COLLÈGE DE FRANCE
1530

Algorithmes quantiques

Circuits quantiques, premiers algorithmes

05-05-2021

Frédéric Magniez

Professeur invité sur la chaire Informatique et sciences numériques

En partenariat avec Inria

Année académique 2020-2021

frederic.magniez@college-de-france.fr

Partie 2 - Les bases algorithmiques

- Concepts du calcul et principales méthodes algorithmiques
- Mise en évidence de propriétés algébriques (déchiffrement)
- Optimisation et applications algorithmiques

05 mai 2021

Cours : Circuits quantiques, premiers algorithmes : portes universelles, algorithmes de Deutsch-Jozsa et Bernstein-Vazirani, supériorité des algorithmes quantiques

Séminaire : Langages graphiques pour programmer et raisonner en informatique quantique
Simon PERDRIX, *CNRS, Nancy*

12 mai 2021

Cours : Transformée de Fourier quantique I : réalisation, estimation de phase, algorithmes de Simon et de Shor (recherche de période et factorisation) et généralisations récentes

Séminaire : Le problème du sous-groupe caché, Miklos SANTHA, *CNRS, Paris et CQT, Singapour*

19 mai 2021

Cours : Optimisation quantique : algorithme de Grover, estimateurs quantiques, chaînes de Markov quantiques, heuristiques quantiques

Séminaire : A Unified Framework for Quantum Walk Search, Stacey JEFFERY, *CWI, Amsterdam*

Récapitulatif des notations sur la page du cours

Superposition : état, mesure, transformation unitaire

Superposition

- S : ensemble d'états classiques

  S={0,1}, S={0,1}ⁿ, S={1,…,N}

- Superposition sur S

  Vecteur à |S| coordonnées complexes $(\alpha_x)_{x\in S}$

- Notation Dirac

  $$|\psi\rangle = \sum_{x\in S} \alpha_x |x\rangle, \quad \alpha_x = \langle x|\psi\rangle$$

Mesures

- Base de calcul

  $$\sum_{x\in S} \alpha_x |x\rangle \longrightarrow \boxed{\text{Mesure}} \xrightarrow{|\alpha_x|^2 \text{ “x”}} |x\rangle$$

- Base orthonormée $(|\psi_x\rangle)_{x\in S}$

  $$|\psi\rangle \longrightarrow \boxed{\text{Mesure } (|\psi_x\rangle)_{x\in S}} \xrightarrow{|\langle\psi_x|\psi\rangle|^2 \text{ “x”}} |\psi_x\rangle$$

Transformations unitaires

- Rotations et symétries G

  $$|x\rangle \dashrightarrow \boxed{G} \dashrightarrow |\psi_x\rangle$$

  $$\sum_{x\in S} \alpha_x |x\rangle \dashrightarrow \boxed{G} \dashrightarrow \sum_{x\in S} \alpha_x |\psi_x\rangle$$

  avec $(|\psi_x\rangle)_{x\in S}$ base orthonormée

- Notation matricielle

  $$G = \big(|\psi_x\rangle\big)_{x\in S} = \big(\langle x|\psi_y\rangle\big)_{x,y\in S}$$

  avec G unitaire : $G^* = ({}^t\overline{G}) = G^{-1}$

  $$|\psi\rangle \dashrightarrow \boxed{G} \dashrightarrow G|\psi\rangle$$

Registres multiples 1/2 : produit tensoriel

Vision multi-registre

- A,B : ensembles d'états classiques
- S=AxB={(x,y): x∈A,y∈B} ensemble des états combinés
- Superposition sur AxB

  Vecteur $(\gamma_{x,y})_{x\in A, y\in B}$

- Notation Dirac

  $$|\psi\rangle = \sum_{x\in A, y\in B} \gamma_{x,y} |x,y\rangle$$

Vision équivalente !

Vision produit tensoriel

- Superposition sur A

  Vecteur $(\alpha_x)_{x\in A}$

  $$|\psi_1\rangle_A = \sum_{x\in A} \alpha_x |x\rangle_A$$

- Superposition sur B

  Vecteur $(\beta_y)_{y\in B}$

  $$|\psi_2\rangle_B = \sum_{y\in B} \beta_y |y\rangle_B$$

- Superposition produit sur AxB

  $$|\psi_1\rangle_A \otimes |\psi_2\rangle_B = \sum_{x\in A, y\in B} \alpha_x \beta_y |x\rangle_A \otimes |y\rangle_B$$

- Superposition générale

  $$|\psi\rangle_{AB} = \sum_{x\in A, y\in B} \gamma_{x,y} |x\rangle_A \otimes |y\rangle_B$$

  Si non produit, alors enchevêtrée (ou intriquée)

