec n bits d'entrée et m bits de sortie. L'entrée peut toujours être initialisée à $(0 \ldots 0)$ car tout $(x_1 \ldots x_n)$ est obtenu par une série de portes NOT. La figure suivante illustre cette definition.



Un celèbre théorème d'Emil Post (∼1950) montre que toute fonction $f : \mathbf{F}_2^n \to \mathbf{F}_2^m$ peut être réalisée par un circuit Booléen. Plus précisément, pour toute fonction $f : \mathbf{F}_2^n \to \mathbf{F}_2^m$ il existe un circuit Booléen qui applique les entrées $(x_1 \dots x_n)$ sur les sorties $(y_1 \dots y_m) = f(x_1 \dots x_n)$. Le circuit est entièrement construit avec les portes NOT, AND, OR, COPY et est un graphe acyclique dirigé. On dit que l'ensemble des portes (NOT, AND, OR, COPY) est universel.

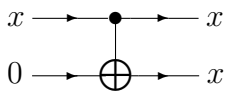
La porte NOT est logiquement réversible. Cela veut dire qu'à partir de la sortie il est possible de récupérer l'entrée. Les portes AND et OR elles, sont logiquement irréversibles. Il est impossible de reconstruire l'entrée à partir de la sortie. L'irréversibilité logique entraine l'irréversibilité physique c.a.d une dissipation de chaleur lors du calcul. En effet la perte d'nformation et l'augmentation d'entropie est reliée à un flux de chaleur du système vers l'environnment. La porte COPY quand à elle, est logiquement réversible, mais physiquement irréversible. En effet l'operation inverse



efface un bit et, comme Landauer l'a montré, cela entraîne une dissipation de chaleur.

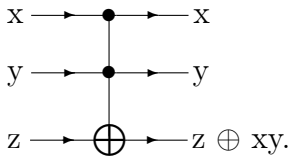
Bennett à montré que n'importe quel circuit Booléen peut être simulé ou remplacer par un circuit réversible. De plus il existe un ensemble universel de portes logiques réversibles (logiquement et physiquement réversibles). Nous n'allons pas donner cette preuve ici mais nous contenter de l'idée essentielle: on peut toujours remplacer les portes AND, OR et COPY par des portes réversibles à condition d'utiliser des bits auxiliaires.

Commencons par la porte COPY. Elle peut être remplacée par



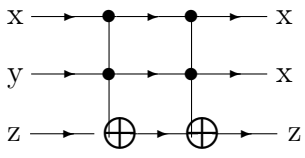
qui est une porte control-NOT (aussi notée CNOT) réversible utilisant deux bits. Le bit de stockage est égal à 0 dans l'entrée.

Pour les portes AND et OR nous utilsons la porte de Toffoli. Celle-ci n'est rien d'autre qu'un CCNOT (control-control-NOT) et utilise trois bits.



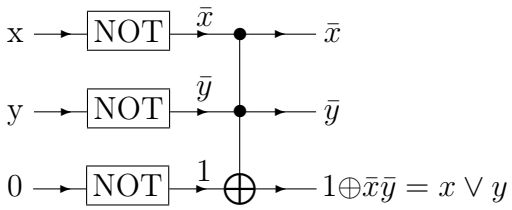
ou $\mathrm{T}(x, y, z) = (x, y, z \oplus xy)$.

Cette porte flip le bit z si les deux bits de controle x et y sont égaux à 1. Sinon z est inchangé. La porte de Toffoli est réversible car



or $\mathrm{T}^2(\mathrm{x}, \mathrm{y}, \mathrm{z}) = (\mathrm{x}, \mathrm{y}, \mathrm{z})$

Si $z = 0$, $T(x, y, 0) = (x, y, xy)$ retourne $x \wedge y$ pour le troisième bit. Pour la porte OR on utilise



Résumons: un circuit Booléen utilisant {AND, OR, COPY, NOT} peut être remplacé par un circuit utilisant les portes universelles {CNOT; Toffoli; NOT}. L'ensemble {AND, OR, COPY, NOT} contient uniquement des portes à un et deux bits, alors que {CNOT; Toffoli; NOT} fait intervenir une porte à trois bits (Toffoli). On peut montrer qu'il n'est pas possible de se passer de portes à tris bits si on veut la réversibilité d'un circuit classique. Nous verrons que dans le cas quantique cela est possible !

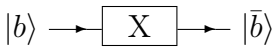
6.3 Circuits quantiques.

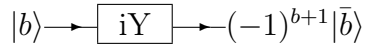
Les circuits quantiques sont analogues aux circuits classiques. En particulier, ils sont construits à partir d'un petit ensemble de "portes quantique" élementaires. Une "porte quantique" n'est rien d'autre qu'une opération unitaire qui agit sur un petit nombre de qubits (typiquement un ou deux qubits). Ces portes sont réversibles car une opération unitaire est inversible, en effet $UU^\dagger = U^\dagger U = I$. Nous verrons dans des chapitres ultérieurs comment elles sont réalisées en pratique. Nous commencons par discuter certaines de ces portes élémentaires.

