9. Introduction au Calcul Quantique

Nous abordons dans cette dernière partie un aspect essentiel du traitement quantique de l 'information, le calcul quantique. C' est en effet l' espoir d 'arriver à mettre en œuvre de façon pratique la logique quantique pour des calculs « utiles » impossibles à faire classiquement (comme la factorisation des grands nombres) qui a été un moteur puissant dans le développement du domaine de physique que nous étudions dans le cours de cette année.

Nous discuterons ici des questions de principe du calcul quantique , dans une situation idéale. Comme nous l 'avons vu dans les leçons précédentes, la décohérence contribue de façon essentielle à la disparition des superpositions d 'états et de l 'intrication dans les systèmes de grande « dimension ». Elle sera donc la grande ennemie du calcul quantique dans la mesure où elle tend à restaurer très vite une logique classique dans les systèmes formés d 'un grand nombre de qubits (en pratique dès quelques qubits). Différentes stratégies plus ou moins réalistes ont été proposées pour lutter contre la décohérence. Certaines ont été abordées dans des séminaires de ce cours (Luiz Davidovich, Peter Knight). Nous n 'en discuterons pas plus cette année et nous nous limiterons à une analyse des principes du calcul quantique dans une situation idéale et simplificatrice d 'un système complètement découplé de l 'environnement.

Après avoir présenté des idées générales sur le calcul quantique, nous décrirons les portes logiques élémentaires qui permettent, par combinaison, d 'effectuer en principe n 'importe quelle opération. Nous conclurons par la description de trois algorithmes de calcul quantique qui illustrent bien les «avantages» de principe de la logique quantique.

Généralités sur le calcul quantique

Il s 'agit d' utiliser des processus quantiques pour réaliser des opérations logiques. Les unités de calcul sont des qubits évoluant dans des superpositions d 'états 0 et 1.

Les interférences quantiques et l 'intrication entre les qubits jouent un rôle essentiel.

Un système de N-qubits évolue dans un espace de Hilbert H (N) à 2^N dimensions. La base naturelle d'états est l'ensemble des produits tensoriels des états de chaque qubit $|x_1, x_2, x_3, \ldots, x_i, \ldots, x_N\rangle$ où $x_i = 0$ ou 1. On notera cette base $|\{x\}\rangle$ ou $|x\rangle$ pour abréger. On appellera $|\{0\}\rangle$ l 'état où tous les qubits sont initialisés dans leur état 0. Les états de base correspondent à l 'ensemble des registres classiques, suites de 0 et de 1. Le symbole x est associé à l 'un des 2^N entiers 0, 1,……$2^N - 1$ dont il représente l 'écriture en base 2.

Un « ordinateur quantique » est un dispositif idéal qui permet de réaliser des opérations unitaires U dans l 'espace H(N), agissant sur les qubits suivant les lois quantiques. L' unitarité des opérations assure leur réversibilité. En appelant $|\psi\rangle$ un état quelconque de H(N), on a:

$$|\psi\rangle_f = U\,|\psi\rangle_i \rightarrow |\psi\rangle_i = U^{\dagger}\,|\psi\rangle_f .$$

Une telle machine peut faire tout ce que fait un ordinateur classique, et bien plus de choses en exploitant la logique quantique….

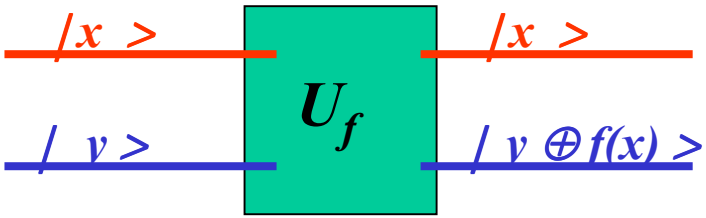
Principe du calcul d 'une fonction

On peut montrer que toute opération unitaire sur un ensemble de qubits peut se réduire à un produit d 'opérations de portes logiques impliquant un ou deux qubits à la fois. Nous indiquerons plus loin les lignes générales de cette démonstration (voir par exemple Nielsen et Chuang pour plus de détails). Toute opération unitaire peut donc être programmée en construisant des portes logiques à un ou deux qubits et en les combinant suivant une architecture convenable.

Il en résulte que l 'on peut calculer de façon réversible n'importe quelle fonction f (x) de l 'ensemble des entiers (0, 2^n -1) dans l 'ensemble des entiers (0, 2^p -1). On procède de la façon (unitaire) suivante: On prépare un registre « d'entrée » A (espace H(n)) dans un état $|x>$ et un registre « de sortie » B (espace H(p)), dans un état $|y>$ et on applique à $|x>|y>$ la transformation:

$$|x>|y> \rightarrow |x>|y \oplus f(x)>$$

Importance de la conservation des bits d 'entrée: assure l 'unitarité et donc la réversibilité



$\oplus$ représente l 'addition modulo 2^p . Il s'agit d 'une transformation unitaire (qui conserve l 'orthogonalité des vecteurs de base dans H(n)$\otimes$H(p)). Elle est donc en principe réalisable par un jeu de portes à un et deux qubits (voir exemples plus loin). Pour déterminer f (x_a) il suffit de préparer les registres initiaux dans l 'état $|x_a>|\{0\}>$ et de mesurer directement l 'état final $|f(x_a)>$ du registre de sortie.

