Les algorithmes quantiques

La logique quantique permet en principe de réaliser certains calculs de façon plus rapide que ne le font les ordinateurs classiques. Nous en donnons ici trois exemples, correspondant à la résolution de problèmes posés sous forme logique simple.

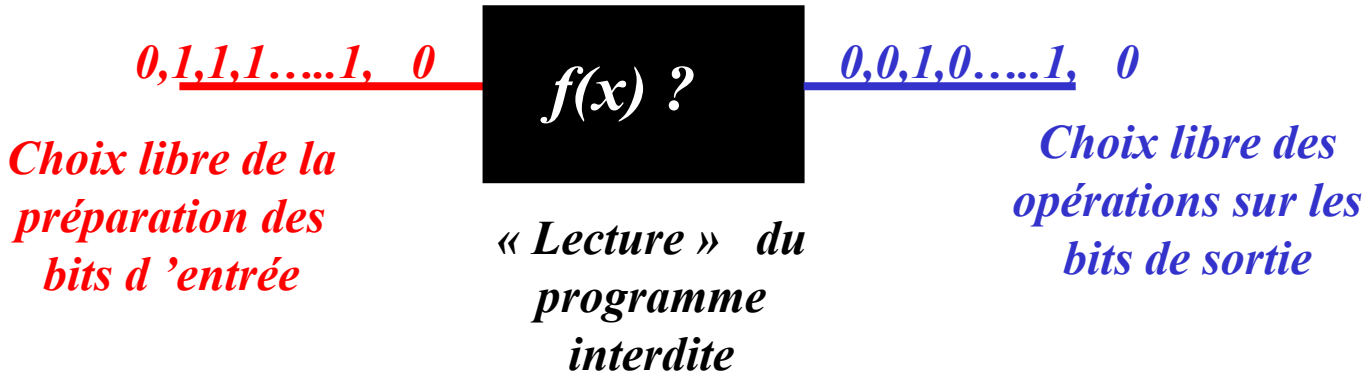
De façon générale, on classe les problèmes de calcul en distinguant ceux qui sont «faciles», dont la résolution demande un temps qui croît de façon polynomiale avec le nombre n de bits impliqués et ceux qui sont «difficiles», pour lesquels il faut un temps augmentant de façon exponentielle avec n. Les problèmes «difficiles» deviennent pratiquement insolubles pour n assez grand. Un exemple en est donné par la factorisation des grands nombres pour laquelle seuls des algorithmes classiques «difficiles» existent (sans qu'il aît été prouvé qu'un algorithme classique facile n'existait pas…).

La logique quantique, exploitant les superpositions de qubits et leur intrication permet de transformer certains problèmes classiquement difficiles en problèmes quantiquement faciles. L'algorithme de Shor décrit par exemple un procédé de factorisation demandant un temps croissant de façon polynomiale avec le nombre de bits du nombre à factoriser. Nous n'aborderons pas dans cette introduction la description de cet algorithme, mais nous en analyserons d'autres, mathématiquement plus simples, qui illustrent bien les avantages de la logique quantique pour certains types de calcul.

Les problèmes posés sous forme d' oracle

Les problèmes logiques que nous allons considérer ici sont posés sous forme d ' «oracle». On suppose qu' une machine programmée selon des règles inconnues (décrite comme une «boîte noire» ou oracle), calcule une fonction dont nous ne connaissons que certaines caractéristiques. Le problème consiste à déterminer une propriété inconnue de la fonction, sans «ouvrir» la boite. Nous pouvons interroger l 'oracle en entrant des données dans la boite et en manipulant sa sortie, sans l'ouvrir pour en analyser le contenu. Le problème sera «facile» si sa résolution demande un nombre total d'opérations croissant de façon polynomiale avec le nombre de bits, «difficile» s'il croit de façon exponentielle avec ce nombre.

Nous allons montrer que le passage du calcul classique au calcul quantique transforme certains oracles classiques difficiles en oracles quantiques faciles. Dans d' autres cas, le problème quantique reste difficile, mais moins que le problème classique (croissance toujours exponentielle du nombre d 'opérations, mais avec un exposant plus petit que classiquement).



Exemples d 'oracles classiquement «difficiles»

L'oracle de Deutsch-Josza

f(x) constante ou balancée ?

f(x) est une fonction booléenne de [0, 2^n -1] dans [0, 1]. On sait qu'elle est soit constante, soit balancée. Est-elle l 'un ou l 'autre ?

Classiquement, il faut « interroger » l 'oracle $2^{n-1} + 1$ fois pour répondre à la question à coup sûr (il faut introduire $2^{n-1} + 1$ valeurs différentes de x et calculer f(x) à chaque fois) → Croissance exponentielle avec n du nombre d 'opérations et problème classique « difficile »

L'oracle de Grover

$f(x) = \delta(x-x_0)$

x_0?

f(x) est une fonction booléenne de [0, 2^n -1] dans [0, 1] qui n'est non nulle que pour $x = x_0$. Trouver x_0.