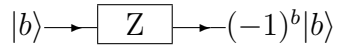
6.3.1 Portes à un qubit

Les portes à un qubit sont des matrices unitaires qui agissent sur les vecteurs d'état de l'espace d'Hilbert $\mathbf{C}^2$. Certaines de ces matrices joueront un role particulièrement important.

- Les trois "matrices de Pauli" $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$; $iY = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ and $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. Ce sont bien des matrices unitaires (attention: Y n'est pas unitaire mais hermitienne, néanmoins iY est unitaire; X et Z sont unitaires et hermitiennes).







  La porte X n'est rien d'autre que la porte NOT quantique. Dans le cas quantique l'entrée peut être une superposition cohérente des états $\{|0\rangle, |1\rangle\}$. Par exemple

$$X(\alpha|0\rangle + \beta|1\rangle) = \alpha|1\rangle + \beta|0\rangle.$$

  Ainsi l'action de la porte NOT est beaucoup plus générale dans le cas quantique. Cette remarque est simple mais cruciale et profonde (et s'applique à toutes les portes quantiques discutées ci-dessous) !

- La porte de Hadamard $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$$|b\rangle \longrightarrow \boxed{\text{H}} \longrightarrow H|b\rangle = \tfrac{1}{\sqrt{2}}\big(|0\rangle + (-1)^b|1\rangle\big)$$

Cette porte n'a pas d'analogue classique.

- La porte $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$

$$|b\rangle \longrightarrow \boxed{\text{T}} \longrightarrow e^{ib\pi/4}|b\rangle = \begin{cases} |0\rangle \\ e^{i\pi/4}|1\rangle \end{cases}$$

Elle agit sur les superpositions comme

$$T(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + \beta e^{i\pi/4}|1\rangle$$

- La porte $S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$

$$|b\rangle \longrightarrow \boxed{\text{S}} \longrightarrow e^{ib\pi/2}|b\rangle = \begin{cases} |0\rangle \\ i|1\rangle \end{cases}$$

Elle agit sur les superpositions comme

$$S(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + i|1\rangle$$

On peut montrer que toute matrice unitaire 2×2 peut être approximée avec une précision arbitraire par des produits des matrices élémentaires H et T. Notez que $S = T^2$.

6.3.2 Portes à deux qubits

Les portes à deux qubits sont des matrices unitaires qui agissent sur les vecteurs d'état de l'espace d'Hilbert $\mathbf{C}^2 \otimes \mathbf{C}^2$. La plus importante est la porte control-NOT quantique.

- La porte CNOT (controlled not) est définie par:

$$\begin{array}{ll} |c\rangle \longrightarrow \bullet \longrightarrow |c\rangle & \text{control bit.} \\ |t\rangle \longrightarrow \oplus \longrightarrow |c \oplus t\rangle & \text{target bit.} \end{array}$$

Dans la base

$$|0\rangle \otimes |0\rangle \; ; \quad |0\rangle \otimes |1\rangle \; ; \quad |1\rangle \otimes |0\rangle \; ; \quad |1\rangle \otimes |1\rangle$$

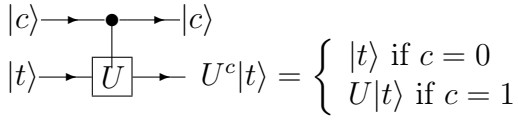
$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \qquad \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

la représentation matricielle est

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

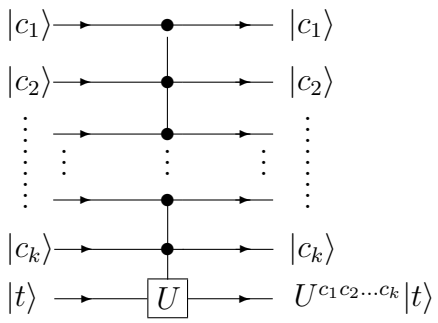
Nous remarquons que cette porte agit en général sur des superpositions cohérentes d'états à deux qubits.

