GRENOBLE INP UGA Ensimag

École nationale supérieure d'informatique et de mathématiques appliquées

Introduction à l'Informatique Quantique

Calcul et algorithmes quantiques

Alastair Abbott

(Centre Inria de l'UGA)

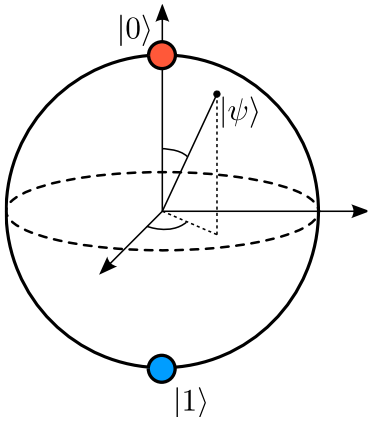
7 février, 2024

Inria

Sommaire

- Rappel des principes de base
- Modèles de calcul quantique
  - Circuits quantiques
- Construction d'algorithmes quantiques
  - Sources des avantages quantiques
  - Exemple : l'algorithme de Grover
  - L'algorithme de Shor
  - Survol des algorithmes quantiques
- La puissance du calcul quantique et ses limitations

Rappel – principes de base

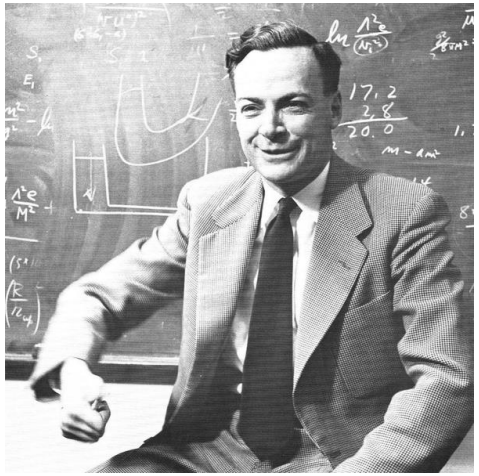


- Qubits :
  - Un qubit : $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$
  - Deux qubits : si $|\psi\rangle = \begin{pmatrix}\alpha_0\\ \alpha_1\end{pmatrix}$ et $|\phi\rangle = \begin{pmatrix}\beta_0\\ \beta_1\end{pmatrix}$ alors $|\psi\rangle \otimes |\phi\rangle = \begin{pmatrix}\alpha_0\beta_0\\ \alpha_0\beta_1\\ \alpha_1\beta_0\\ \alpha_1\beta_1\end{pmatrix}$
  - n qubits : $|\psi\rangle = \sum_{x_1 \dots x_n} \alpha_{x_1 \dots x_n} |x_1 \dots x_n\rangle$
- Mesures :
  - Règle de Born : probabilité de mesurer 0 est $|\alpha_0|^2$, etc.
  - L'« effondrement » de l'état du (des) qubit(s)
- Transformations unitaires : $U^\dagger U = UU^\dagger = \mathbb{I}$
  - Linéaire, réversible
  - Exemples : X, Z, H, CNOT,…

Pourquoi le calcul quantique ?

Richard Feynman (1981)

"Can quantum systems be probabilistically simulated by a classical computer? [. . .] The answer is almost certainly, No! [. . .] if you want to make a simulation of nature, you'd better make it quantum mechanical."

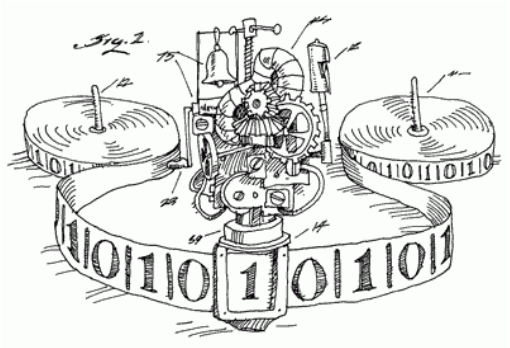


- Dimension de l'espace de Hilbert exponentielle en nombre de qubits
  - n qubits – superposition de 2^n valeurs ($|\psi\rangle = \sum_{x_1 \dots x_n} \alpha_{x_1 \dots x_n} |x_1 \dots x_n\rangle$)
  - Parallélisme massif venant de superposition et la linéarité des transformations
- Avantages potentiels en calcul
- Comprendre les limites fondamentales du calcul

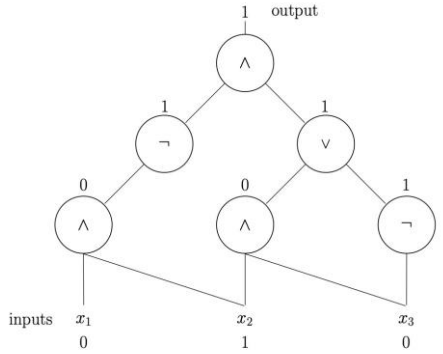
Modèles

Modèles de calcul

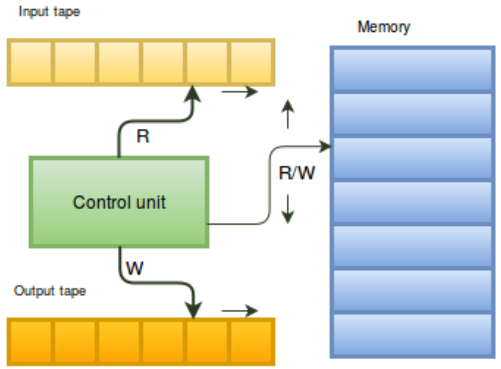
- Comment représenter, modéliser des calculs ?
  - Qu'est-ce qu'on peut calculer ? Comment quantifier le coût d'un calcul ?
- Besoin d'un modèle formel d'un calcul ou un ordinateur



Machines de Turing



Circuits booléens

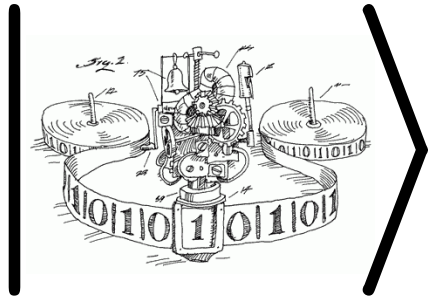


Random access machines

- Robuste : tous ces modèles sont essentiellement équivalents
- On formaliser tout algorithme « informel » dans n'importe quel de ces modèles

Modèles de calcul quantique

- Quel modèle pour le calcul quantique ?
- Première idée : machine de Turing quantique (MTQ)



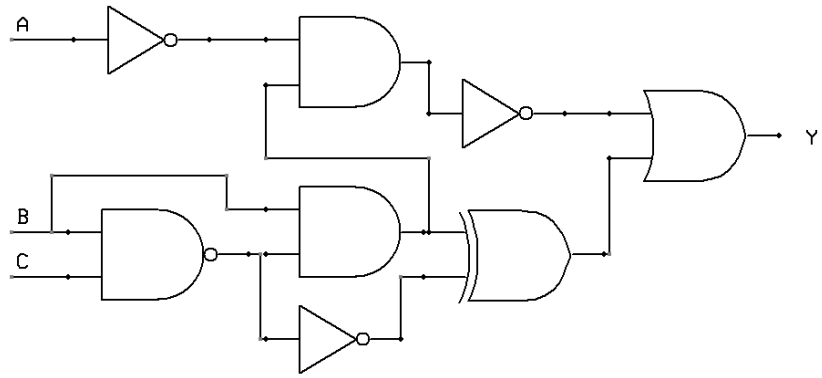
David Deutsch (1985)

- Difficile à travailler avec des MTQs
- Comment formaliser ou décrire simplement un algorithme ?
- Nous utilisons plutôt un modèle inspiré par des circuits booléens
  - Plus proche à ce qu'on sait construire en réalité

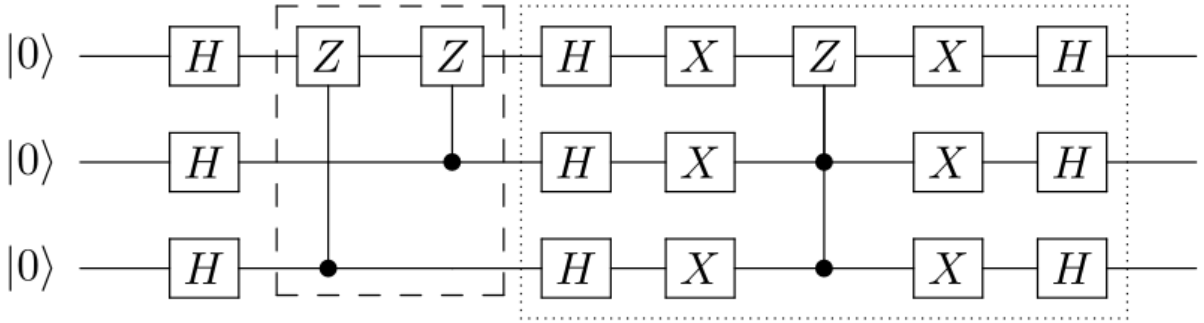
Circuits : booléennes et quantiques

- Circuits booléens basés sur la logique booléenne
  - Fils = variables booléennes (bits)
  - Portes logiques : AND, OR, XOR, NOT, …

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



- Pour le quantique, besoin d'une nouvelle logique
  - Fils = qubits
  - Portes quantiques : unitaires



Rappel : opérations élémentaires

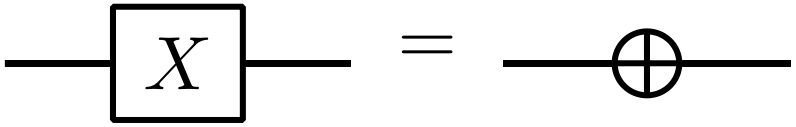
- La porte X (ou **NOT**) : $X|0\rangle = |1\rangle,\ X|1\rangle = |0\rangle$ $\quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
- La porte Z (« phase-flip ») : $Z|0\rangle = |0\rangle,\ Z|1\rangle = -|1\rangle$ $\quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
- La porte H (ou **de Hadamard**) : $H|0\rangle = |+\rangle,\ H|1\rangle = |-\rangle$ $\quad H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$,