Equivaut à la recherche «inversée» d'un abonné dans un annuaire à partir de son numéro connu a. Les x sont les abonnés, f(x) vaut 1 si a est le numéro de x, 0 sinon.

Classiquement, il faut calculer f(x) («consulter» l 'annuaire) $N = 2^n - 1$ fois pour trouver à coup sûr. Problème classique difficile.

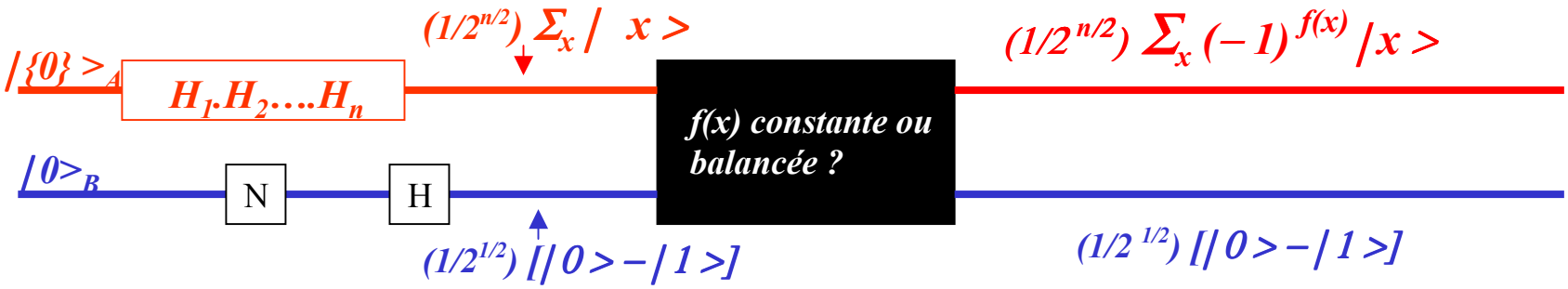
L'oracle de Simon

« Période » de f(x) ?

f(x) est une fonction de [0, 2^n -1] dans [0, 2^n -1] telle que f (x') = f (x) ssi $x = x \oplus s$ où s est une suite inconnue à n termes de « 0 » et « 1 » et $\oplus$ représente l 'addition « bit à bit » (s : «période» de f). Déterminer s

Classiquement, il faut calculer f (x) pour des valeurs aléatoires de x jusqu 'à trouver deux x et x' tels que f (x) = f (x'). Alors $x \oplus s = x'$ et $x \oplus x' = x \oplus x \oplus s = s$ (car $x \oplus x = \{0\}$). Il faut $2^{n-1} + 1$ opérations pour trouver la réponse à coup sûr → Problème classique « difficile ».

L 'Algorithme quantique de Deutsch-Josza



Le registre d'entrée A (n qubits) est préparé (par application de la transformation de Hadamard H sur chaque qubit) dans la superposition symétrique des 2^n états $|x>$ possibles.

Le registre de sortie B (1 qubit) est inversé par N, puis préparé par H dans $(1/2^{1/2})\ [|0>-|1>]$.

Action de l 'oracle: $|x>|0> \rightarrow |x>|f(x)>$ et $|x>|1> \rightarrow |x>|1 \oplus f(x)>$

Si $f(x)=0$: $|x>\ [|0>-|1>] \rightarrow |x>\ [|0>-|1>]$

Si $f(x)=1$: $|x>\ [|0>-|1>] \rightarrow -|x>\ [|0>-|1>]$

$\Big\}\ (-1)^{f(x)}\ |x>\ [|0>-|1>]$

Et par superposition:

$$(1/2^{(n+1)/2})\ \Sigma_x\ |x>\ [|0>-|1>] \rightarrow (1/2^{(n+1)/2})\ (\Sigma_x\ (-1)^{f(x)}\ |x>)\ [|0>-|1>]$$

Les registres restent non intriqués après l'oracle. Déphasage des amplitudes dans le registre A en $(-1)^{f(x)}$.