- Une généralisation importante est la porte control-U ou U est une opération unitaire à un qubit:



6.3.3 Portes muti-control-U

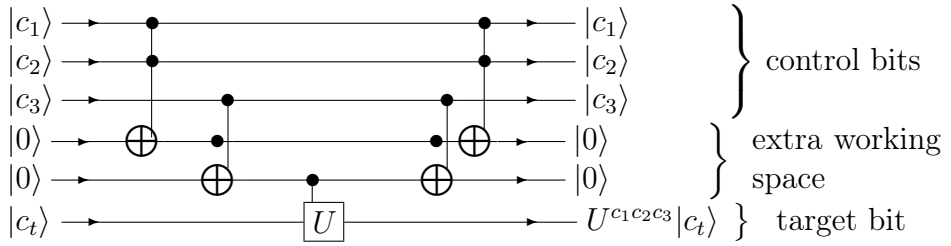
Une généralisation des portes précédentes est



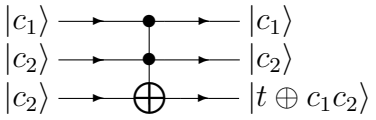
La porte U agit sur le dernier bit si tous les bits de controle sont égaux à 1. A nouveau il est important de remarquer que ces portes agissent sur des

superpositions cohérentes détats de base de l'espace de Hilbert $\mathbf{C}^2 \otimes \ldots \mathbf{C}^2$. Ce sont des matrices $2^n \times 2^n$.

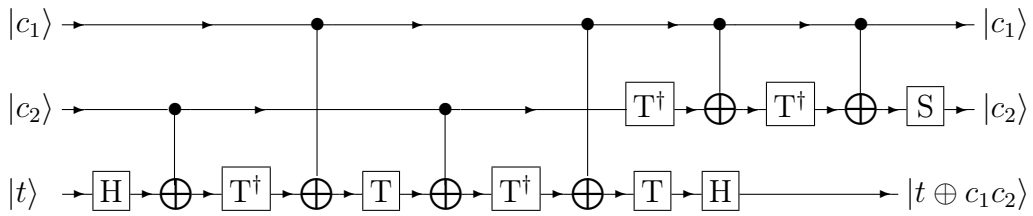
En augmentant le nombre de qubits auxiliaires la porte multi-control-U peut être réalisée grâce à une concaténation d'un control-control-NOT and un control-U. En effet (exercices):



La porte control-control-NOT gate s'appelle aussi porte de Toffoli quantique (la différence avec la porte classique est que les entrées peuvent être des superpositions d'états). Remarquablement cette porte quantique peut être représentée par des portes à un & deux qubits {T, S, H, CNOT}. On vérifie que:



est équivalent au circuit suivant (exercices):



Résumons cette discussion: Toute porte multi-control-U de dimension $2^n \times 2^n$ peut être réalisée grace à l'ensemble {T, S, H, CNOT, U} où U est une porte à un qubit. De plus nous avons aussi vu que U peut être approximée avec une précision arbitraire par un produit de matrices H et T.

6.3.4 Le modèle des circuits quantiques de Deutsch

Le dernier résultat du paragraphe précédent peut être généralisé. Un théorème important affirme que toute opération unitaire agissant sur l'espace à n qubits

(c.a.d toute matrice $2^n \times 2^n$) peut être approximée avec une précision arbitraire par l'ensemble des portes {T, S, H, CNOT}. Ce théorème est à la base du modèle des circuits quantiques.

La complexité du circuit dépendra du nombre de matrices utilisées pour approximer l'opération unitaire à n qubits. Nous verrons par exemple que l'opération unitaire utilisée pour la factorisation des entiers requiert $O(\text{poly}N)$ portes quantiques élémentaires. Cela n'est pas possible avec un circuit classique. Néanmoins on sait qu'il existe des matrices unitaires qui requièrent un nombre exponentiellement grand (en n) de portes élémentaires.

Définissons maintenent le modèle des circuits:

a) Un circuit quantique est graphe dirigé acyclique avec les vertex qui sont les portes {T, S, H, CNOT}. Les liens "portent" des qubits $(\alpha|0\rangle+\beta|1\rangle)$.

b) L'entrée est donnée par l'état produit:

$$|0\rangle \otimes ... \otimes |0\rangle$$

c) La sortie est le résultat de l'évolution unitaire. Cette sortie prend la forme générale:

$$|\Psi\rangle = \sum_{c_1...c_n} A(c_1...c_n)|c_1c_2...c_n\rangle$$

ou $A(c_1...c_n)$ sont des coefficients complexes et $|c_1c_2...c_n\rangle = |c_1\rangle\otimes\ldots|c_n\rangle$ sont les états de la base computationnelle.

d) Finalement une opération de mesure est effectuée sur $|\Psi\rangle$ avec un appareil mesurant dans la "base computationnelle" $\{|c_1c_2...c_n\rangle, c_i = 0, 1\}$. Le résultat de l'opération de mesure est l'état $|c_1...c_n\rangle$ avec probabilité $|\mathrm{A}(c_1...c_n)|^2$ (règle de Born). Le résultat du calcul quantique est donc un résultat probabiliste. En pratique l'algorithme est bon si la probabilité est concentrée sur le résultat cherché. La plupart du temps on répète l'expérience (le calcul du circuit) pour amplifier cette probabilité (nous verrons cela en pratique).