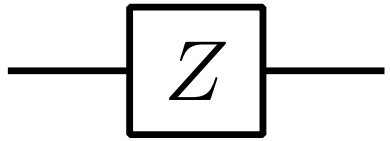
  où $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$
- La porte phase R_φ : $R_\varphi|0\rangle = |0\rangle,\ R_\varphi|1\rangle = e^{i\varphi}|1\rangle$ $\quad R_\varphi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}$
- La porte NOT-contrôlé (CNOT) : $\mathrm{CNOT}|xy\rangle = |x\rangle \otimes |x \oplus y\rangle$

  $\mathrm{CNOT}|00\rangle = |00\rangle$
  $\mathrm{CNOT}|01\rangle = |01\rangle$
  $\mathrm{CNOT}|10\rangle = |11\rangle$
  $\mathrm{CNOT}|11\rangle = |10\rangle$

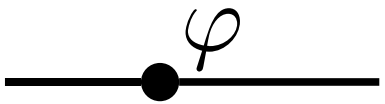
  $\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

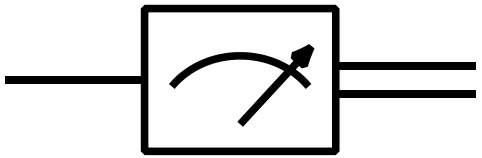
Portes quantiques sur 1 qubit

- X : 

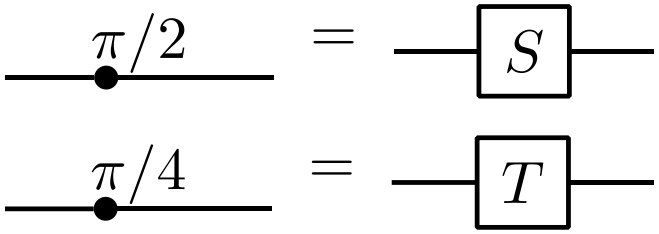
- Z : 

- H : 

- R_φ : 

- Mesure : 

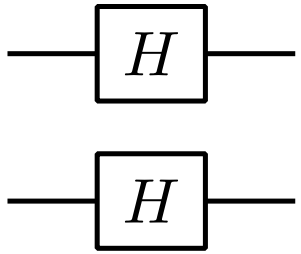
Cas spéciaux :



Portes quantiques sur 2 qubits

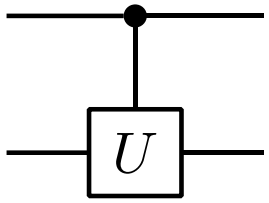
- Portes à 1 qubit en parallèle

ex : $H \otimes H$:

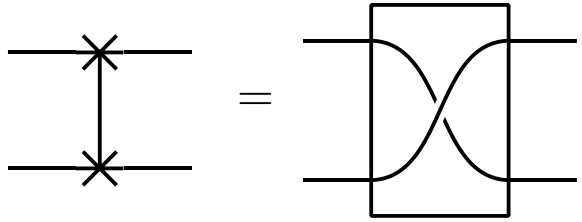


- CNOT :

- c-U (U contrôlé) :

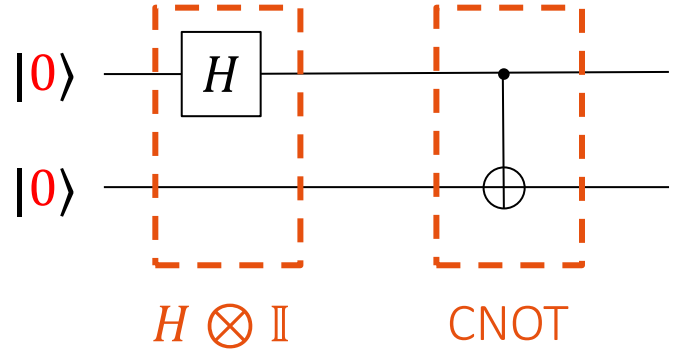


- SWAP :



Exemple d'un circuit

- Quel état est créé par le circuit suivant ?



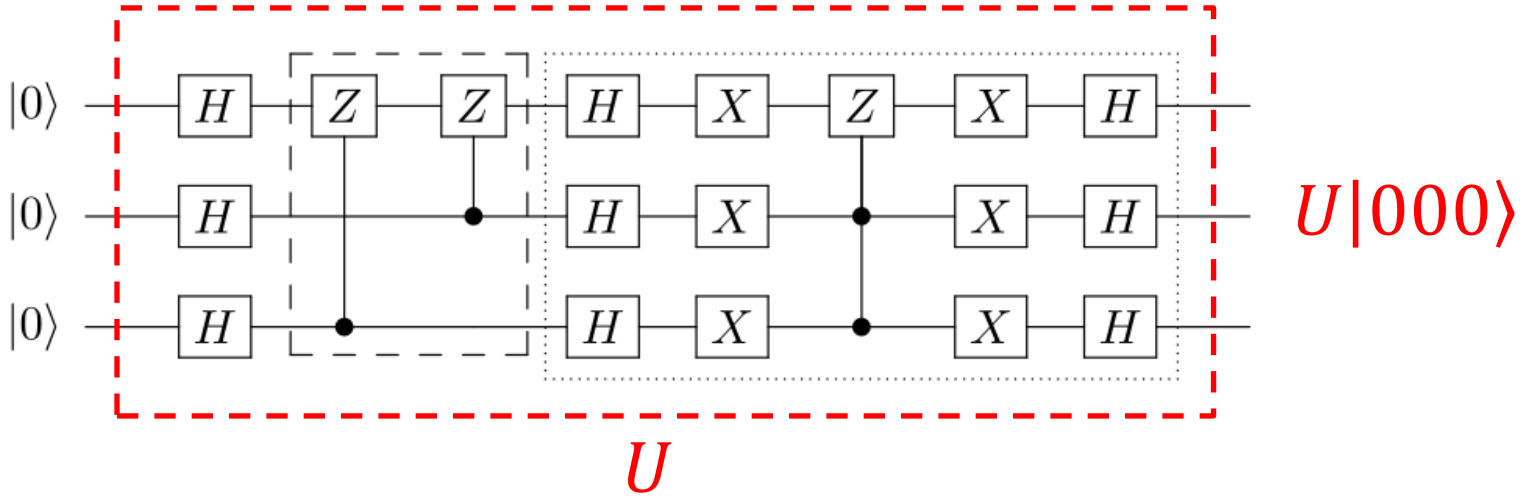
$$(H \otimes \mathbb{I})|00\rangle = H|0\rangle \otimes \mathbb{I}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

$$\mathrm{CNOT} \cdot (H \otimes \mathbb{I})|00\rangle = \frac{1}{\sqrt{2}}\mathrm{CNOT}(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

- Comment le modifier pour générer l'état $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$?

Calculer avec des circuits quantiques

- Chaque circuit définit une transformation unitaire U sur n qubits



- Est-ce qu'on peut réaliser toute unitaire avec un circuit ? Avec quelles portes ?
- Rappel : toute fonctionne booléenne peut être réalisée avec un circuit booléen et les portes AND, OR, NOT

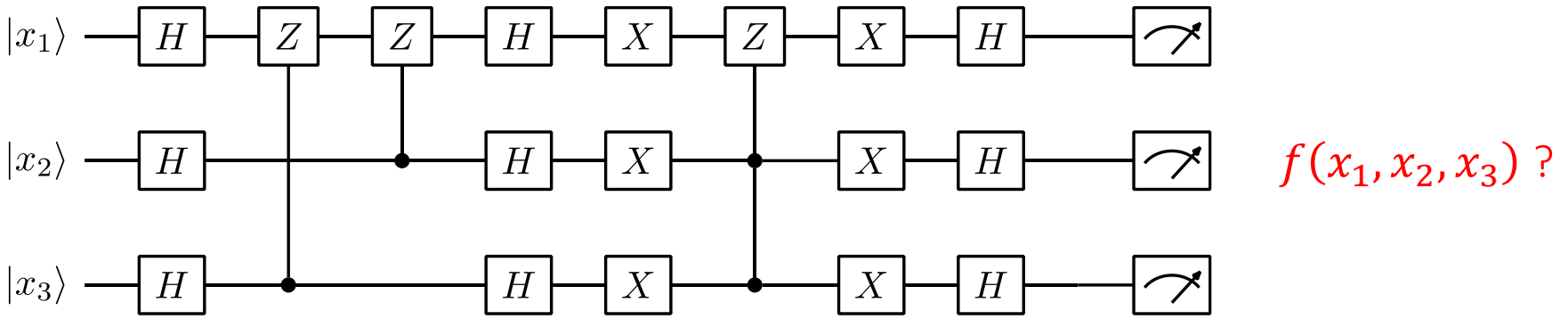
Ensembles des portes universels

- L'importance de CNOT
  - L'ensemble des portes unitaires à 1 qubit + la porte CNOT est universel
  - Toute unitaire peut être réalisé avec ces portes
- Est-il raisonnable de devoir se servir d'un ensemble infini pour cela ?
  - Avec un ensemble fini, impossible à réaliser toute unitaire parfaitement

- Solovay & Kitaev (1995) : L'ensemble $\{H, T, \mathrm{CNOT}\}$ est universel d'une manière approximative
  - On peut approximer toute unitaire à une précision arbitraire avec ses portes
- Conséquence : on peut se focaliser sur des circuits construits des portes élémentaires

L'étape finale : l'extraction du résultat

- Pour calculer, il faudra extraire de l'information classique
  - On mesure un (ou plusieurs) qubits à la fin du circuit



- Le calcul quantique est donc un modèle de calcul probabiliste
  - On devrait obtenir le bon résultat avec une grande probabilité
  - Et en essayant de minimiser le nombre de portes utilisées (temps du calcul)

Algorithmes

Source de la puissance quantique

- Pourquoi penser que le quantique peut aider résoudre des problèmes ?
- n qubits peuvent être dans une superposition de 2^n états de base, et sont définis par 2^n paramètres complexes
  - On pourrait penser que la quantité d'information qu'ils tiennent est aussi exponentielle
- On peut « calculer » en parallèle en appliquant une unitaire à tous les inputs possibles en même temps *parce que l'informatique quantique est linéaire*

$$U(\alpha_0|0\rangle + \alpha_1|1\rangle) = \alpha_0 U|0\rangle + \alpha_1 U|1\rangle$$

$$U\left(\sum_{x_1 \dots x_n} \alpha_{x_1 \dots x_n} |x_1 \dots x_n\rangle\right) = \sum_{x_1 \dots x_n} \alpha_{x_1 \dots x_n} U|x_1 \dots x_n\rangle$$