Pour résoudre le problème, il s'agit de décider entre deux possibilités:

Si f(x) est constante: f (x) = 0 ∀x ou f (x) =1 ∀x →

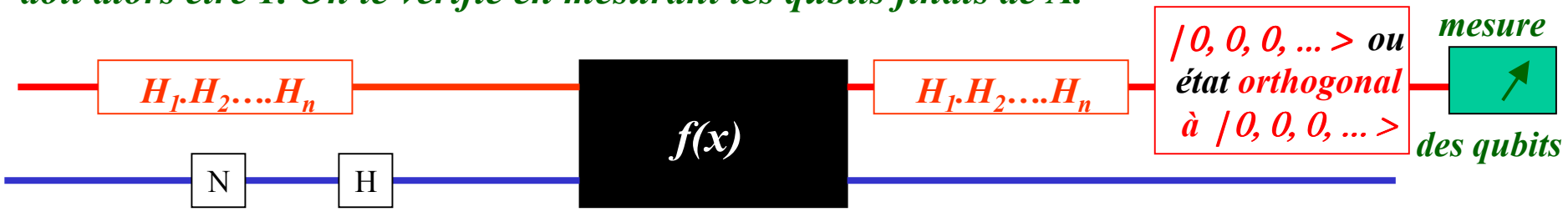
$(1/2^{n/2}) \sum_x (-1)^{f(x)} |x> = \pm (1/2^{n/2}) \sum_x |x>$ → *registre A inchangé (au signe près)*

Si f(x) est balancée: autant de f (x) = 0 que de f(x) = 1 →

Autant d 'amplitudes +1 que d 'amplitudes –1 dans la superposition finale du registre A

→ $\sum_x (-1)^{f(x)} |x>$ *orthogonal à* $\sum_x |x>$

Résoudre l 'oracle revient à distinguer deux états orthogonaux de l'état final du registre A: On applique à nouveau H à tous les qubits. Comme $H^2=1$, *on retrouve l'état initial* $|\{0\}>$ *si f(x) est constante, un état orthogonal si f(x) est balancée → au moins un des qubits doit alors être 1. On le vérifie en mesurant les qubits finals de A.*



La réponse nécessite au plus 3n+2 opérations à un qubit (2n + 1 opérations H, une opération de bascule (N) et au plus mesure de n qubits (on peut arrêter dès qu'on trouve un 1) → problème quantiquement «facile».

Remarques sur l 'algorithme de Deutsch-Josza

1. Où est l 'intrication?

Les qubits de A ne sont pas intriqués à B qui reste inchangé. De l 'intrication est en général cependant créée entre les qubits de A:

Exemple. Cas d 'une fonction balancée agissant sur un registre A de trois qubits:

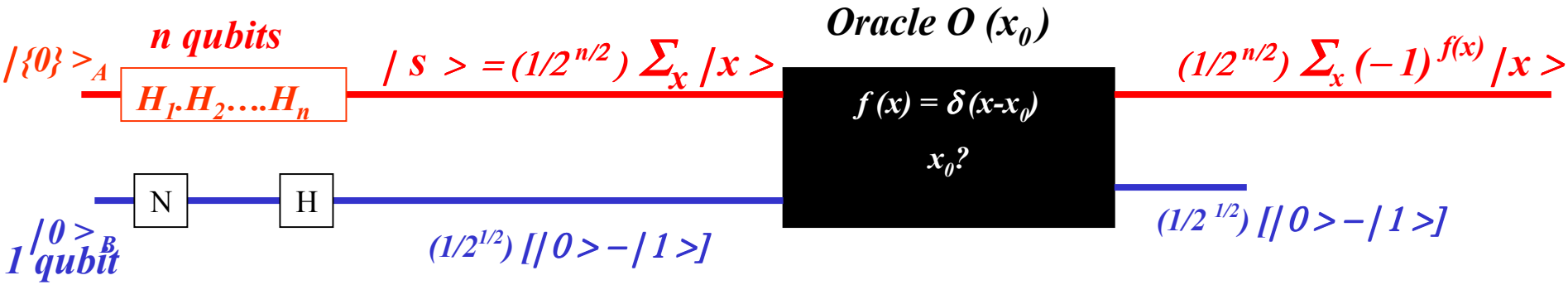
$$\Sigma_x(-1)^{f(x)}|x\rangle = |000\rangle - |001\rangle + |010\rangle - |011\rangle + |100\rangle - |101\rangle - |110\rangle + |111\rangle$$
$$= |0\rangle_1 [|00\rangle - |01\rangle + |10\rangle - |11\rangle]_{23} + |1\rangle_1 [|00\rangle - |01\rangle - |10\rangle + |11\rangle]_{23}$$
$$= |0\rangle_1 |\Psi\rangle_{23} + |1\rangle_1 |\Phi\rangle_{23} \quad \text{avec} \quad {}_{23}\langle\Phi|\Psi\rangle_{23} = 0$$

Cette décomposition de Schmidt montre que le qubit 1 et l 'ensemble des qubits 2 et 3 sont maximalement intriqués.

2. Cet algorithme est-il vraiment avantageux par rapport à la procédure classique?

L' avantage de l' algorithme quantique n'existe que si on cherche une réponse certaine. Si on s' autorise une probabilité finie ε d' erreur, aussi petite soit-elle, l' algorithme classique (calcul successif de f (x) pour des valeurs de x tirées au hasard) donne un résultat acceptable au bout de $k \cong -\log_2(\varepsilon)$ opérations (nombre indépendant de n). Le problème classique devient donc «facile» dès qu 'on accepte un taux fini d 'erreur. Ceci diminue considérablement l 'intérêt de l 'algorithme quantique puisqu' il faut être sûr de pouvoir l 'effectuer sans aucune décohérence pour qu 'il soit avantageux par rapport à la version classique.