Remarquons les points importants:

1. Des "qutrits" au lieu des "qubits" ne changeraient rien de fondamental (et la nature offre des qubits).

2. Faire les opérations de mesure à des stades intermédiares ou bien à la fin ne change rien.

3. Faire les opérations de mesure dans une autre base est équivalent à faire des opération de changement de base - qui sont unitaires car toutes les bases de mesure sont orthonormées - et donc à changer le circuit et faire la mesure dans la base computionnelle. Néanmoins cela peut modifier la complexité (la réduire ou l'agrandir).

4. Commencer avec une autre entrée est aussi équivalent à ajouter des opérations unitaires et donc à changer le circuit et initialiser avec l'entrée $|0, \cdots, 0\rangle$. Néanmoins cela peut modifier la complexité (la réduire ou l'agrandir)..

5. Il existe aussi d'autres ensembles de portes universelles.

6. Le calcul quantique est réversible (pas de perte d'information, pas d'augmentation d'entropie et pas de dissipation de chaleur) car le circuit est une opération unitaire, tant que l'opération de mesure n'a pas été effectuée.

7. Une opération classique réversible peut être représentée par une opération unitaire quantique. En effet:

$$\tilde{f}(x_1...x_n, y) = (x_1...x_n, y \oplus f(x_1...x_n))$$

induit l'unitaire

$$U_f|x_1...x_n, y\rangle = |x_1...x_n, y \oplus f(x_1...x_n)\rangle.$$

Si la sortie $f(x_1...x_n)$ possède m composantes on prend y avec m composantes et l'addition modulo 2 (le XOR) est faite composante par composante. On vérifie aisément que U_f est unitaire, en vérifiant que la matrice préserve le produit scalaire.

8. Le point précédent montre que tout calcul réversible classique est contenu dans le modèle des circuits quantiques.

9. La puissance du calcul quantique vient de l'action simultané ou parallèle de l'évolution unitaire sur toutes les "suites classiques" $c_1 \dots c_n$ d'un état de n qubits. C'est ce qu'on appelle parfois le parallélisme quantique: celui-ci provient des principes 1 (superposition) et 5 (produit tensoriel des systèmes composés) mis ensemble. La complexité du calcul est donné par la taille du circuit. Le résultat est aléatoire. En général il faut répéter un certain nombre de fois que le calcul quantique pour obtenir le résultat voulu avec une probabilité proche de 1. Cette répétition peut augmenter la complexité.

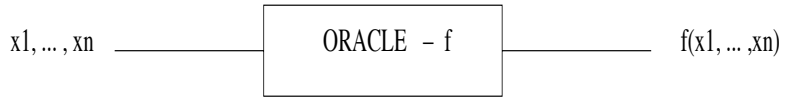


Figure 6.1: Oracle classique retournant les valeurs de la fonction f.

6.4 Le problème de Deutsch-Jozsa

L'algorithme de Deutsch et Jozsa est probablement l'algorithme quantique le plus simple. Dans sa version initiale il fut inventé par David Deutsch dans son article fondateur[1] en 1985. L'algorithme fut ensuite amélioré par Deutsch et Jozsa (1992) et finalement par Cleve-Ekert-Macchiavello-Mosca (1998). Cette version finale fait clairement apparaître que l'algorithme est le prototype d'une classe plus vaste, étudiée dans les chapitre ultérieurs, basée sur la *transformée de Fourier quantique* et le *principe d'interférence des chemins quantiques*. De plus, il constitue une très bonne illustration d'un cas où l'on peut tirer parti du *parallélisme quantique* de facon assez spectaculaire.

Formulons d'abord le problème à résoudre. Soit

$$f : \mathbf{F}_2^n \to \mathbf{F}_2, \qquad (x_1 \dots x_n) \mapsto f(x_1 \dots x_n) \in \{0, 1\}$$

une fonction booléenne dont on sait a priori qu'elle est *constante* ou *balancée*. La fonction est constante si l'image prend toujours la même valeur quelle que soit l'arguement $(x_1 \dots x_n)$. Notez qu'il y a 2^n arguments possibles. *Balancée* signifie que pour une moitié des arguments (c.a.d 2^{n-1}) elle prend la valeur 0 et pour l'autre moitié (c.a.d 2^{n-1}) elle prend la valeur 1.

Le problème de Deutsch-Jozsa est un *problème de décision avec Oracle*. Cela veut dire que l'on ne connaît pas la fonction f mais que l'on a à disposition un *Oracle* qui donne la réponse $f(x_1 \dots x_n)$ pour toute entrée $x_1 \dots x_n$ qui lui est soumise. Le problème est de décider si f est constante. Le nombre de questions nécessaires à l'Oracle détermine la complexité temporelle de la résolution. Le but est de prendre la décision correcte en posant le moins de questions possibles.