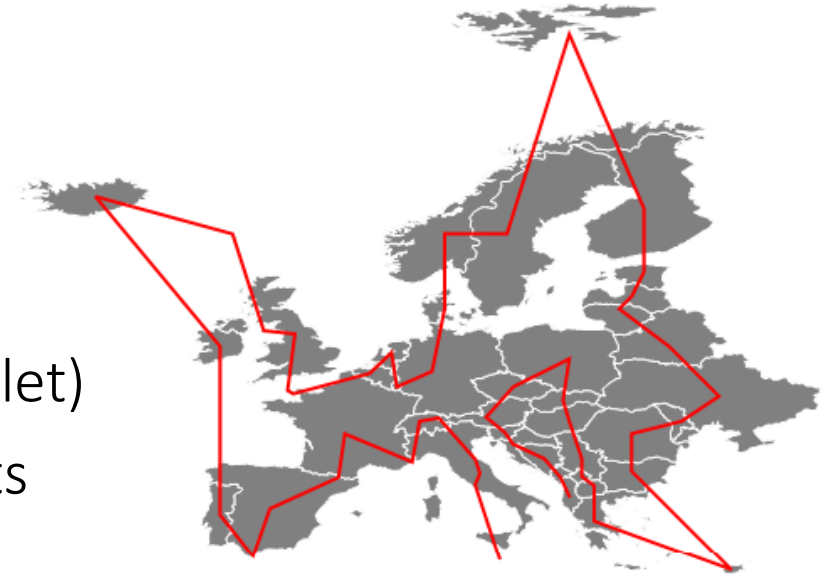
- Intuition naïve : exploiter superposition et linéarité

Comment exploiter le quantique ?

- Essayons bêtement d'utiliser ces principes pour résoudre un problème

Problème du voyageur de commerce
Trouver le circuit x le plus court qui passe par chacune de n villes

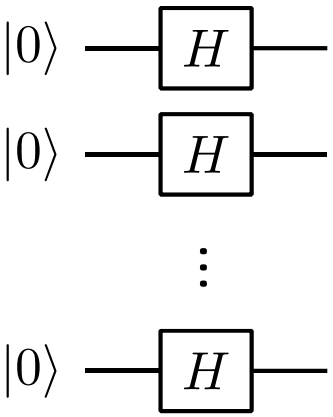
- Si on appelle les villes $1, \dots, n$, un circuit est une permutation (ex. 372681954)
- Soit $f(x)$ la longueur d'un circuit. On veut donc trouver le x qui minimise $f(x)$
- Classiquement ce problème est difficile (NP-complet)
- Intuitivement : mieux est d'essayer tous les circuits possibles ($n! > 2^n$ possibilités)



Une solution quantique?

- **Tentative 1** : si l'on peut trouver un circuit quantique qui calcule $f(x)$, on peut calculer $f(x)$ sur toute permutation x dans une superposition, et simplement prendre le x qui minimise $f(x)$
  - On peut voir x comme un numéro en binaire de $\approx n$ bits
  - On utilise des portes de Hadamard pour créer la superposition :

$$H^{\otimes n}|00\ldots0\rangle = \left(\frac{1}{\sqrt{2}}\right)^n (|0\rangle + |1\rangle)\ldots(|0\rangle + |1\rangle)$$
$$= \frac{1}{\sqrt{2^n}}\sum_{x_1\ldots x_n}|x_1 x_2 \ldots x_n\rangle = \frac{1}{\sqrt{2^n}}\sum_x |x\rangle$$

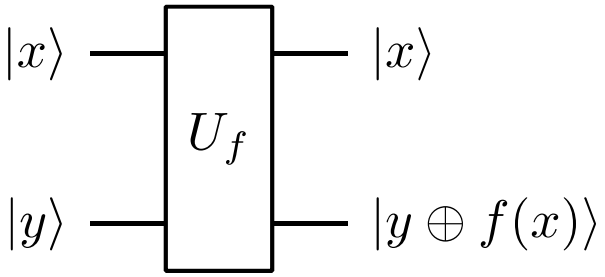


  - Il nous faut une U_f telle que $U_f|x\rangle = |f(x)\rangle$ ….

Une solution quantique?

- Problème 1 : une telle U_f n'existe pas en générale !!
  - Cette transformation n'est pas unitaire : les transformations unitaires sont *réversibles*, alors que plusieurs valeurs de x puissent donner la même valeur de $f(x)$
  - Si $U_f|x\rangle = |f(x)\rangle = U_f|x'\rangle = |f(x')\rangle$, alors $U_f^\dagger|f(x)\rangle =$??

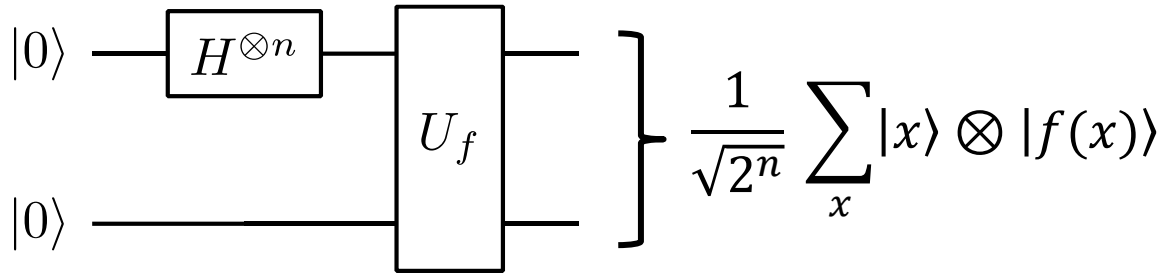
- Solution : on garde une copie de x et met le résultat ailleurs



$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle|0\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_x |x\rangle|f(x)\rangle$$

Une solution quantique?

- Supposons que l'on a trouvé une telle U_f pour le problème du voyageur de commerce
  - Notez : c'est facile à calculer classiquement $f(x)$ – ici, la longueur d'un circuit – et ce calcul peut être transformé en circuit quantique sans trop de difficulté majeure
- On a alors un circuit qui calcule la superposition voulue :



- Problème 2 : comment trouver le x qui minimise $f(x)$?
  - Si l'on mesure l'état, on trouve une paire $(x, f(x))$ avec probabilité $1/2^n$
  - Pas mieux que d'essayer un x au hasard

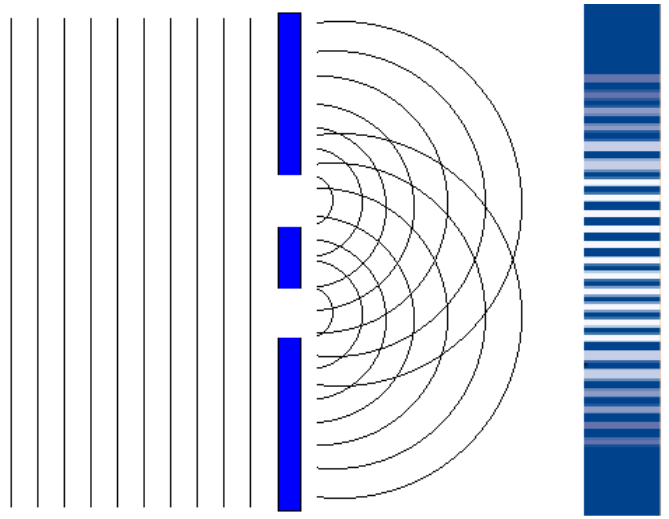
Les limites du parallélisme quantique

- Ce problème est plus difficile à surmonter : on peut essayer toutes les possibilités en parallèle, mais on ne peut pas accéder à tous les résultats en parallèle
  - Les contraintes des mesures probabilistes, transformations unitaires, etc. sont très fortes !
- Il faut assurer que seulement les bonnes solutions (ici, les petites $f(x)$) ont une grande amplitude
  - Quand on mesure, on veut une bonne probabilité à trouver une bonne solution
  - Trouver un circuit qui fait ça est toute la difficulté
  - Équilibre délicat entre les effets (superposition, intrication) et les contraints

Le calcul quantique ne marche pas simplement parce qu'il peut essayer toutes les possibilités simultanément !!

Interférence quantique

- Comment utiliser le parallélisme quantique pour faire quelque chose d'utile ?
- Idée clé : interférence quantique



  - Appliquer des unitaires pour que les amplitudes des termes désirés s'interfèrent constructivement, et les autres s'annihilent

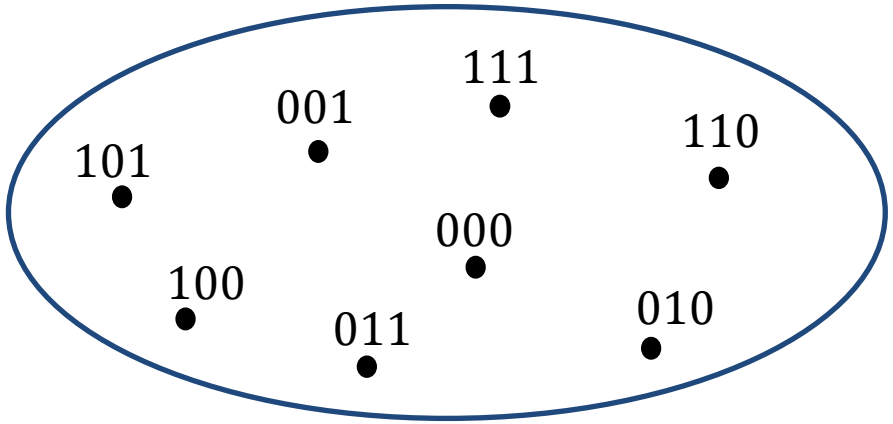
- Exemple : $H\left(\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\right) = \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|0\rangle - \frac{1}{2}|1\rangle = |0\rangle$

Exemple : recherche non-structurée

- Regardons une version abstraite du problème précèdent

Trouver x tel que $f(x) = 1$
où $x \in \{0,1\}^n$ est une chaine de n bits, et $f(x) \in \{0,1\}$ est une fonctionne booléenne