Algorithme de recherche quantique (Grover)



Le problème commence comme celui de Deutsch-Josza. La fonction Booléenne de [0, $2^n - 1$] dans [0, 1] est réalisée par un oracle agissant sur la superposition symétrique des $N = 2^n$ qubits de A et sur le qubit B préparé par les opérations N et H dans l 'état $(1/2^{1/2})\ [|0> - |1>]$.
Revient à inverser l 'amplitude de l 'élément $x = x_0$ dans la superposition des états de la base {x} sans toucher aux autres. Comment détecter l 'élément dont l 'amplitude a été inversée?

On applique au registre A, après l'opération de l' Oracle $O(x_0)$, l 'opérateur unitaire $U_s = 2\,|s><s| - 1$ où $|s><s|$ est le projecteur sur la superposition symétrique de tous les états de la base x de A ($|s> = (1/\sqrt{N})\ \Sigma_x\ |x>$).

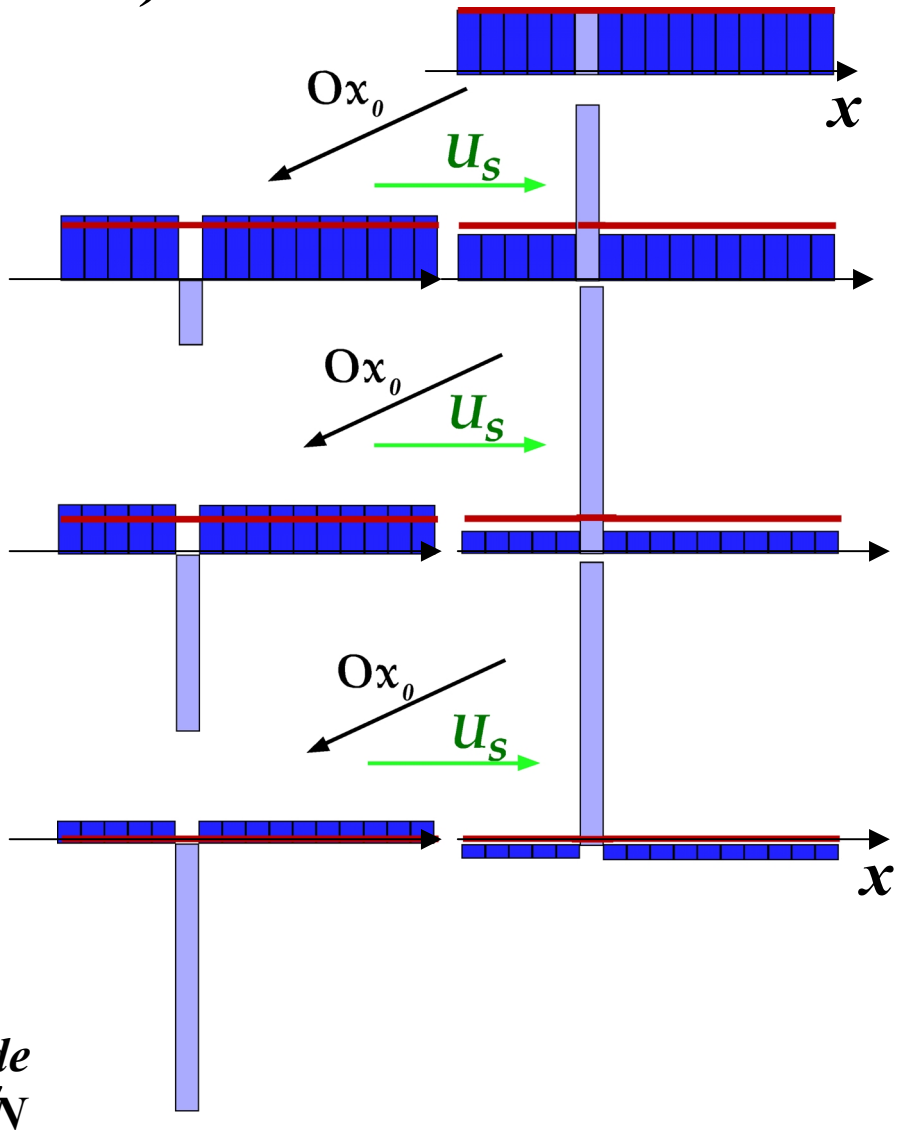
L 'action de U_s sur une superposition quelconque $|\psi> = \Sigma_x\ a_x\,|x>$ revient à symétriser les amplitudes de $|\psi>$ par rapport à leur moyenne. De la relation $<s|\psi> = (1/\sqrt{N})\ \Sigma_x\ a_x = \sqrt{N}\,<a>$ où $<a>$ est la « moyenne des amplitudes » de l 'état $|\psi>$, on déduit en effet:

$$U_s\,|\psi> = 2\,|s><s|\psi> - |\psi> = \Sigma_x\,(2<a> - a_x)\,|x> = \Sigma_x\ b_x\,|x>$$

avec: $(a_x + b_x)/2 = <a>$

Illustration de l 'algorithme de recherche quantique dans le cas n = 4 (N = 16).