Discutons d'abord la solution classique. Si on se limite à utiliser un *algorithme déterministe*, il existe des fonctions f pour lesquelles la complexité temporelle est de $2^{n-1} + 1$, c'est-à-dire exponentielle par rapport à la taille des entrées. En effet supposons que f soit constante et prenne la valeur 0 si bien que l'Oracle retourne toujours la réponse 0. Si l'on pose strictement moins de $2^{n-1} + 1$ questions à l'Oracle on n'a aucun moyen de savoir si la

[1] *Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer* Proc. Roy. Soc of London A **400** pp 97-117 (1985)

prochaine réponse sera aussi 0. Par contre si à la $2^{n-1}+1$ ième question la réponse est 0 alors on peut affirmer avec certitude que la fonction n'est pas balancée car si elle l'était cette réponse aurait été 1.

Notons que si la fonction est balancée le nombre minimum de questions à poser est deux et le maximum est $2^{n-1}+1$. Le point important est qu'il existe des situations défavorables qui nécessitent un nombre exponentiel de questions.

Nous allons voir qu'il existe un algorithme quantique qui permet de déterminer si f est balancée ou constante avec une et une seule utilisation de l'Oracle et ceci quel que soit la fonction f. Cela est assez spectaculaire. Notons que par rapport à un algorithme classique déterministe, le gain est exponentiel.

6.5 L'Oracle quantique

Pour chaque n on construit un circuit qui constitue l'algorithme. Les constituants du circuit sont la porte quantique de Hadamard et un *Oracle quantique*. Ici nous discutons la modélisation de l'Oracle.

L'Oracle quantique est une porte donnée (par la Nature par exemple; c'est-à-dire que cela pourrait être un système physique) qui effectue l'opération *unitaire* suivante :

$$U_f|x_1 \dots x_m, y\rangle = |x_1 \dots x_m, y \oplus f(x_1 \dots x_m)\rangle$$

Cet opérateur agit sur un Ket de Dirac à plusieurs qubits appartenant à l'espace

$$\underbrace{\mathbf{C}^2 \otimes \mathbf{C}^2 \dots \mathbf{C}^2}_{n \text{ fois}} \otimes \mathbf{C}^2$$

et donne un autre ket appartenant au même espace. C'est donc une matrice de dimensions $2^{n+1} \times 2^{n+1}$.

L'Oracle agit donc comme une porte *multicontrôle* (c.a.d qu'il y a plusieurs qubits de contrôle). Son circuit quantique est représenté sur la figure 6.2 Vérifions que l'on a bien à faire a une matrice *unitaire*. Pour cela il suffit de montrer qu'elle préserve le produit scalaire. Prenons deux vecteurs

$$U_f|x_1 \dots x_m, y\rangle \text{ et } U_f|x'_1 \dots x'_m, y'\rangle$$

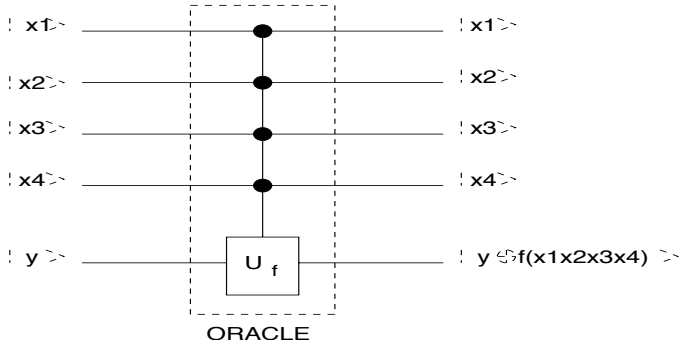


Figure 6.2: L'Oracle quantique retourne le résultat de f stocké dans les bits auxiliaires

et effectuons les produits scalaires :

$$\begin{aligned}\langle x'_1 \dots x'_m, y' | U_f^\dagger U_f | x_1 \dots x_m, y\rangle &= \langle x'_1 \dots x'_m, y' \oplus f(x'_1 \dots x'_m) | x_1 \dots x_m, y \oplus f(x_1 \dots x_m)\rangle \\ &= \langle x'_1 | x_1\rangle \dots \langle x'_m | x_m\rangle \langle y' + f(x'_1 \dots x'_m) | y + f(x_1 \dots x_m)\rangle \\ &= \delta_{x_1 x'_1} \dots \delta_{x_m x'_m} \langle y' + f(x'_1 \dots x'_m) | y + f(x_1 \dots x_m)\rangle \\ &= \delta_{x_1 x'_1} \dots \delta_{x_m x'_m} \delta_{y' y}\end{aligned}$$

D'autre part

$$\begin{aligned}\langle x'_1 \dots x'_m, y' | x_1 \dots x_m, y\rangle &= \langle x'_1 | x_1\rangle \dots \langle x'_m | x_m\rangle \langle y' | y\rangle \\ &= \delta_{x'_1 x_1} \dots \delta_{x'_m x_m} \delta_{y' y}\end{aligned}$$