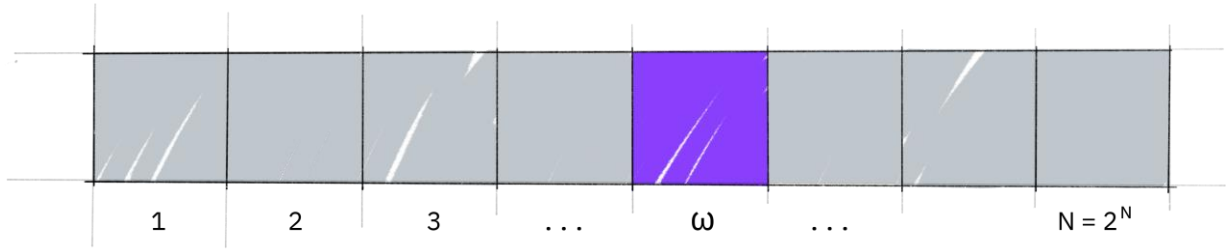
$n = 3$

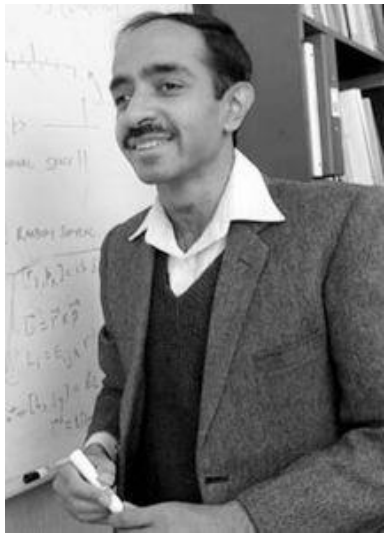


- Meilleur algorithme classique : $O(2^n)$ evaluations de f

L'algorithme de Grover

- Lov Grover a prouvé un avantage quantique en 1996
- Il trouve une approche pour amplifier l'amplitude recherchée



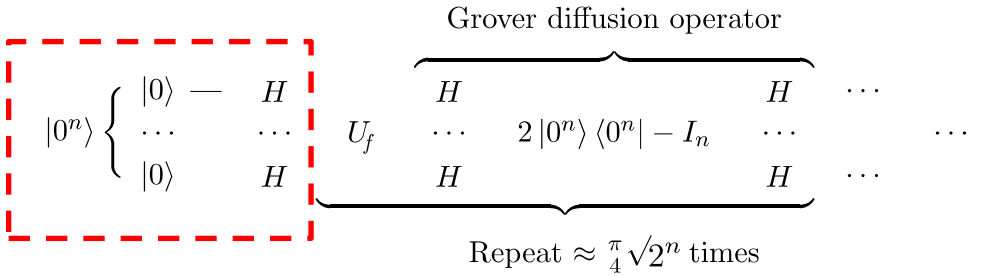


1. Identifier $|\omega\rangle$ et invertir son amplitude en multipliant par -1
2. Réfléchir les amplitudes autour du moyen pour amplifier celle de $|\omega\rangle$ (interférence)
3. Répéter étapes 1 et 2

$$|0^n\rangle \left\{ \begin{array}{lllll} |0\rangle & — & H & & \overbrace{H \quad \cdots \quad 2\,|0^n\rangle\langle 0^n| - I_n \quad \cdots \quad H}^{\text{Grover diffusion operator}} \quad \cdots \\ \cdots & & \cdots & U_f & \\ |0\rangle & & H & & \underbrace{H \quad \cdots \quad H}_{\text{Repeat} \approx \frac{\pi}{4}\sqrt{2^n} \text{ times}} \quad \cdots \end{array} \right. \quad \cdots$$

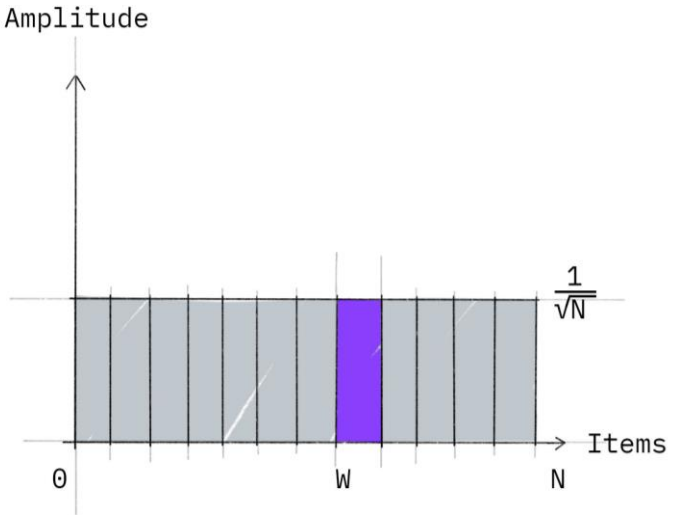
L'algorithme de Grover – détails

Étape 1 : créer la superposition



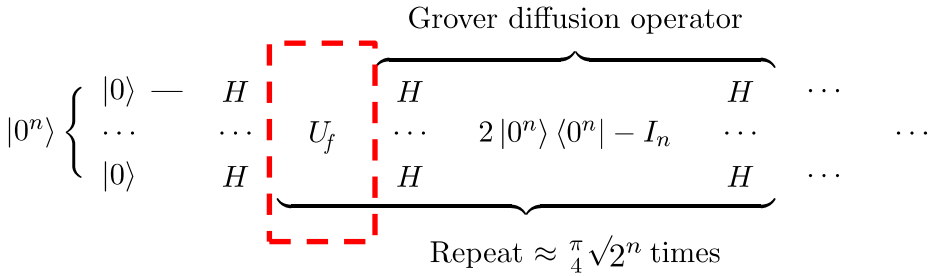
- L'algorithme commence par créer une superposition uniforme sur tous les x

$$H^{\otimes n}|00\ldots0\rangle = \left(\frac{1}{\sqrt{2}}\right)^n (|0\rangle + |1\rangle)\ldots(|0\rangle + |1\rangle)$$
$$= \frac{1}{\sqrt{2^n}}\sum_{x_1\ldots x_n}|x_1x_2\ldots x_n\rangle$$
$$= \frac{1}{\sqrt{2^n}}\sum_x|x\rangle$$
$$= |s\rangle$$



L'algorithme de Grover – détails

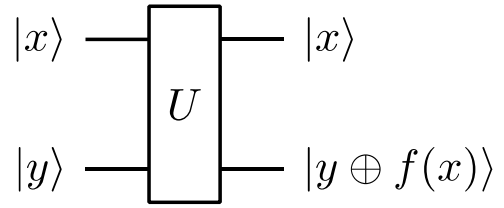
Étape 2 : invertir l'amplitude de $|\omega\rangle$



- L'algorithme utilise une unitaire qui fait la transformation :

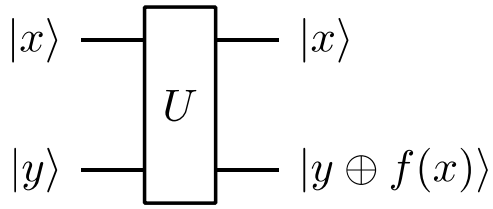
$$U_f|x\rangle = (-1)^{f(x)}|x\rangle = \begin{cases} |x\rangle \text{ si } x \neq \omega \\ -|x\rangle \text{ si } x = \omega \end{cases}$$

  - Clairement U_f est unitaire
  - Comment construire U_f à partir de la forme habituelle de représenter f ?



L'algorithme de Grover – détails

Retour de phase



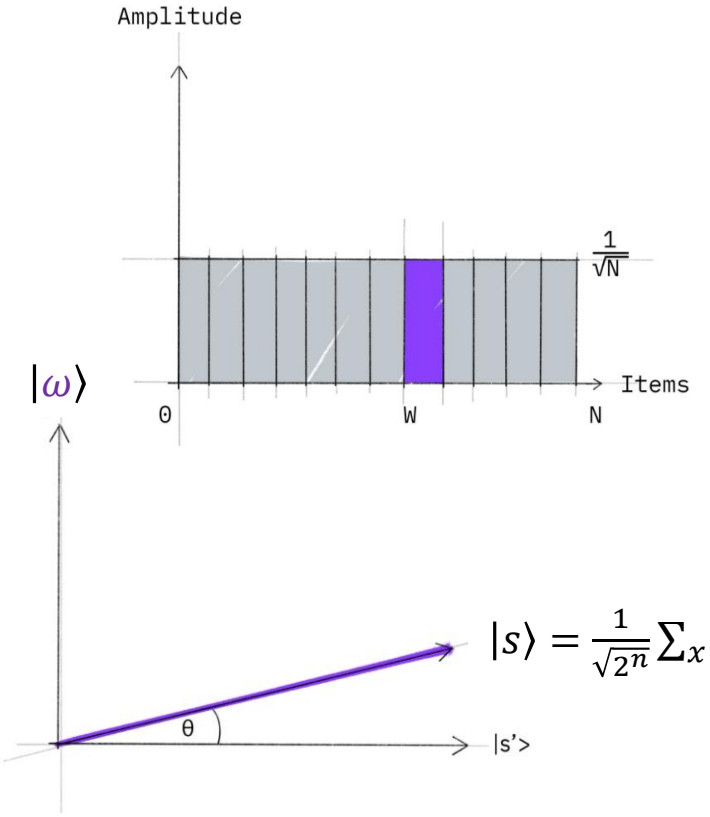
- Utilisons l'unitaire « normale » pour U_f avec $|y\rangle = |-\rangle$

$$U_f|x\rangle|-\rangle = \frac{1}{\sqrt{2}}\left(U_f|x\rangle|0\rangle - U_f|x\rangle|1\rangle\right)$$
$$= \frac{1}{\sqrt{2}}\left(|x\rangle|f(x)\rangle - |x\rangle|1 \oplus f(x)\rangle\right)$$
$$= (-1)^{f(x)}\,|x\rangle\,|-\rangle$$