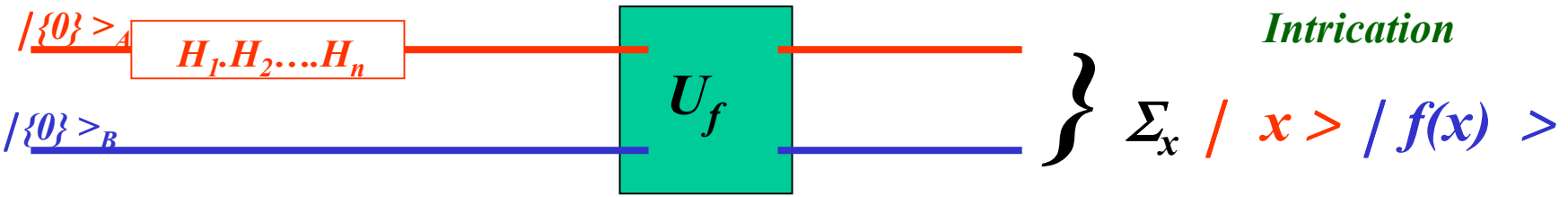
Principe du calcul quantique parallèle

Le principe de superposition permet d 'effectuer le calcul de toutes les valeurs d 'une fonction à la fois:

1. On prépare (grâce à l 'application de la transformation de Hadamard H à chaque qubit) le registre d 'entrée dans l 'état superposition de toutes les valeurs possibles:

$$H_1.H_2....H_n\,|\{0\}>_A = (1/2^{n/2})\,[|0>+|1>]_1 \otimes [|0>+|1>]_2 \;....\otimes [|0>+|1>]_n = (1/2^{n/2})\,\Sigma_x\,|\;x>$$

2. On applique ensuite U_f à $[\,H_1.H_2....H_n\,|\{0\}>_A\,]\otimes|\{0\}>_B$:



Une mesure directe du registre A donne une valeur aléatoire et projette B dans l 'état corrélé: on détermine ainsi f (x) pour une seule valeur de l 'argument à la fois. On peut cependant exploiter le parallélisme quantique en effectuant des manipulations avant la détection de façon à obtenir une information intéressante plus rapidement que par un calcul classique (voir exemples d 'algorithmes plus loin).

Lien avec la théorie de la mesure

A chaque fonction f (x) de A → B on peut associer l 'opérateur hermitique (observable) de A:

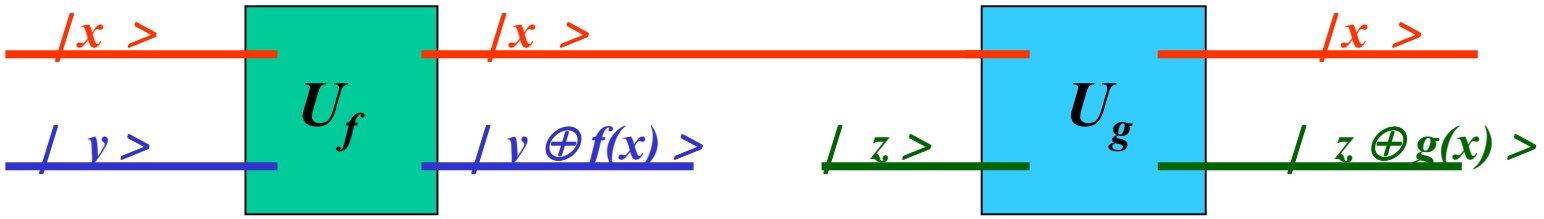
$$O_f = \Sigma_x \, | \, x > \; f(x) \; < x \; |$$

dont les v.p sont des réels ∈ [0 , 2^p -1]. L 'opération de calcul quantique de f(x) équivaut à une pré-mesure de l 'observable O_f dans laquelle le registre B joue le rôle de « mètre ».
L 'intrication associée au calcul quantique parallèle est celle que nous avons déjà analysée dans notre étude de la mesure idéale. Pour en tirer avantage, il faut effectuer toutes les opérations avant que la décohérence ne soit intervenue….

Remarques:

(a) un cas particulier très simple de mesure: A et B sont des registres à un qubit chacun. La porte C- NOT réalise dans ce cas le calcul de la fonction f (0) = 0; f (1) = 1.