6.6 Algorithme quantique de Deutsch-Jozsa

Pour chaque n on construit le circuit de la figure 6.3 L'algorithme est initialisé dans l'état (instant t_0).

$$\underbrace{|0\rangle \otimes \dots \otimes |0\rangle}_{n \text{ fois}} \otimes |0\rangle = |\Psi_{in}\rangle$$

et se termine par une mesure dans la base computationnelle des n premiers qubits. Les projecteurs utilisés dans la mesure sont

$$P(b_1 \dots b_n) = |b_1 \dots b_n\rangle\langle b_1 \dots b_n|$$

Nous allons analyser l'évolution temporelle effectuée par un circuit aux instants t_{in}, t_2, t_3, t_{fin}. L'état final est donné par

$$|\Psi_{\text{fin}}\rangle = U(t_{\text{fin}}, t_{\text{in}})|\Psi_{in}\rangle = U(t_{\text{fin}}, t_4)U(t_4, t_3)U(t_3, t_2)U(t_2, t_{\text{in}})|\Psi_{in}\rangle$$

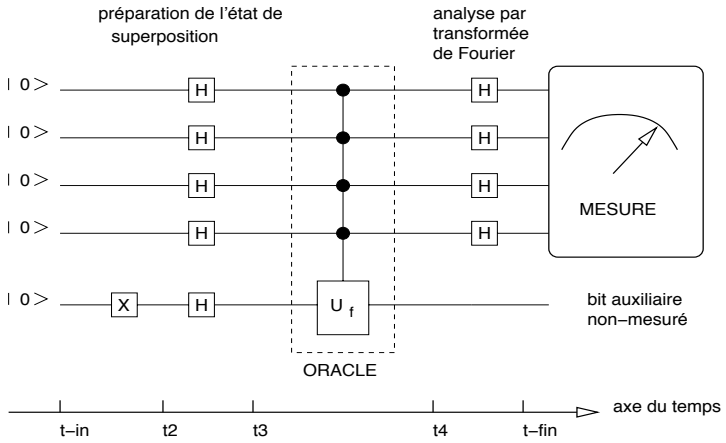


Figure 6.3: Circuit de l'algorithme de Deutsch-Jozsa

Les opérations d'évolution de chaque tranche sont

$$U(t_2, t_{\text{in}}) = (\underbrace{Id \otimes Id \otimes \cdots \otimes Id}_{n \text{ fois}}) \otimes X$$
$$U(t_3, t_2) = (\underbrace{H \otimes H \otimes \cdots \otimes H}_{n \text{ fois}}) \otimes H$$
$$U(t_4, t_3) = U_f$$
$$U(t_{\text{fin}}, t_2) = (\underbrace{H \otimes H \otimes \cdots \otimes H}_{n \text{ fois}}) \otimes Id$$

Etat à l'instant t_3.

$$\underbrace{(H \otimes H \otimes \cdots \otimes H)}_{n \text{ fois}} \otimes HX(\underbrace{|0\rangle \otimes \cdots \otimes |0\rangle}_{n \text{ fois}}) \otimes |0\rangle$$
$$= (\underbrace{H|0\rangle \otimes H|0\rangle \otimes \cdots \otimes H|0\rangle}_{n \text{ fois}}) \otimes HX|0\rangle$$
$$= \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \cdots \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes H|1\rangle$$
$$= \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \cdots \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$
$$= \left(\frac{1}{2^{\frac{n}{2}}} \sum_{b_1 \ldots b_n} |b_1 \ldots b_n\rangle\right) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

A cet instant l'entrée est une *superposition cohérente de toutes les entrées classiques possibles*. Le dernier bit $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ est un bit auxiliaire qui va servir à stocker le résultat de l'Oracle.

Etat à l'instant t_4.

$$\begin{aligned} U_f \frac{1}{2^{\frac{n}{2}}} \sum_{b_1 \dots b_n} |b_1 \dots b_n\rangle \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - |1\rangle \right) \\ = \frac{1}{2^{\frac{n}{2}}} \sum_{b_1 \dots b_n} \left(\frac{1}{\sqrt{2}} U_f|b_1 \dots b_n, 0\rangle - \frac{1}{\sqrt{2}} U_f|b_1 \dots b_n, 1\rangle \right) \\ = \frac{1}{2^{\frac{n}{2}}} \sum_{b_1 \dots b_n} \left(\frac{1}{\sqrt{2}}|b_1 \dots b_n, f(b_1 \dots b_n)\rangle - \frac{1}{\sqrt{2}}|b_1 \dots b_n, \overline{f(b_1 \dots b_n)}\rangle \right) \end{aligned}$$

Notons que si $f(b_1 \dots b_n) = 0$ le terme entre parenthèses vaut

$$|b_1 \dots b_n\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

et que si $f(b_1 \dots b_n) = 1$ le terme entre parenthèses vaut

$$|b_1 \dots b_n\rangle \otimes \frac{|1\rangle - |0\rangle}{\sqrt{2}}$$

On peut donc écrire l'état à l'instant t_4 comme suit:

$$\left(\frac{1}{2^{\frac{n}{2}}} \sum_{b_1 \dots b_n} (-1)^{f(b_1 \dots b_n)} |b_1 \dots b_n\rangle \right) \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Cet état est à nouveau une superposition cohérente ou l'Oracle a déphasé chaque entrée classique de 0 à π suivant que[2] l'image de f est 0 ou 1.