- On peut construire la forme requise pour l'algorithme

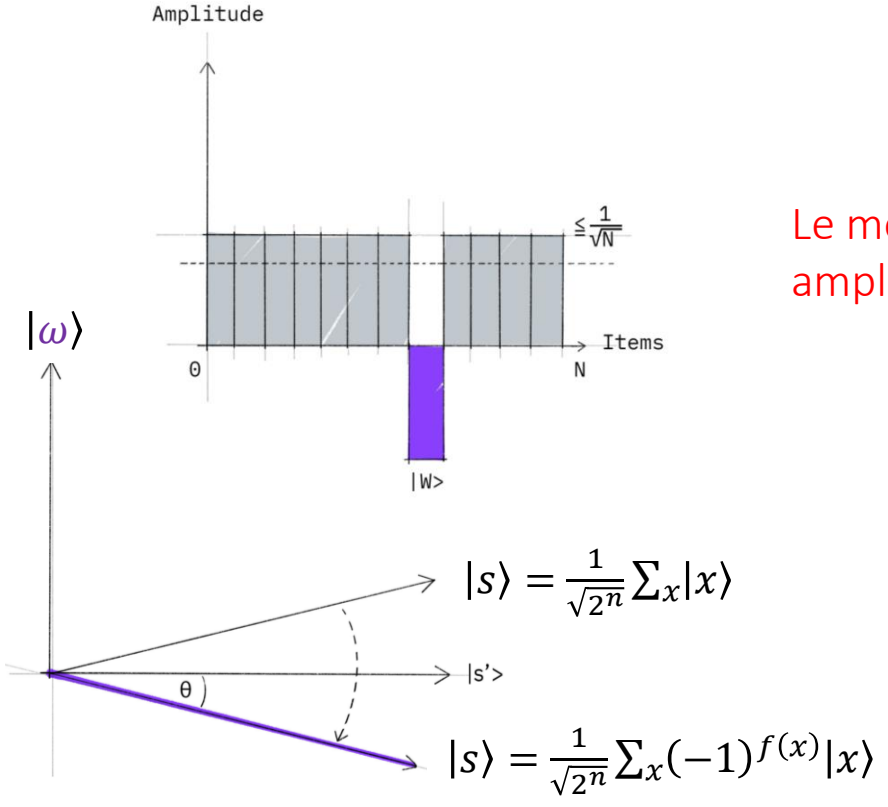
L'algorithme de Grover – détails

Après étape 1



$$|s\rangle = \frac{1}{\sqrt{2^n}}\sum_x |x\rangle$$

Après étape 2



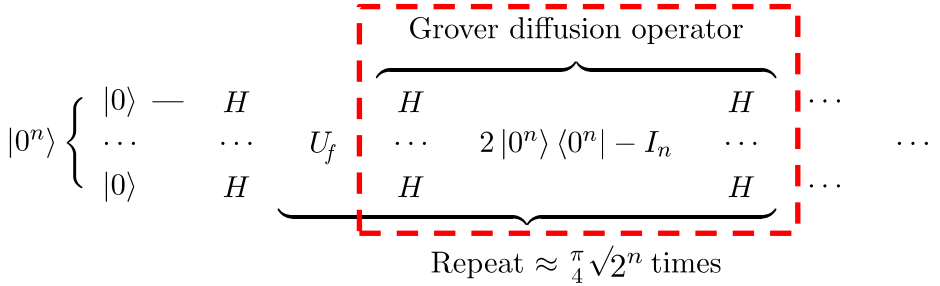
$$|s\rangle = \frac{1}{\sqrt{2^n}}\sum_x |x\rangle$$

$$|s\rangle = \frac{1}{\sqrt{2^n}}\sum_x (-1)^{f(x)}|x\rangle$$

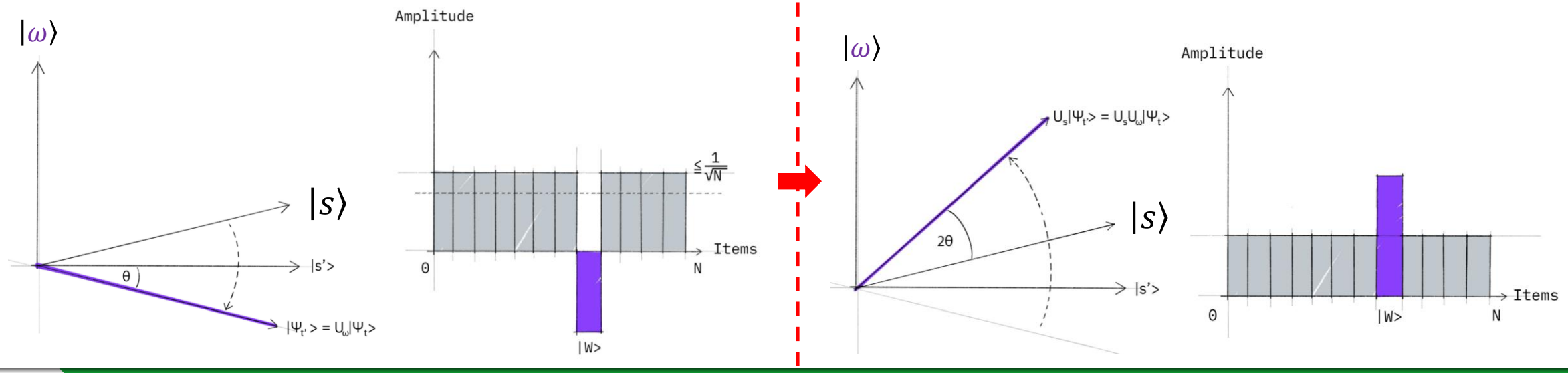
Le moyen des amplitudes diminue

L'algorithme de Grover – détails

Étape 3 : inversion autour du moyen

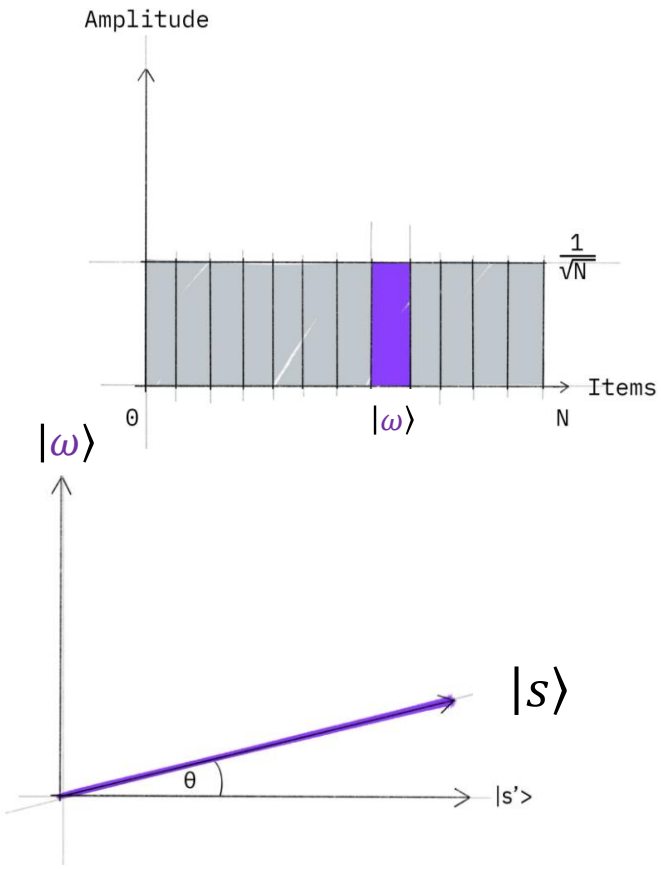


- On réfléchit les amplitudes autour de leur moyen
- Géométriquement, on fait tourner l'état autour de $|s\rangle = \frac{1}{\sqrt{2^n}}\sum_x |x\rangle$