(b) toutes les fonctions f (x) de A dans B correspondent à des opérateurs O_f qui commutent et sont donc simultanément calculables. Ceci est évident si on remarque que le registre d'entrée, inchangé, peut être utilisé plusieurs fois pour calculer des fonctions différentes.



Les portes logiques

a. Portes à un qubit (opérations unitaires sur un seul bit):

$U = 1$ — $|\phi\rangle$ $\quad$ $|\phi\rangle$

Canal quantique laisse le bit inchangé

$U = \sigma_x$ — *N*

Bascule le bit

$U = H = \left(\frac{1}{\sqrt{2}}\right)\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ — *H*

Transforme le bit en superposition à poids égaux (Hadamard)

$U_\varphi = \begin{pmatrix} 1 & \\ & e^{i\varphi} \end{pmatrix}$ — φ

Déphase les deux états du qubit l 'un par rapport à l 'autre

Remarques:

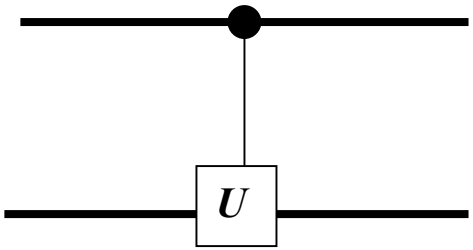
$H = (1/\sqrt{2})\,(\sigma_x + \sigma_z)$ *et* $H^2 = 1$;

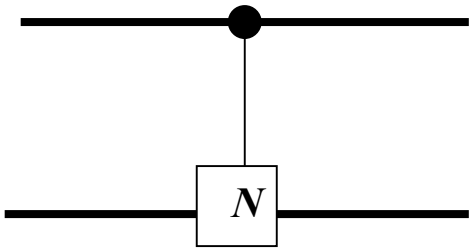
$\sigma_z \equiv U_\pi$; $U_{\pi/2}$ *et* $U_{\pi/4}$ *sont également utiles;*

σ_y *s 'obtient en combinant* σ_x *et* σ_z $(\sigma_y = i\,\sigma_x\,\sigma_z)$.

b. Portes logiques à deux qubits

Ces portes réalisent l 'opération logique élémentaire « si A est vrai, alors faire B…. »

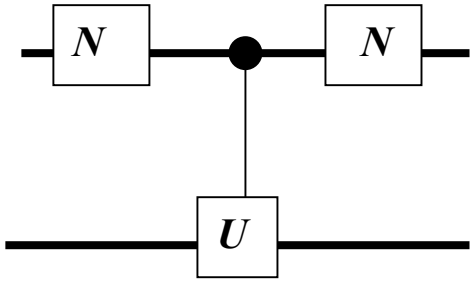




Porte générale « control-U »: U appliqué au bit cible si le bit source vaut 1 et cible non changée si la source vaut 0.
Source toujours invariante.

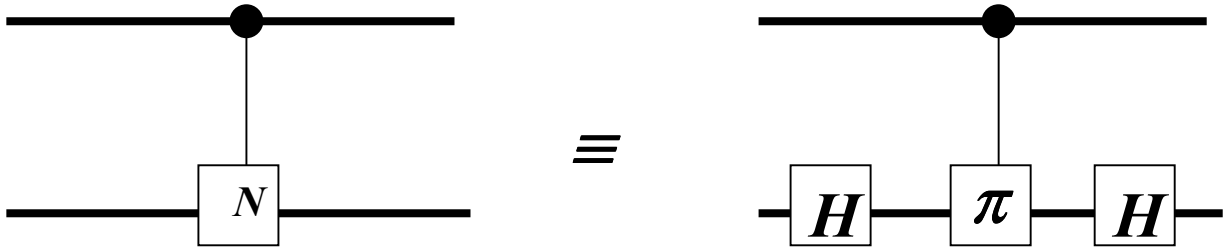
Cas particulier important: $U = \sigma_x$
Porte C-NOT (bascule conditionnelle)

On peut aussi conditionner l 'évolution de la cible à la valeur 0 de la source:



La cible n 'évolue que si la source vaut initialement 0 (effet de la première porte σ_x).
La source revient finalement dans l 'état initial (deuxième porte σ_x)

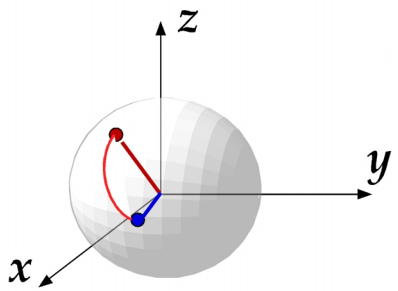
Remarque: La porte C-NOT peut être réalisée avec une porte de phase-π et deux Hadamard:



Plusieurs façons de réaliser en pratique ces portes (voir leçons 9-10)