Etat à l'instant t_{fin}. On applique finalement l'opérateur unitaire $\underbrace{H \otimes H \otimes \cdots \otimes H}_{n \text{ fois}} \otimes Id$, ce qui donne par linéarité :

$$|\Psi_{\text{fin}}\rangle = \left(2^{\frac{n}{2}} \sum_{b_1 \dots b_n} (-1)^{f(b_1 \dots b_n)} H|b_1\rangle \otimes \cdots \otimes H|b_n\rangle \right) \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Notons que

$$H|b\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + (-1)^b |1\rangle \right) = \frac{1}{\sqrt{2}} \sum_{c=0,1} (-1)^{cb} |c\rangle$$

[2]Car $e^{\mathrm{i}0} = 1$ et $e^{\mathrm{i}\pi} = -1$.

si bien que

$$\begin{aligned} H|b_1\rangle \otimes \cdots \otimes H|b_n\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{b_1}|1\rangle\right) \otimes \cdots \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{b_n}|1\rangle\right) \\ &= \frac{1}{\sqrt{2}} \sum_{c_1=0,1} (-1)^{c_1 b_1}|c_1\rangle \otimes \cdots \otimes \sum_{c_n=0,1} (-1)^{c_n b_n}|c_n\rangle \\ &= \frac{1}{2^{\frac{n}{2}}} \sum_{c_1 \ldots c_n} (-1)^{\sum_{i=1}^n b_i c_i}|c_1 \ldots c_n\rangle \end{aligned}$$

Ainsi

$$|\Psi_{\text{fin}}\rangle = \sum_{c_1 \ldots c_n} \left\{ \frac{1}{2^n} \sum_{b_1 \ldots b_n} (-1)^{f(b_1 \ldots b_n)} (-1)^{\sum_{i=1}^n b_1 c_i} \right\} |c_1 \ldots c_n\rangle \otimes \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

L'état final est à nouveau une superposition cohérente d'états classiques affectés d'amplitudes

$$\frac{1}{2^n} \sum_{b_1 \ldots b_n} (-1)^{f(b_1 \ldots b_n)} (-1)^{\sum_{i=1}^n b_i c_i}$$

Les amplitudes contiennent de l'information sur la fonction f. Si nous les connaissions toutes nous pourrions en fait déterminer cette fonction. Mais la seule chose qui est à notre disposition est la totalité de l'état $|\Psi_{\text{fin}}\rangle$ (pris dans sa globalité) et *la seule chose que nous puissions faire, pour extraire de l'information, est une mesure.*

6.6.1 Dernière étape de l'algorithme: la mesure

Appliquons le postulat de la mesure. Si nous mesurons l'état des n premiers qubits dans la base computationnelle$\{|c_1 \ldots c_n\rangle, c_i = 0, 1\}$, l'état est projeté (ou réduit) sur *un* des états $|c_1 \ldots c_n\rangle$ avec probabilité (règle de Born ou postulat de la mesure)

$$\begin{aligned} \text{Prob}(c_1 \ldots c_n) &= [\text{carré de l'amplitude devant } |c_1 \ldots c_n\rangle] \\ &= \frac{1}{2^{2n}} \left| \sum_{b_1 \ldots b_n} (-1)^{f(b_1 \ldots b_n)} (-1)^{\sum_{i=1}^n c_i b_i} \right|^2 \end{aligned}$$

La signification de cette assertion est la suivante : tant que la mesure est faite une fois sur un unique état $|\Psi_{\text{fin}}\rangle$ l'état final observé est un des états $|c_1 \ldots c_n\rangle$ et il n'y a aucun moyen de prédire lequel. Si l'on dispose d'un

ensemble d'états $|\Psi_{\text{fin}}\rangle$, en répétant l'expérience plusieurs fois la fréquence des observations $|c_1 \dots c_n\rangle$ est donnée par $\text{Prob}(c_1 \dots c_n)$.