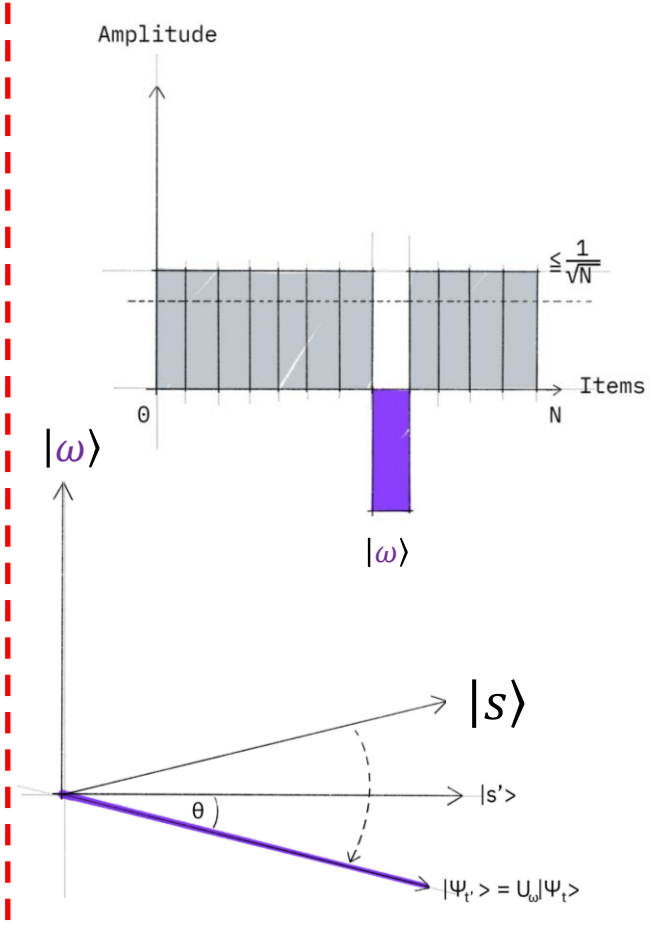


L'algorithme de Grover – résumé d'étapes

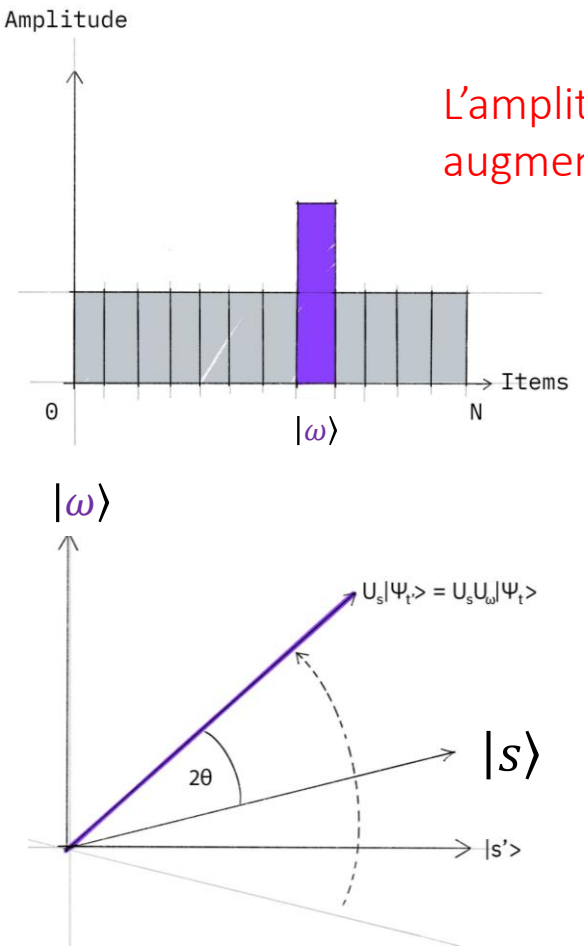
Après étape 1



Après étape 2

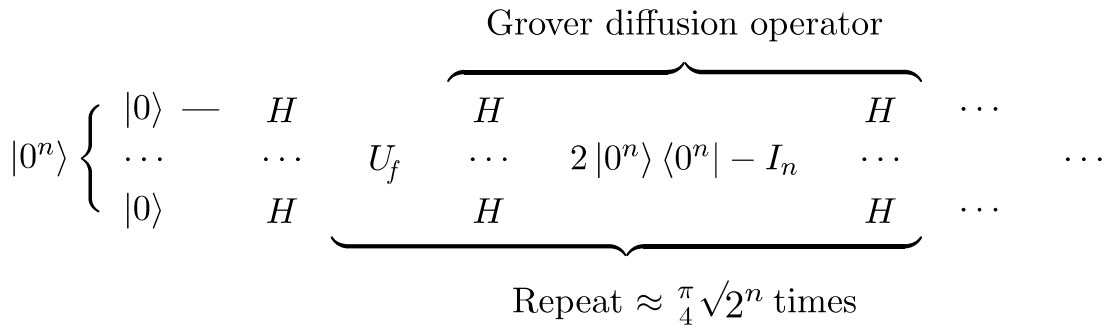


Après étape 3



L'amplitude de $|\omega\rangle$ a augmenté !

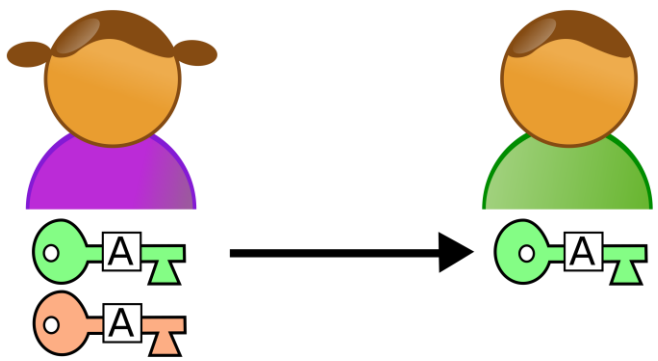
L'algorithme de Grover – bilan

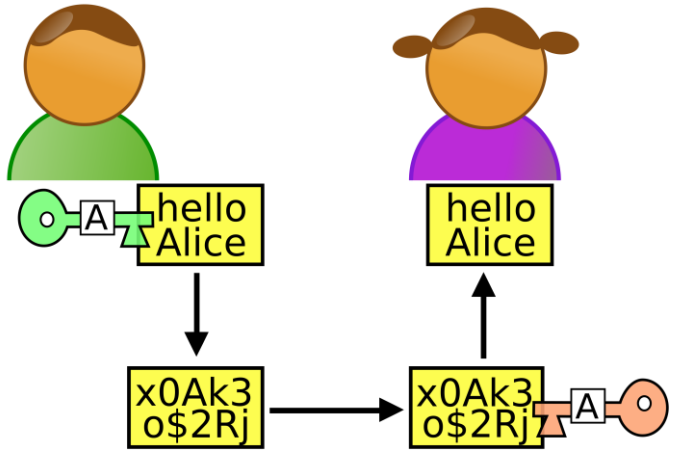


- Après $\frac{\pi}{4}\sqrt{2^n}$ itérations, l'amplitude de $|\omega\rangle$ est (presque) 1
  - On peut simplement mesurer les qubits pour trouver ω
- On évalue f $O(\sqrt{2^n})$ fois, tandis qu'un algorithme classique exige $O(2^n)$
- Applicable à des grandes classes de problèmes, comme le voyageur du commerce
- Est-ce qu'on peut faire mieux qu'un avantage quadratique ?

Cryptographie asymétrique

- La cryptographie à clé publique est omniprésente dans le monde numérique pour sécuriser nos communications et pour l'authentification





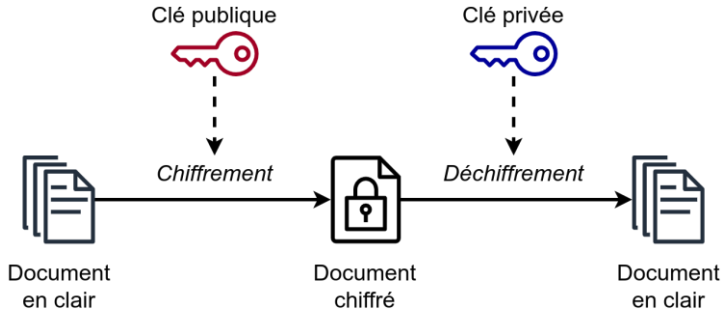
- Deux clés liées sont utilisées
  - Une clé publique, connue à tous, pour chiffrer des messages
  - Une clé privée, gardée secrète, pour déchiffrer les messages

Factorisation de nombres entières

Chiffrement RSA (Rivest-Shamir-Adleman)

- L'un des cryptosystèmes asymétriques le plus utilisé
- Basé sur le problème de décomposition en facteurs premiers

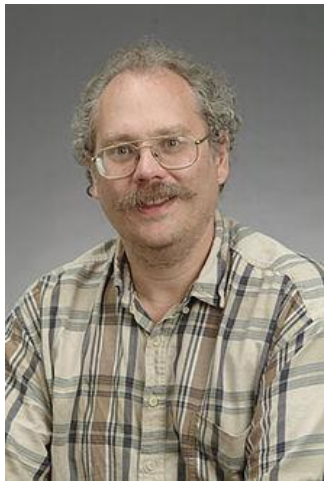
Donné un N tel que $N = pq$, trouver p et q

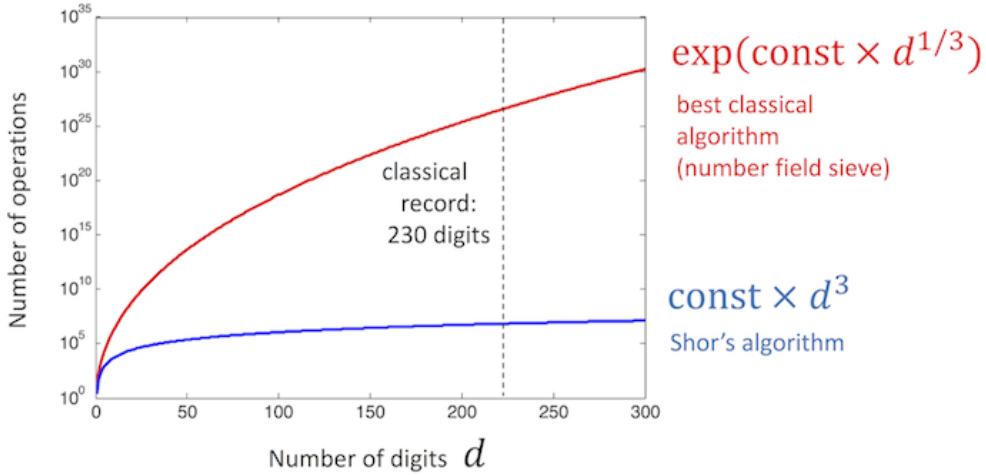


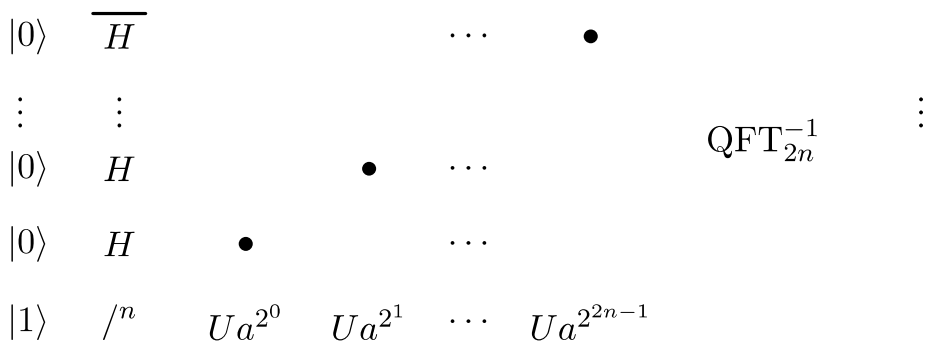
- On le croit un problème difficile à résoudre avec un ordinateur classique
  - Les meilleurs algorithmes classiques nécessitent $O\left(e^{\sqrt[3]{n}}\right)$ opérations pour un nombre N à $n \approx \log_{10} N$ chiffres
  - Longueur de clé typique : 2048 ou 4096 bits

L'algorithme de Shor

- Le problème de factorisation peut être (classiquement) réduit à un problème algébrique :
  - Donné $a < N$, trouver le plus petit r tel que $a^r \equiv 1 \bmod N$
- En 1994, Peter Shor a trouvé un algorithme quantique pour ce problème qui nécessite seulement $O(n^3)$ portes quantiques
- Un avantage exponentiel sur les meilleurs algorithmes classiques