Toute porte «control-U» peut se décomposer en portes C-NOT et portes à un qubit

Toute opération unitaire U sur un qubit peut se définir de façon générale (à une phase près) comme une rotation d'angles d' Euler α , β , γ sur la sphère de Bloch (produit d'une rotation d'angle α autour de Oz, d'une rotation d 'angle β autour de 0y et d 'une rotation d 'angle α à nouveau autour de Oz):



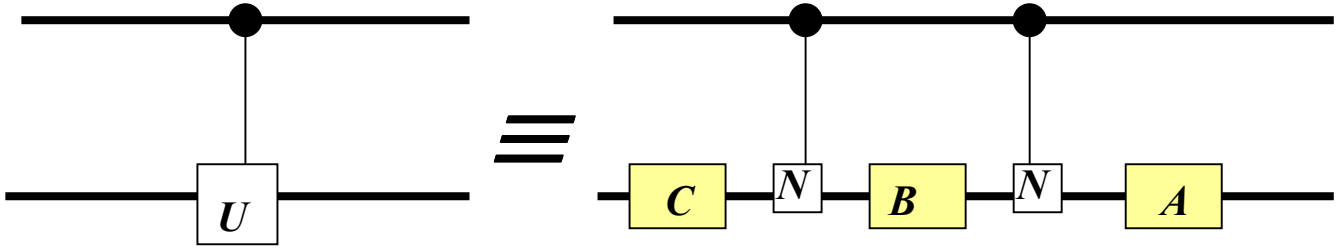
$$U(\alpha,\beta,\gamma) \;=\; R_z(\alpha)\,R_y(\beta)\,R_z(\gamma) = e^{-i\alpha\sigma_z/2}e^{-i\beta\sigma_y/2}e^{-i\gamma\sigma_z/2}$$

Définissons les trois transformations unitaires:

$$A = R_z(\alpha)\;R_y\left(\frac{\beta}{2}\right) \qquad B = R_y\left(-\frac{\beta}{2}\right)\;R_z\left(-\frac{\gamma+\alpha}{2}\right) \qquad C = R_z\left(\frac{\gamma-\alpha}{2}\right)$$

qui satisfont les relations : ABC = 1 (évident) et $A\,\sigma_x\,B\,\sigma_x\,C = U$ (voir plus bas *).

On vérifie alors, en considérant successivement les cas où le qubit source est 0 et 1:

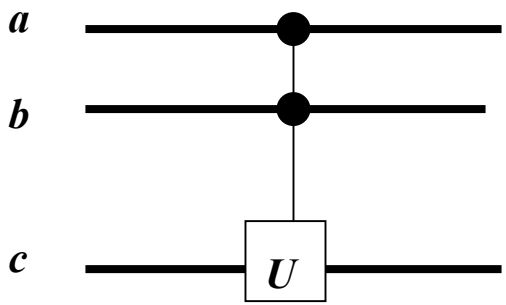


(*) Pour démontrer cette relation, on utilise l 'identité: $\sigma_x B \sigma_x = \sigma_x R_y\left(-\frac{\beta}{2}\right)\;\sigma_x\;\sigma_x R_z\left(-\frac{\gamma+\alpha}{2}\right)\sigma_x$ ***puis***

$$\sigma_x R_y\left(-\frac{\beta}{2}\right)\;\sigma_x = \sigma_x\left[\cos\left(\frac{\beta}{4}\right) - i\sin\left(\frac{\beta}{4}\right)\sigma_y\right]\sigma_x = R_y\left(+\frac{\beta}{2}\right) \quad \textbf{et} \quad \sigma_x R_z\left(-\frac{\gamma+\alpha}{2}\right)\;\sigma_x = \sigma_x\left[\cos\left(\frac{\gamma+\alpha}{4}\right) - i\sin\left(\frac{\gamma+\alpha}{4}\right)\sigma_z\right]\sigma_x = R_z\left(+\frac{\gamma+\alpha}{2}\right)$$

c. Portes à plus de deux bits

On peut être aussi amené à effectuer des opérations du type « si A et B sont vrais, alors faire C». Exige des portes conditionnelles à multiples qubits. Un exemple simple en est la porte à deux bits source et un bit cible:



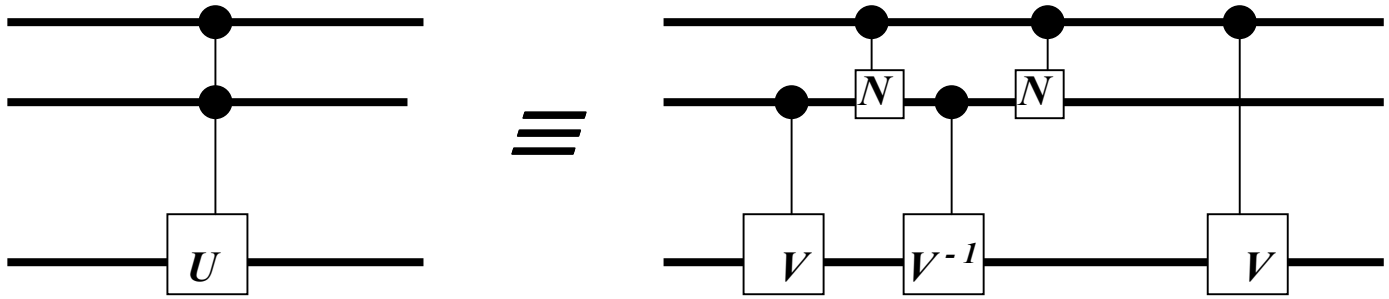
Applique U au bit 3 ssi les bits 1 et 2 valent 1:

$$|a>|b>|c> \rightarrow |a>|b> \otimes U^{a.b}|c>$$

Les bits sources sont toujours invariants

Cas particulier important : avec $U = \sigma_x$ *(opération N), on réalise la porte de Toffoli.*

Une porte à deux bits source et un bit cible se décompose en portes à deux qubits:



Avec V unitaire tel que $V^2 = U$

Lorsque $U = \sigma_x$ *(Toffoli), on prendra* $V = \frac{e^{-i\pi/2}}{\sqrt{2}}(1 + i\sigma_x)$ *(*$V =$ *«* $\sqrt{NOT}$ *»*

On peut ensuite décomposer les portes « control-V » en CNOT et portes à un qubit…Le raisonnement se généralise à plus de deux bits source. Il est possible de ramener toute porte à une combinaison de CNOT et de portes à un bit….

Universalité des portes à un et deux qubits pour le calcul quantique

Principe de la démonstration:

(1) Tout U de H (n) peut se décomposer comme un produit de transformations unitaires $U^{(ab)}$ «à deux niveaux» n 'impliquant chacune que deux états $|\{a\}\rangle$ et $|\{b\}\rangle$:

$$U = \Pi_{(ab)} U^{(ab)}$$

$$U^{(ab)}|\{a\}\rangle = U_{aa}|\{a\}\rangle + U_{ba}|\{b\}\rangle; \quad U^{(ab)}|\{b\}\rangle = U_{ab}|\{a\}\rangle + U_{bb}|\{b\}\rangle;$$

$$U^{(ab)}|\{c\} \neq \{a\}, \{b\}\rangle = |\{c\}\rangle$$

(2) Deux états $|\{a\}\rangle$, $|\{b\}\rangle$ quelconques sont reliés par une chaîne d' au plus n – 1 états intermédiaires tels que deux états consécutifs ne diffèrent l 'un de l 'autre que par un des bits:

Exemple (n = 6):

$\{a\} \equiv$ 0 1 1 1 0 1

$\{a'\}$ 0 0 1 1 0 1

$\{a''\}$ 0 0 1 0 0 1

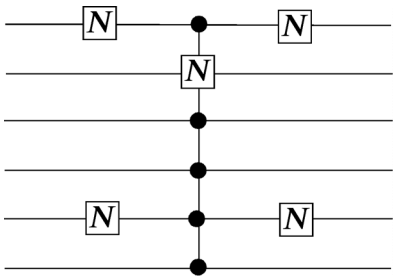
$\{b\}$ 0 0 1 0 1 1

On passe d 'un vecteur au suivant de la chaîne en faisant agir des portes conditionnelles CNOT à (n–1) bits sources:

$$T^{(1)}_{CNOT}|\{a\}\rangle = |\{a'\}\rangle; \quad T^{(1)}_{CNOT}|\{a'\}\rangle = |\{a\}\rangle;$$

$$T^{(1)}_{CNOT}|\{c\} \neq \{a\}, \{a'\}\rangle = |\{c\}\rangle.$$

Porte « bascule » avec bit 2 comme cible, conditionnée à bit 1 et 5 valant 0, bits 3,4 et 6 valant 1: échange a et a' en laissant tous les autres états invariants



$T^{(1)}$

On continue l ’opération pour échanger {a’} et {a’’}:

$\{a\} \equiv 0\ 1\ 1\ 1\ 0\ 1$

$\{a'\}\ \ 0\ 0\ 1\ 1\ 0\ 1$

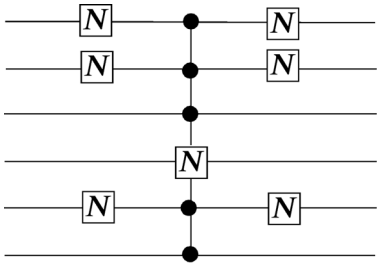
$\{a''\}\ \ 0\ 0\ 1\ 0\ 0\ 1$

$\{b\}\ \ 0\ 0\ 1\ 0\ 1\ 1$

$$T^{(2)}_{CNOT}\,|\{a'\}\rangle = |\{a''\}\rangle;\quad T^{(2)}_{CNOT}\,|\{a''\}\rangle = |\{a'\}\rangle;$$

$$T^{(2)}_{CNOT}\,|\{c\} \neq \{a'\}, \{a''\}\rangle = |\{c\}\rangle.$$

Porte « bascule » avec bit 4 comme cible, conditionnée à bit 1, 2 et 5 valant 0, bits 3, et 6 valant 1: échange a’ et a’’ en laissant tous les autres états invariants



$T^{(2)}$

Finalement, après au plus n – 1 opérations, {a} est transformé en un état {a’’} ne différant de {b} que par un bit (ici le 5). Récapitulons l ’action du produit des portes:

$$T^{(2)}_{CNOT}\,T^{(1)}_{CNOT}\,|\{a\}\rangle = |\{a''\}\rangle;$$

$$T^{(2)}_{CNOT}\,T^{(1)}_{CNOT}\,|\{a'\}\rangle = |\{a\}\rangle;$$

$$T^{(2)}_{CNOT}\,T^{(1)}_{CNOT}\,|\{a''\}\rangle = |\{a'\}\rangle;$$

$$T^{(2)}_{CNOT}\,T^{(1)}_{CNOT}\,|\{b\}\rangle = |\{b\}\rangle;$$

$$T^{(2)}_{CNOT}\,T^{(1)}_{CNOT}\,|\{c\}\rangle = |\{c\}\rangle$$

(si $\{c\} \neq \{a\},\{a'\},\{a''\}$);

On applique ensuite la porte conditionnelle effectuant l 'opération unitaire $\hat{U}\,|0\rangle = U_{aa}|0\rangle + U_{ba}|1\rangle\,;\ \hat{U}\,|1\rangle = U_{ab}|0\rangle + U_{bb}|1\rangle$ ***sur le qubit qui diffère entre {a''} et { b }, si et seulement si les autres bits ont les valeurs communes qu' ils ont dans {a''} et { b }:***

$\{a\} \equiv \ 0\ 1\ 1\ 1\ 0\ 1$

$\{a'\}\ \ 0\ 0\ 1\ 1\ 0\ 1$

$\{a''\}\ \ 0\ 0\ 1\ 0\ 0\ 1$

$\{b\}\ \ \ 0\ 0\ 1\ 0\ 1\ 1$

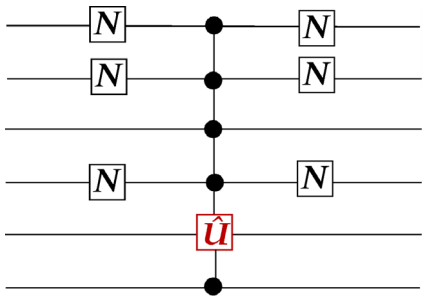
$$T^{(3)}_{\hat{U}}\ T^{(2)}_{CNOT}\ T^{(1)}_{CNOT}\,|\{a\}\rangle = U_{aa}|\{a''\}\rangle + U_{ba}|\{b\}\rangle;$$

$$T^{(3)}_{\hat{U}}\ T^{(2)}_{CNOT}\ T^{(1)}_{CNOT}\,|\{b\}\rangle = U_{ab}|\{a''\}\rangle + U_{bb}|\{b\}\rangle;$$

$$T^{(3)}_{\hat{U}}\ T^{(2)}_{CNOT}\ T^{(1)}_{CNOT}\,|\{a'\}\rangle = |\{a\}\rangle\,;\ T^{(3)}_{\hat{U}}\ T^{(2)}_{CNOT}\ T^{(1)}_{CNOT}\,|\{a''\}\rangle = |\{a'\}\rangle;$$

$$T^{(3)}_{\hat{U}}\ T^{(2)}_{CNOT}\ T^{(1)}_{CNOT}\,|\{c\} \neq \{a\}, \{a'\}, \{a''\}, \{b\}\rangle = |\{c\}\rangle$$

Porte $T^{(3)}$ ***«control-Û» avec bit 5 comme cible, conditionnée à bit 1, 2 et 4 valant 0, bits 3 et 6 valant 1: Le produit*** $T^{(3)}\ T^{(2)}\ T^{(1)}$ ***effectue une opération qui applique le sous espace*** $|\{a\}\rangle, |\{b\}\rangle$ ***dans*** $|\{a''\}\rangle, |\{b\}\rangle$ ***(et transforme*** $|a'\rangle$ ***en*** $|a\rangle$, $|a''\rangle$ ***en*** $|a'\rangle$***)***