On applique l 'oracle $O(x_O)$ sur $| s >$, puis U_s , et on itère les deux opérations. L 'effet de $O(x_O)$ est d 'abaisser à chaque fois la valeur moyenne $< a >$ (ligne horizontale rouge). L 'opération de symétrie par rapport à cette moyenne (U_s) réduit le « fond » et fait ressortir la valeur marquée par l 'oracle. Au bout de trois itérations, le fond devient très petit. On mesure alors le registre A (4 qbits) et on trouve avec une probabilité de 96% la valeur $x_0 = 5$ choisie par l 'oracle. Il faut savoir quand arrêter la procédure (si on va trop loin, le fond augmente en valeur absolue en devenant négatif).

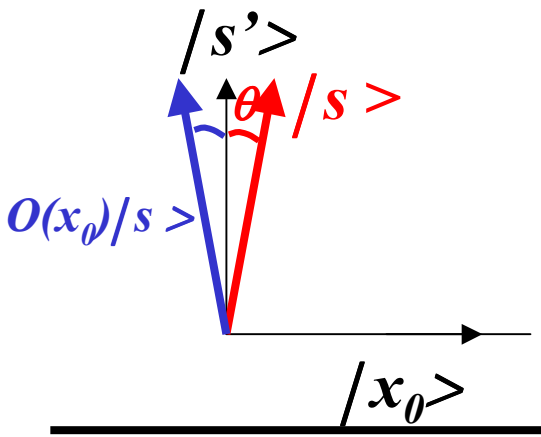


Nous allons montrer que l' itération demande un nombre d'opérations croissant comme $\sqrt{N}$

Interprétation géométrique de l' algorithme de recherche quantique

Décomposons la superposition symétrique de tous les états du registre A en séparant $|x_0\rangle$ des autres termes regroupés dans $|s'\rangle$, vecteur normé:

$$|s\rangle = (1/\sqrt{N})\ \Sigma_x\ |x\rangle = \sqrt{\frac{N-1}{N}}\,|s'\rangle + \frac{1}{\sqrt{N}}\,|x_0\rangle \quad \textit{avec} \quad |s'\rangle = \left(\frac{1}{\sqrt{N-1}}\right)\sum_{x\neq x_0}|x\rangle$$



Introduisant une représentation géométrique, on peut décrire ces états comme des vecteurs d'un plan sous tendu par deux vecteurs orthonormés correspondant à $|x_0\rangle$ et $|s'\rangle$. L 'état initial $|s\rangle$ de A a une petite composante le long de $|x_0\rangle$ et une grande composante le long de $|s'\rangle$. L'angle θ entre $|s\rangle$ et $|s'\rangle$ vérifie $\sin(\theta) = 1/\sqrt{N}$. L 'action unitaire de l 'oracle $O(x_0)$ change $|s\rangle$ en $O(x_0)\,|s\rangle$, son symétrique par rapport à $|s'\rangle$.

L 'opération unitaire $U_s = 2\,|s\rangle\langle s| - I$ appliquée ensuite laisse la composante de $O(x_0)\,|s\rangle$ le long de $|s\rangle$ inchangée et change le signe de sa composante normale à $|s\rangle$: elle équivaut à une symétrie par rapport à la direction de $|s\rangle$. $U_s \otimes O(x_0)$ est donc représentée par une rotation d 'angle 2θ (produit de symétries par rapport à deux directions faisant l 'angle θ), qui rapproche la direction du vecteur associé au registre A de celle de $|x_0\rangle$.

Chaque application de $U_s \otimes O(x_0)$ fait tourner le vecteur de $2\theta \approx 2/\sqrt{N}$. Il faut donc T itérations telles que $(2T+1)/\sqrt{N} = \pi/2$ $\rightarrow$ $\boxed{T \approx (\pi/4)\sqrt{N}}$