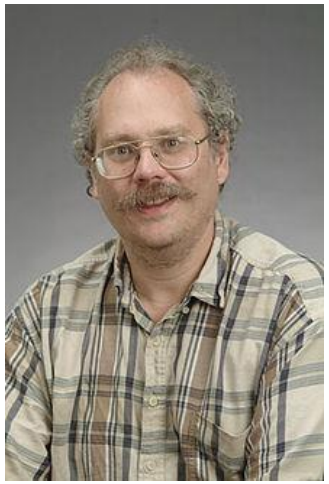


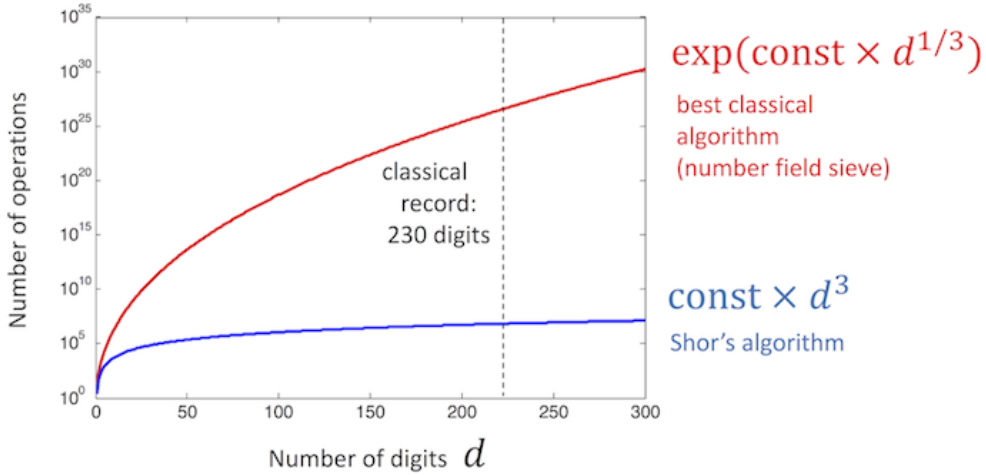




L'algorithme de Shor

- Le problème de factorisation peut être (classiquement) réduit à un problème algébrique :
  - Donné $a < N$, trouver le plus petit r tel que $a^r \equiv 1 \bmod N$
- En 1994, Peter Shor a trouvé un algorithme quantique pour ce problème qui nécessite seulement $O(n^3)$ portes quantiques
- Un avantage exponentiel sur les meilleurs algorithmes classiques





- Au contraire de l'algo de Grover, on n'a pas de preuve qu'on ne pouvait pas faire aussi bien classiquement !
  - C'est difficile à démontrer des avantages quantiques

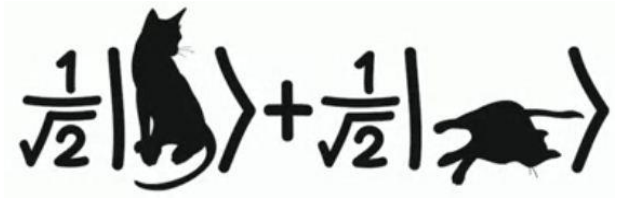
Les implications de l'algorithme de Shor

- Des implications profondes pour la sécurité
  - Un ordinateur quantique suffisamment puissant rendrait les cryptosystèmes d'aujourd'hui obsolètes
    - L'algorithme de Shor casse la plupart des protocoles utilisés à ce jour
  - C'est même un problème pour aujourd'hui
    - Toute communication chiffrée avec RSA pourrait être stockée et déchiffrée dès qu'un ordinateur quantique suffisamment puissant est construit
  - Besoin de cryptosystèmes qui sont sécures contre des adversaires quantiques !

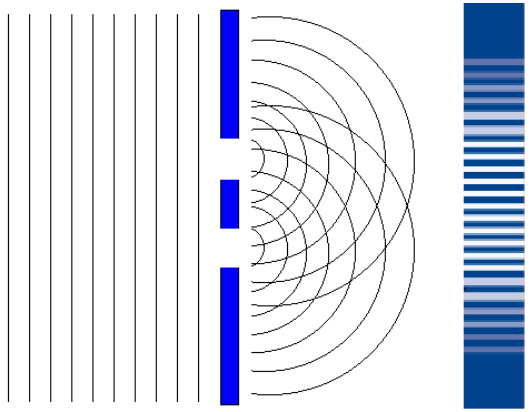
- Et même la sécurité des cryptomonnaies (ex. BitCoin)
  - Signatures peuvent être contrefaites avec l'aide de l'algorithme de Shor
    - Grand problème si fait suffisamment vite avec un ordinateur quantique
  - Des avantages plus modestes (quadratiques) pour obtenir une « preuve de travail »

Résumé : que fait tourner un algorithme quantique ?

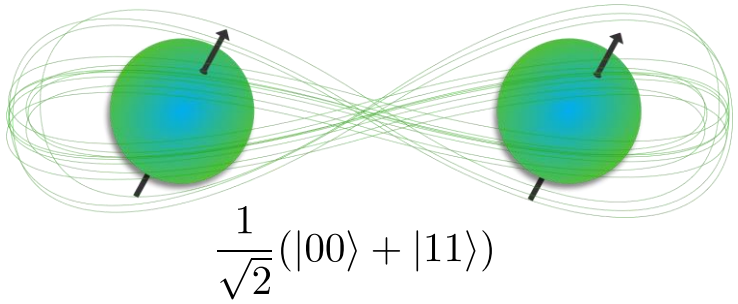
- Un calcul quantique n'est pas plus rapide simplement grâce à un parallélisme quantique
- Interaction délicate entre plusieurs phénomènes quantiques



Superposition (parallélisme)



Interférence



Intrication

Comment créer des algorithmes quantiques ?

- Inventer des nouveaux algorithmes n'est pas facile…
  - Fortes contraintes des mesures, transformations unitaires, réversibles, …
  - Besoin de structure dans le problème pour pouvoir exploiter interférence et superposition
- On n'a pas d'intuition pour « penser en quantique »
  - Exemple : unitarité implique « non-clonage » – l'impossibilité à copier d'information quantique
    - Il n'y a pas de U tel que $U(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |\psi\rangle$ pour tout $|\psi\rangle$
  - Pas de suppression non plus
- Programmation de haut niveau difficile
  - Pas de si clauses, boucles, copier, …
  - Circuits sont très proches à la physique, comme un langage d'assembleur

L'histoire des algorithmes quantiques

4. Problèmes génériques, algorithmes hybrides

QAOA : Quantum Approximate Optimization Algorithm (2014)

VQE : Variational Quantum Eigensolver (2014)

---

3. Problèmes en algèbre linéaire, « big data »

Algorithmes d'apprentissage automatique quantique

L'algorithme de HHL (2009)

---

2. Problèmes réels, structurés

Algorithmes des marches-quantiques

L'algorithme de Grover (1996)

L'algorithme de Shor (1994)

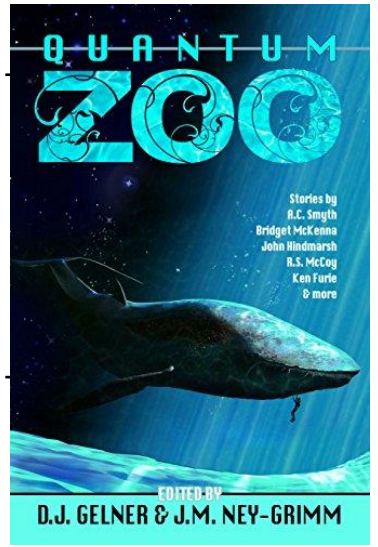
---

1. Phase d'expérimentation

L'algorithme de Simon (1994)

L'algorithme de Deutsch-Jozsa (1992)

L'algorithme de Deutsch (1985)



https://quantumalgorithmzoo.org

Du quantique pour l'IA ?

- Le cœur des algorithmes d'apprentissage automatique repose sur le calcul des opérations matricielles en algèbre linéaire
  - Inversion d'une matrice, calcul des valeurs propres, …

- **HHL (Harrow, Hassidim, Llyod, 2008)**
  - Algorithme quantique pour résoudre un système d'équations linéaires
    - $A\,\vec{x} = \vec{b} \longrightarrow A^{-1}|b\rangle = |x\rangle$
  - Avantage exponentielle si le système d'équations a certaines propriétés (creuse, …)
- Généralisations pour résoudre des équations différentielles, invertir des matrices, …
  - Dans presque tous les cas, seulement sous certaines conditions…

Du quantique pour l'IA ?

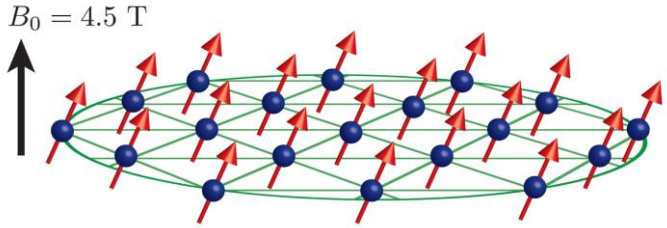
- Est-ce que l'on peut attendre à une future d'IA quantique ?

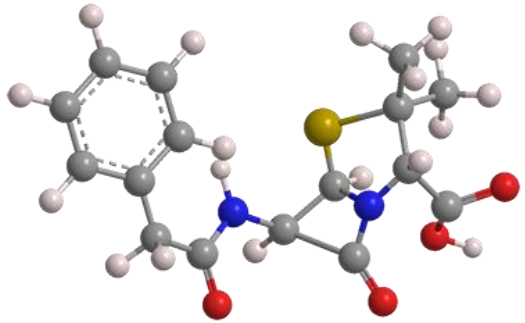
- De nombreux algorithmes proposés, mais :
  - Les avantages théoriques supposent des fortes hypothèses sur la forme des données
  - Les données doivent être préparées sous forme d'états quantiques
  - Ex : Superposition $\frac{1}{\sqrt{N}}\sum_i |\vec{x}_i\rangle$ à partir d'un tableau $(\vec{x}_i)_i$ des données

- Impact potentiel du quantique pour l'IA reste débattu
  - Dans tous les cas : nécessite un très puissant ordinateur quantique