Calculons cette probabilité. Si f est constante on trouve

$$\begin{aligned}
\text{Prob}(c_1 \dots c_n) &= \frac{1}{2^{2n}} \left| \sum_{b_1 \dots b_n} (-1)^{\sum_{i=1}^n c_i b_i} \right|^2 \\
&= \frac{1}{2^{2n}} \left| \sum_{b_1 \dots b_n} \prod_{i=1}^n (-1)^{c_i b_i} \right|^2 \\
&= \frac{1}{2^{2n}} \left| \prod_{i=1}^n \left((-1)^{c_i 0} + (-1)^{c_i 1} \right) \right|^2 \\
&= \begin{cases} 1 & \text{si } (c_1 \dots c_n) = (0 \dots 0) \\ 0 & \text{dans tous les autres cas} \end{cases}
\end{aligned}$$

Donc si f est constante nous observerons certainement $(0 \dots 0)$ (c.a.d avec probabilité 1) en faisant une seule expérience ! Par contre si f est balancée on constate que

$$\text{Prob}(0 \dots 0) = \frac{1}{2^{2n}} \left| \sum_{b_1 \dots b_n} (-1)^{f(b_1 \dots b_n)} \right|^2 = 0$$

et on n'observera certainement pas $(0 \dots 0)$.

En conclusion : après le processus de mesure si $(0 \dots 0)$ est observé on peut conclure "f constante" et si autre chose est observé on peut conclure "f balancée". Remarquablement, il suffit de faire l'expérience et d'utiliser l'Oracle quantique une seule fois ! Ceci n'est pas typique des autres algorithmes quantiques: nous verrons par exemple que pour l'algorithme de Shor il faut faire l'expérience plusieurs fois pour amplifier la probabilité de succès.

Note sur la complexité de l'algorithme. En general nous devons distinguer la complexité du circuit et celle de l'algorithme proprement dit. Dans ce cours la complexité des circuits sera mesurée en terme de deux grandeurs, la "profondeur" et la "largeur". La taille du circuit est alors définie comme le produit $(profondeur) \times (largeur)$. Ici le circuit contient 3 tranches de temps intermédiaires: on dit que ce circuit à une profondeur égale à 3 (ce qui est important ici c'est qu'elle est $O(1)$). Si le temps élémentaire requis pour effectuer une porte quantique (par RMN par ex) est de τ alors le temps de calcul du circuit est de 3τ. La largeur du circuit est donnée par le nombre de bits d'entrée plus le nombre de bits auxiliaires, ici $n+1$, et représente le nombre de calculs que le circuit quantique effectue à chaque tranche temporelle.

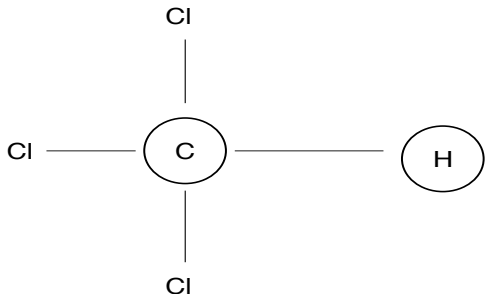


Figure 6.4: L'algorithme DJ a été réalisé par Résonance Magnétique Nucléaire sur les molécules de $CHCl_3$. On agit sur deux qubits associés aux spins nucléaires des atomes H et C entourés

La taille du circuit de DJ est donc $3(n+1) = O(n)$. Finalement quelle est la complexité de l'algorithme lui même ? Puisqu'on peut résoudre le problème de DJ en posant une seule question à l'Oracle, l'algorithme possède un temps de calcul $O(1)$ avec un circuit de taille $O(n)$.

6.7 Quelques remarques sur les réalisations expérimentales

Nous reviendrons sur les réalisations expérimentales à la fin du cours. Ce paragraphe peut être omis en première lecture.

L'algorithme de DJ fut un des premiers algorithmes quantique a être réalisé experimentalement (2001) pour le cas $n = 1$ par la résonance magnétique nucléaire. Cette experience utilise un liquide de $CHCl_3$ (le chloroforme, fig. 6.4). Pour $n = 1$ il faut deux bits quantiques, un pour l'entrée et un bit auxiliare pour stocker le résultat de l'Oracle. Ceux-ci sont matérialisés par les spins nucléaires de l'atome d'hydrogène H et de carbone C. Les portes de Hadamard et l'Oracle peuvent être réalisés en manipulant ces deux spins nucléaires par des champs magnétiques de radiofréquence. Un des problèmes principaux est de préparer toutes les molécules du liquide dans l'état initial $|00\rangle$. En effet on ne peut pas se débarasser complètement des fluctuations thermiques qui induisent des transitions vers les autres états pour une fraction des molécules du liquide. Pour augmenter n il faut prendre de plus grosses molécules avec des atomes appropriés. Ceci pose un problème ("scalability problem") parceque plus la molecule est grosse plus les niveaux d'énergie sont nombreux et rapprochés (le spectre devient continu en quelque sorte) et il devient de plus en plus difficile de manipuler sélectivement les bits quantiques grâce à des transitions de radiofréquence. Plus récemment, l'agorithme

de DJ a aussi été réalisé grâce aux technologies des pièges à ions et des cavités resonantes (cavity QED). Toutes ces experiences sont limitées à un faible nombre de qubits (< 10 qubits).