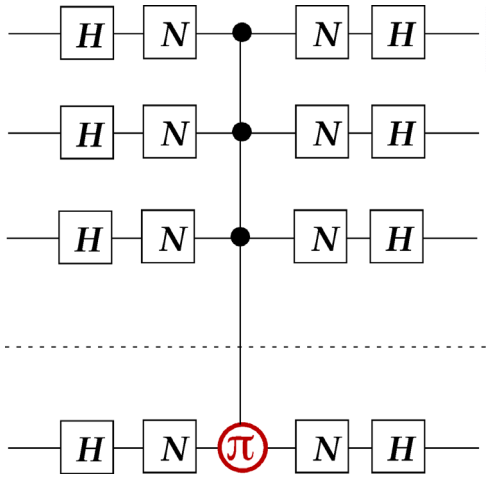
Réalisation de l'opération de symétrie $U_s = 2\,|s><s|\,-1$

Il reste à montrer comment on réalise l 'opération unitaire U_s*. Pour que l 'algorithme quantique soit intéressant,* U_s *doit demander un nombre d'opérations élémentaires ne croissant pas de façon exponentielle avec n , donc n'augmentant pas de façon linéaire avec N… Pour construire* U_s *, on remarque d' abord l 'identité:*

$$U_s = 2\,|s><s| - 1 = H_1.H_2....H_n\,[\,2\,|\{0\}><\{0\}| - 1\,]\,H_1.H_2....H_n \ ,$$

puis le fait que l'opérateur $[\,2\,|\{0\}><\{0\}| - 1\,]$ *laisse l 'état* $|\{0\}>$ *invariant et déphase de* π *les* $2^n - 1$ *autres états de la base* $|\{x\}>$*. Cet opérateur est donc égal à* (-1) *fois l 'opérateur qui déphase de* π *l 'état* $|\{0\}>$*, laissant tous les autres états invariants. Ce dernier opérateur est à son tour identique à celui qui déphase de* π *le seul état* $|1, 1, 1, 1... 1>$*, conjugué à droite et à gauche par des portes N (qui transforment* $|\{0\}>$ *en* $|1, 1, 1, 1...1>$ *et inversement).*

On en conclut finalement que U_s *est réalisé (au signe près) par le circuit schématisé par :*



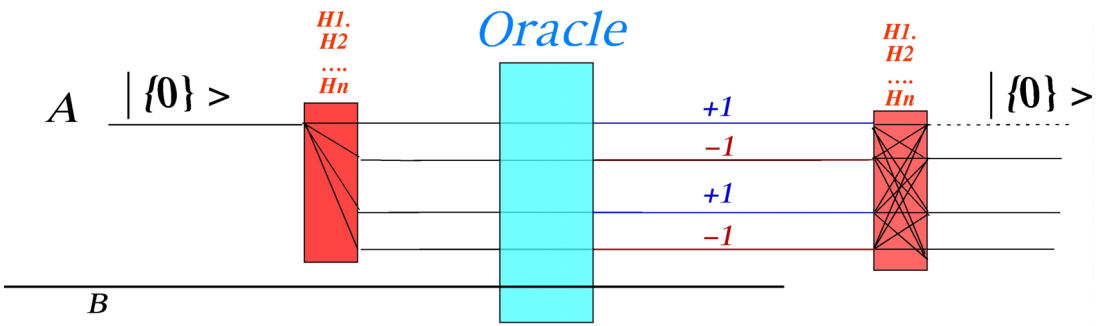
La porte de phase-π *est réalisée par* σ_z *(ou encore par une porte CNOT (*σ_x*) conjuguée par deux Hadamard). Le circuit quantique ci-contre à n – 1 bits source et un bit cible peut se décomposer en portes à un et deux qubits en nombre proportionnel à* $n = \log_2 N$*.*

Finalement, l 'algorithme de recherche de Grover demande donc un nombre d 'opérations en $\sqrt{N}\log_2 N$ ***, inférieur à celui exigé classiquement (en N).***

Rôle des interférences dans les algorithmes de Deutsch et de Grover.