Simulation quantique

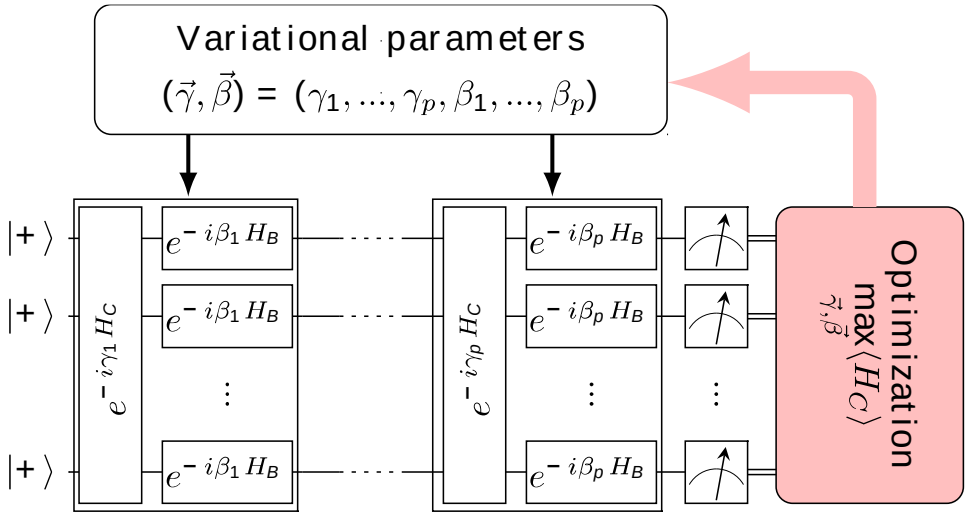
- Utiliser un ordinateur quantique pour des problèmes quantiques ?
  - Rappelez la motivation de Feynman
- Simulation quantique : comment se comporte un système quantique (physique, chimie, …) très compliqué ?
- Problèmes d'optimisation
  - Trouver l'état fondamental d'un système physique
  - Permet de modéliser des molécules et réactions chimiques
  - Repliement des protéines





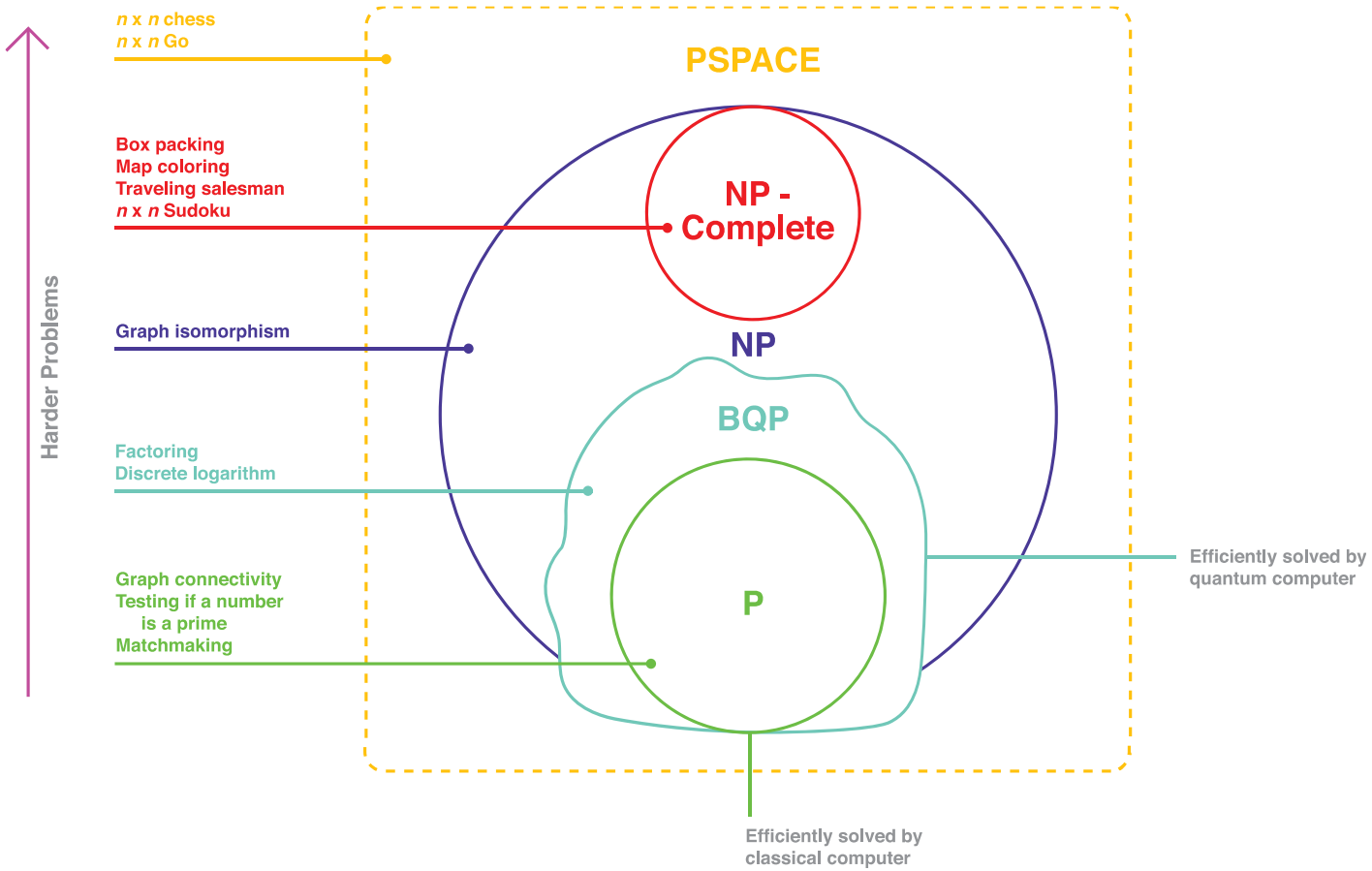
Algorithmes hybrides « classiques - quantiques »

- Ces algorithmes typiquement ont besoin d'un grand ordinateur quantique
  - Qu'est-ce que l'on peut faire avec un petit prototype ?
- Algorithmes hybrides pour des problèmes d'optimisation
  - Ex : Quantum Approximate Optimization Algorithm (QAOA)



- Très générique, mais avantage incertain

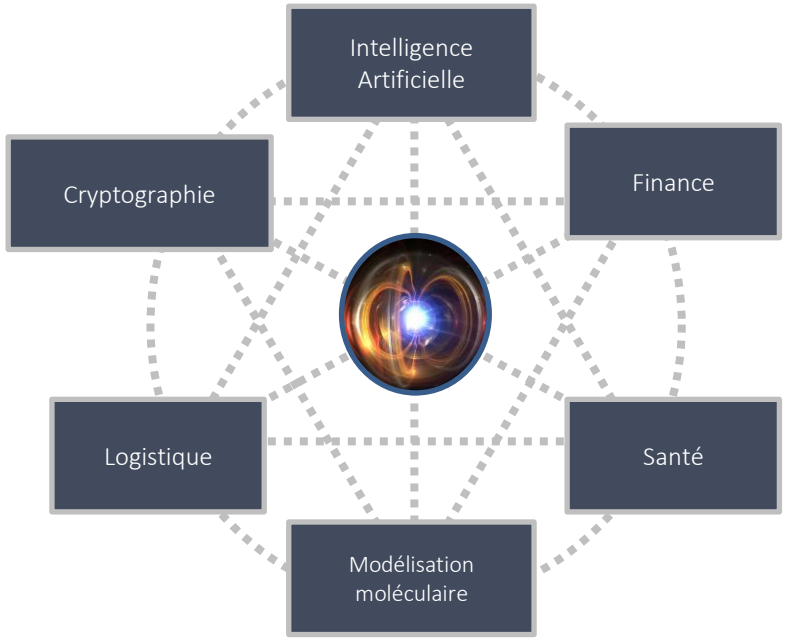
Quelle puissance ?



- Beaucoup d'algorithmes quantique n'ont que des avantages supposés
- Le quantique semble aider avec certains problèmes, mais pas tous !

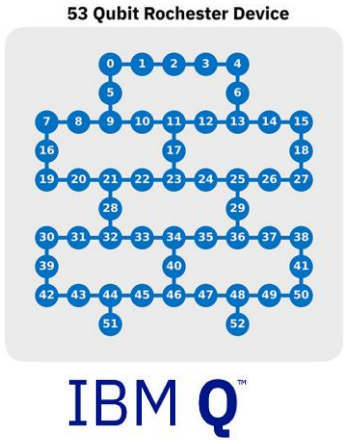
Résumé : algorithmes quantiques

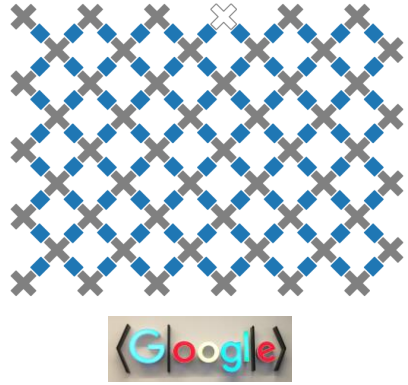
- Algorithmes quantiques applicables à de nombreux domaines
  - Grande potentielle pour révolutionner le calcul
  - Mais pas mal d'incertitude sur les avantages théoriques et pratiques de certains algorithmes ou dans certaines tâches



Epilogue : vers le calcul quantique en pratique

- Les ordinateurs quantiques qu'on commence à construire sont bruités, avec des architectures restreintes
  - Comment compiler et optimiser un algorithme ou un circuit pour une architecture donnée ?
- Qubits sont analogiques, très susceptible aux erreurs
  - L'intrication : une erreur sur un qubit peut avoir des effets partout…
- Comment corriger ces erreurs et assurer un calcul fiable ?
  - À voir demain !





Conclusions à retenir

- Le calcul quantique exploite des effets quantiques d'une manière subtile
  - Pas simplement une machine massivement parallèle, ni un « super IA »
- L'informatique quantique a beaucoup de contraintes (mesures destructives et probabilistes, portes linéaires et réversibles, impossibilité de copier et effacer d'information, …)
- Il faut prendre soin en construisant un algorithme quantique, exploiter de l'interférence quantique et la structure du problème
- Le calcul quantique a le potentiel d'offrir des avantages en calcul (quadratique et dès fois même exponentiel)

Quelques ressources

- Livres
  - O. Ezratty: *Comprendre l'Informatique Quantique* (https://www.oezratty.net/wordpress/2020/comprendre-informatique-quantique-edition-2020/)
  - M. Nielsen, I. Chuang: *Quantum Computation and Quantum Information* (http://mmrc.amss.cas.cn/tlb/201702/W020170224608149940643.pdf)
  - R. de Wolf: *Quantum Computing: Lecture Notes* (https://homepages.cwi.nl/~rdewolf/qcnotes.pdf)
- Programmation
  - Qiskit, Quantum Computing Hanbook (https://qiskit.org/textbook/preface.html)
- Vidéos
  - F. Magniez, cours au Collège de France (https://www.college-de-france.fr/site/frederic-magniez/course-2020-2021.htm)