$T^{(3)}$

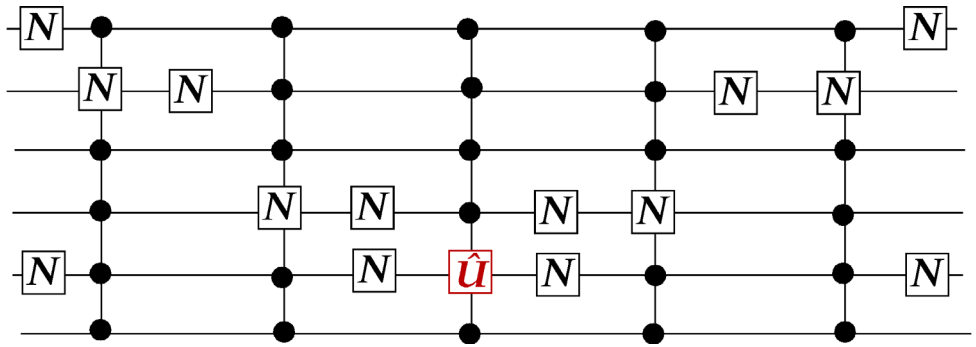
Revenons enfin de {a''} à { a } en appliquant à nouveau $T^{(2)}_{CNOT}$ ***et*** $T^{(1)}_{CNOT}$ ***(dans cet ordre):***

$$T^{(1)}_{CNOT}\ T^{(2)}_{CNOT}\ T^{(3)}_{\hat{U}}\ T^{(2)}_{CNOT}\ T^{(1)}_{CNOT}\,|\{a\}\rangle = U_{aa}|\{a\}\rangle + U_{ba}|\{b\}\rangle;$$

$$T^{(1)}_{CNOT}\ T^{(2)}_{CNOT}\ T^{(3)}_{\hat{U}}\ T^{(2)}_{CNOT}\ T^{(1)}_{CNOT}\,|\{b\}\rangle = U_{ab}|\{a\}\rangle + U_{bb}|\{b\}\rangle.$$

$$T^{(1)}_{CNOT}\ T^{(2)}_{CNOT}\ T^{(3)}_{\hat{U}}\ T^{(2)}_{CNOT}\ T^{(1)}_{CNOT}\,|\{c\} \neq \{a\}, \{b\}\rangle = |\{c\}\rangle$$

Finalement, l 'opération unitaire à deux états est décomposable en un produit de portes conditionnelles de type CNOT et Control-U: dans l 'exemple choisi ici, ce produit est donné par:



$$T^{(1)}_{CNOT}\; T^{(2)}_{CNOT}\; T^{(3)}_{\hat{U}}\; T^{(2)}_{CNOT}\; T^{(1)}_{CNOT}$$

Chacune des portes conditionnelles à n–1 bits sources peut se décomposer en un produit de portes à un et deux qubits, et les portes à deux qubits se réduire à des portes C-NOT (ou des portes de phase π) et des portes à un qubit:

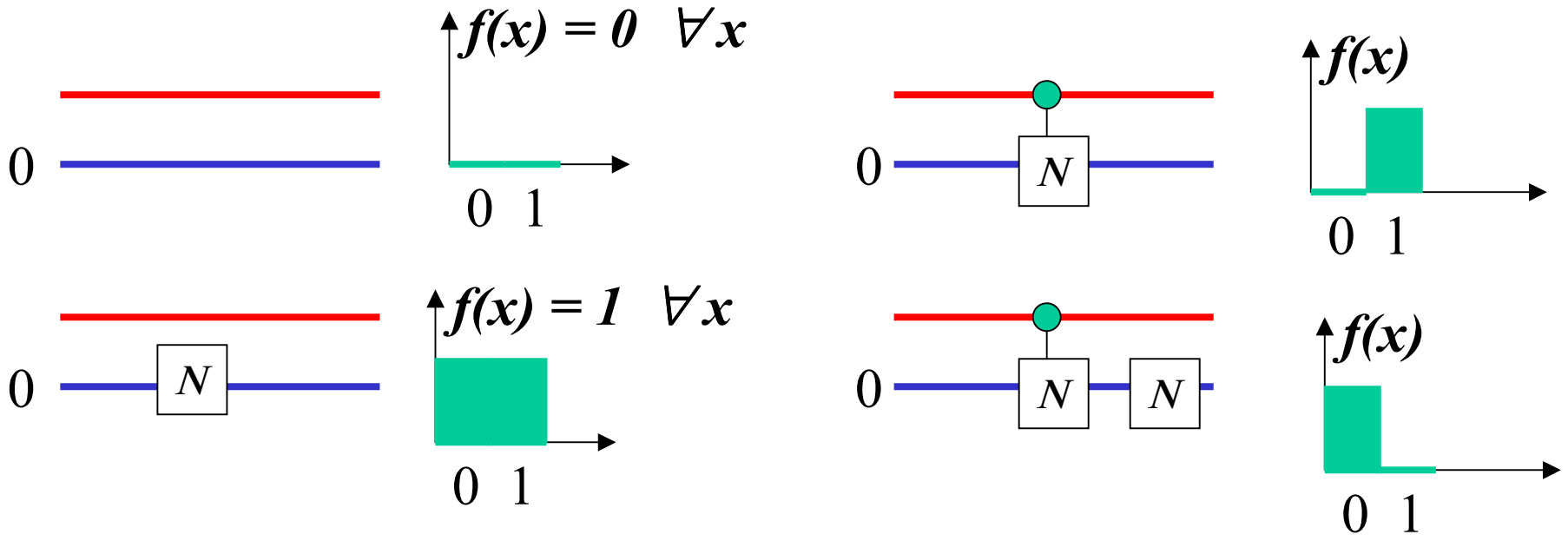
<u>Les portes CNOT (ou portes de phase–π) à deux qubits associées à des portes à un qubit sont universelles.</u>