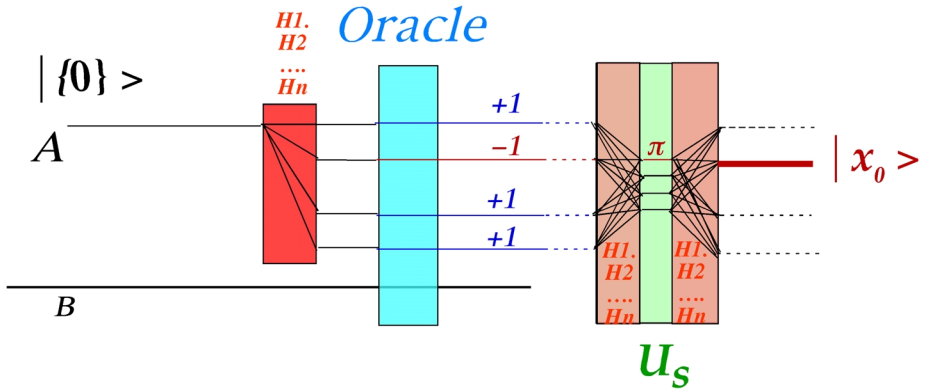
Deutsch

explicité pour n=2, cas où f(x) est balancée



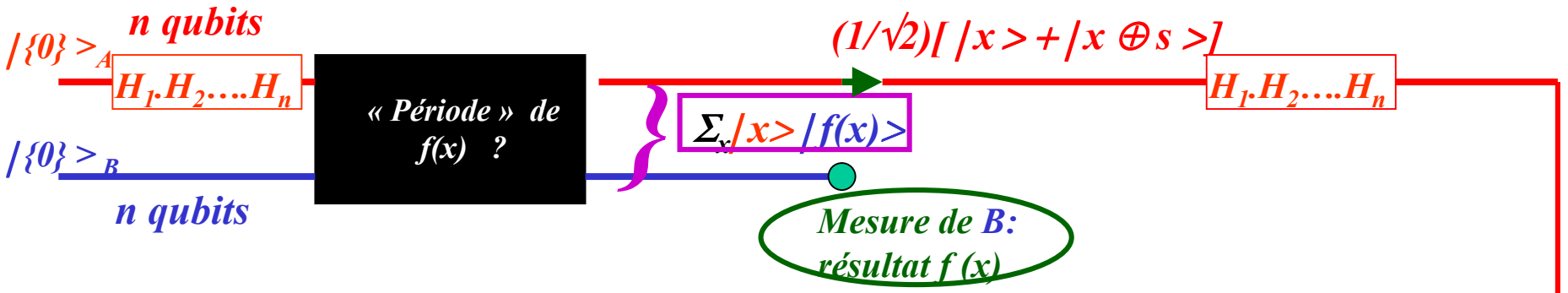
Grover

explicité pour n = 2 (N = 4); l 'état marqué est '1' = (0, 1). Une seule itération suffit dans ce cas (θ = Arcsin (1/2) = π /6).



Les algorithmes de Deutsch et de Grover peuvent se décrire comme des processus d' interférence quantique à 2^n chemins : Les $H_1.H_2....H_n$ jouent le rôle de « lames séparatrices » qui décomposent et recombinent les états du registre A. L 'oracle O est l 'analogue d 'une lame déphasante qui modifie les phases relatives des états. Dans Grover, chaque symétrie U_s combine «en sandwich» deux lames séparatrices avec une lame déphasante. Dans les deux «interféromètres», l'effet recherché est de favoriser dans l'état final un état (analogue d' une «frange brillante») dont la mesure finale fournit la réponse voulue.

L' Algorithme de Simon



On réalise la suite d'opérations schématisée ci-dessus: calcul parallèle de toutes les valeurs de la fonction suivie d'une mesure du registre B projetant A dans une superposition de deux états qui diffèrent bit à bit de la quantité inconnue s. On applique alors à nouveau les transformations de Hadamard aux n qubits de A: elles font évoluer chaque qubit suivant la loi: $|0> \rightarrow (1/\sqrt{2})\,[\,|0> + |1>\,]$ *et* $|1> \rightarrow (1/\sqrt{2})\,[\,|0> - |1>\,]$. *Un état* $|\{x\}>$ *(produit de n états* $|x_i>$ *avec* $x_i = 0$ *ou 1) devient une superposition de produits d'états* $|y_i>$ *(avec* $y_i = 0$ *ou 1). Les coefficients de cette superposition valent + 1 ou − 1 suivant la parité de la somme* $\Sigma_i x_i y_i$ *:*

$$|\{x\}> = |x_1, x_2, x_3, ...x_n> \rightarrow = (1/2^{(n+1)/2})\,\Sigma_{\{y\}}\ (-1)^{\{\Sigma_i x_i y_i\}}\,|y_1, y_2, y_3, ...y_n>$$

L'état final de A est donc:

$$(1/2^{(n+1)/2})\,\Sigma_{\{y\}}\,[\,(-1)^{\{\Sigma_i x_i y_i\}} + (-1)^{\{\Sigma_i (x_i \oplus s_i) y_i\}}]\,|y_1, y_2, y_3, ...y_n>$$

$$= \boxed{(1/2^{(n+1)/2})\,\Sigma_{\{y\}}\,(-1)^{\{\Sigma_i x_i y_i\}}\,[\ 1 + (-1)^{\{\Sigma_i s_i y_i\}}]\,|y_1, y_2, y_3, ...y_n>}$$

Une mesure répétée ~ n fois de A va alors nous permettre de déterminer s

Détermination de la «période» inconnue s

$$|\Psi(final)\rangle_A = (1/2^{(n+1)/2}) \sum_{\{y\}} (-1)^{\{\sum_i x_i y_i\}} [\ 1 + (-1)^{\{\sum_i s_i y_i\}}] \ |y_1, y_2, y_3, \ldots y_n\rangle$$

Amplitude non nulle ssi $\sum_i s_i y_i = 0$ *(modulo 2)*

Une mesure des qubits individuels donne une suite $y_{1a}, y_{2a}, y_{3a}, \ldots y_{na}$ ***de valeurs 0 et 1 qui satisfait la condition:***

$$\sum_i s_i\ y_{ia} = 0 \text{ (modulo 2).}$$

On recommence n fois l 'opération et on obtient ainsi, en général, n relations indépendantes (si par hasard deux mesures donnent le même vecteur, on recommence une fois de plus):

$$\sum_i s_i\ y_{ia} = 0$$

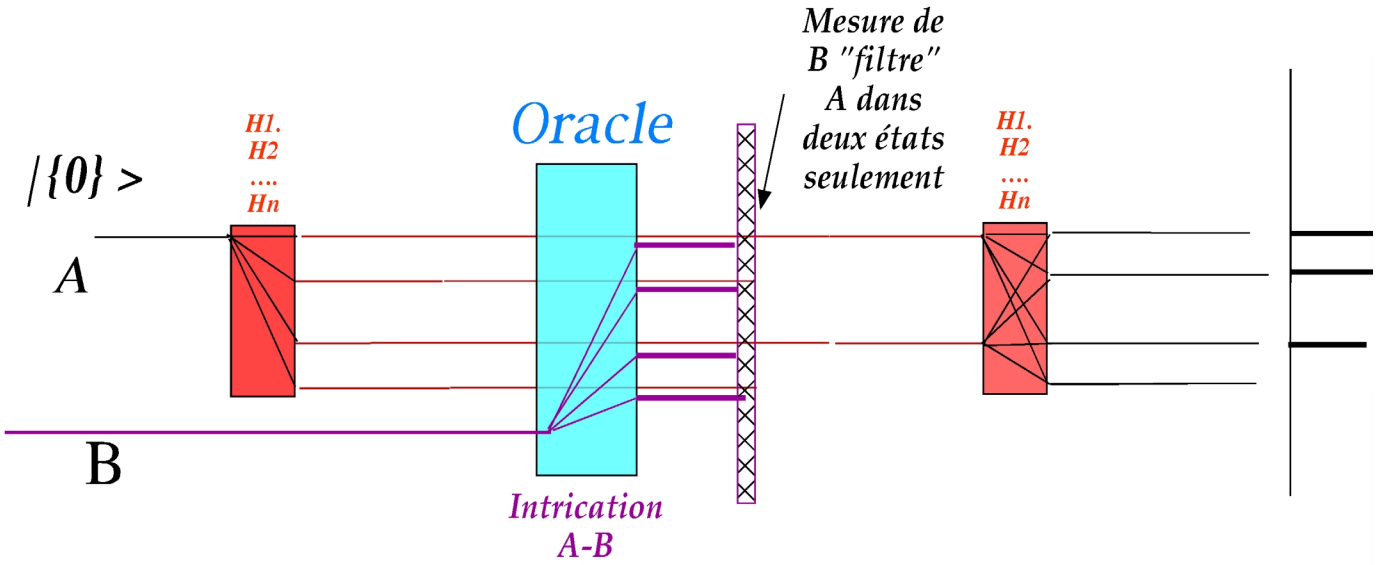
$$\sum_i s_i\ y_{ib} = 0$$

$$\ldots\ldots\ldots$$

$$\sum_i s_i\ y_{in} = 0$$

La résolution de ce système d 'équations donne s. Le processus requiert $\cong 4n^2$ ***opérations. Le problème est donc quantiquement facile. De plus, il tolère des erreurs puisqu 'on peut toujours vérifier le résultat en comparant f (x) et f (x ⊕ s) une fois s obtenu.***

Rôle de l' intrication et de la mesure dans l 'algorithme de Simon



Dans l' algorithme de Simon, l' intrication et la mesure projective jouent un rôle plus essentiel que dans ceux de Deutsch et Grover. L'oracle intrique les registres A et B, puis la mesure de B projette A dans une superposition de deux états seulement. Après mélange par la « lame séparatrice », la signature du signal d 'interférence final nous renseigne sur la séparation de ces deux états, donc sur la période cherchée. Quoique mathématiquement plus compliqué, l' algorithme de Shor, basé sur la recherche de la période d 'une fonction, ressemble beaucoup dans son principe à celui de Simon.

<u>Remarque:</u> il n' est même pas besoin de «lire» la mesure du registre B. Il suffit d' avoir intriqué B à son appareil de mesure , ce qui réduit A à un mélange statistique de superpositions $|x> + |x \oplus s>$. Leur recombinaison finale par $H_1.H_2 \ldots H_n$ ne conduit, quel que soit x, à une interférence constructive que pour les états $|y_1, y_2, y_3, \ldots .y_n>$ satisfaisant les équations linéaires de la page précédente (on peut réduire le nombre d'opérations à $3n^2$).