Calcul de fonctions simples à l 'aide de portes quantiques élémentaires

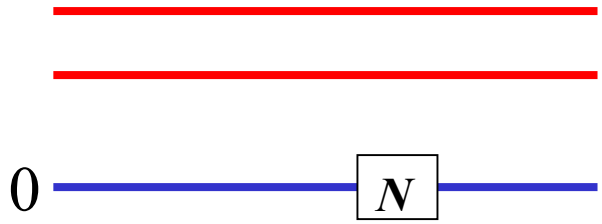
Toutes les fonctions de A (n qubits) dans B (p qubits) peuvent se décomposer en p fonctions booléennes de A dans [0, 1]. Nous allons montrer comment on peut explicitement calculer ces fonctions pour n = 1 [applications de (0, 1) dans (0, 1)] et pour n = 2 [(0, 1, 2, 3) dans (0, 1)].

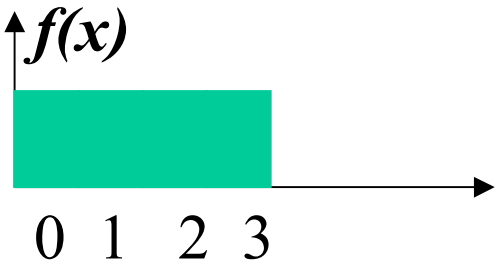
Applications de [0,1] dans [0,1]:

4 fonctions:

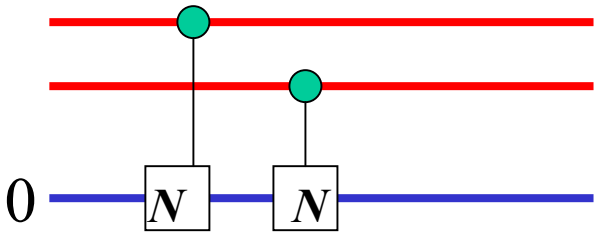


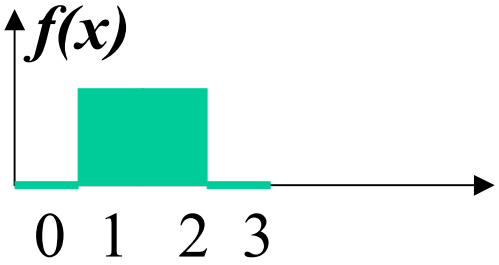
Quelques applications de [0, 1, 2, 3] dans [0, 1]
(2^4 = 16 fonctions en tout)



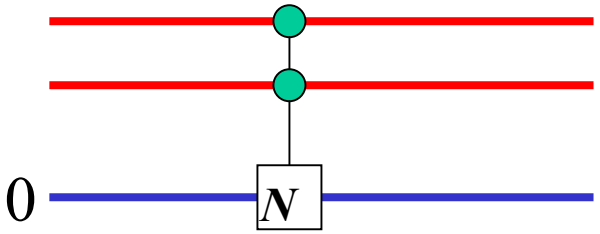


Exemple de fonction constante (deux)

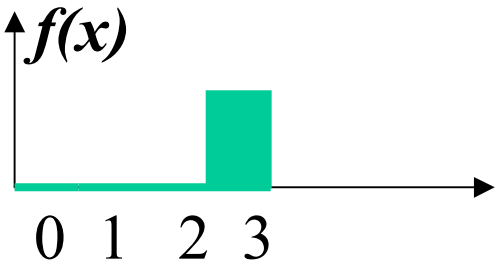




Exemple de Fonction balancée (autant de f(x) = 0 que de f(x) = 1) (six)



Porte de Toffoli décomposable en portes CNOT et portes à un qubit (à faire en exercice)



Fonction ni constante ni balancée (huit)

Tous les autres cas se déduisent simplement de ceux-ci