Registres multiples 2/2 : transformations partielles

Transformations unitaires

- G_A une transformation unitaire sur les superpositions de A
- G_B une transformation unitaire sur les superpositions de B
- $G_A \otimes G_B$ est défini sur les états séparés

  $$(G_A \otimes G_B)\big(|\psi_1\rangle_A \otimes |\psi_2\rangle_B\big) = G_A\big(|\psi_1\rangle_A\big) \otimes G_B\big(|\psi_2\rangle_B\big)$$

  et étendu par linéarité

  $$(G_A \otimes G_B)\Big(\sum_{xy} \gamma_{xy} |x,y\rangle_{AB}\Big) = \sum_{xy} \gamma_{xy} G_A\big(|x\rangle_A\big) \otimes G_B\big(|y\rangle_B\big)$$

Mesures partielles (sur A)

- Se ramener à une décomposition sur la base de A

  $$\sum_{x\in A} \alpha_x |x\rangle_A |\psi_x\rangle_B \longrightarrow \boxed{\text{Mesure A}} \xrightarrow{|\alpha_x|^2 \text{ “x”}} |x\rangle_A |\psi_x\rangle_B$$

Remarques

- Faire une opération G_A sur A, revient à faire l'opération $G_A \otimes Id_B$ sur AxB
- Les transformations G_A sur A commutent avec les opérations G_B sur B
- Mesurer sur A puis B, est identique à tout mesurer

La genèse du calcul

Modèles : théorie et technologie liées

- Machine de Turing, lambda calcul [Church 1936, Turing 1936]
  - Calculabilité, <u>Universalité</u> (programme en mémoire)
  - Accès séquentiel à la mémoire
- Circuits logiques [Shannon 1949]
  - Adaptés au calcul parallèle, plus proches des circuits imprimés
- Machines RAM (random-access machine) [Shepherdson, Sturgis 1963]
  - Adaptées aux accès directs à la mémoire
- …, calcul parallèle, réseaux de neurones…
- <u>Tous ces modèles de calcul se simulent les uns les autres de façon *efficace*</u>
  - *Thèse de Church-Turing : version quantitative/algorithmique*

Langages

- Assembleur (1949)
- Fortran (1957)
- Cobol (1959), Algol (1960), Lisp (1960)
- …
- 1980-90 : C++, Java, OCaml…

Question de Richard Feynman

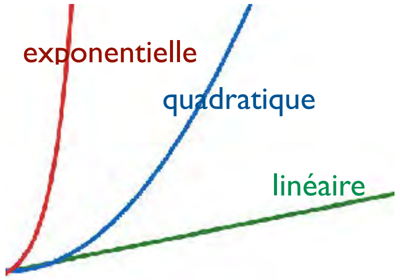
- Exposé invité où il questionne :

  Can quantum systems
  be probabilistically simulated
  by a classical computer?

Argumentations

- *[…] Quantum mechanics can't seem to be imitable by a local classical computer.*

  → Explosion des ressources nécessaires



- *Can you do it with a new kind of computer - a quantum computer? […] It's not a Turing machine, but a machine of a different kind. […] I'm not sure that it's sufficient.*

  → Remise en cause de la version quantitative
  de la thèse de Church-Turing

Construction théorique

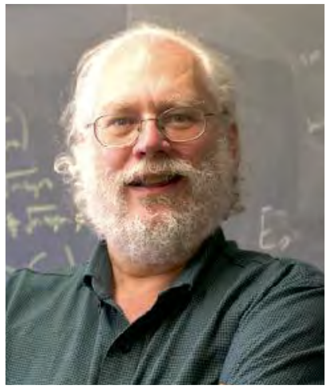
- David Deutsch, 1985
    - Machine de Turing quantique, premiers algorithmes surprenants
- Ethan Bernstein et Umesh Vazirani, 1993
    - Machine de Turing quantique universelle et efficace
    - Des avantages exponentiels, mais pour des problèmes artificiels


David Deutsch


Umesh Vazirani

Le développement algorithmique

- Succès : Daniel Simon puis Peter Shor, 1994
    - Remise en cause d’une grande partie des techniques de chiffrement massivement utilisées
- Inquiétude : Instabilité du calcul
    - Réponse : Codes correcteurs quantiques, Peter Shor 1995


Peter Shor

Réponse à la question de Feynman ?

- Seth Lloyd 1996 : Simulation hamiltonienne
- Caractère universel
- Une panoplie de plus en plus grande et variée d’algorithmes depuis…

Modèles et technologiques quantiques

Modèles

- Machine de Turing quantique [Deutsch 1985], [Bernstein, Vazirani 1993]
- Circuits quantiques [Yao 1993]
- Calcul par mesure [Raussendorf, Briegel 2001]
- Calcul adiabatique quantique [Farhi, Goldstone, Gutmann, Sipser 2000]
- Calcul QRAM [Conventions for quantum pseudocode - Knill'96]

Technologies

- Les débuts : 2 qubits [ENS (Haroche) 1995], 5 qubits [IBM (Chuang) 2000]
- En 2020/21 : 50-100 qubits

  Superconducteurs : Alibaba, Google, IBM, Intel, Rigetti…

  Trappes à ions : IonQ / à atomes de Rydberg : Pasqal
- Des prévisions optimistes

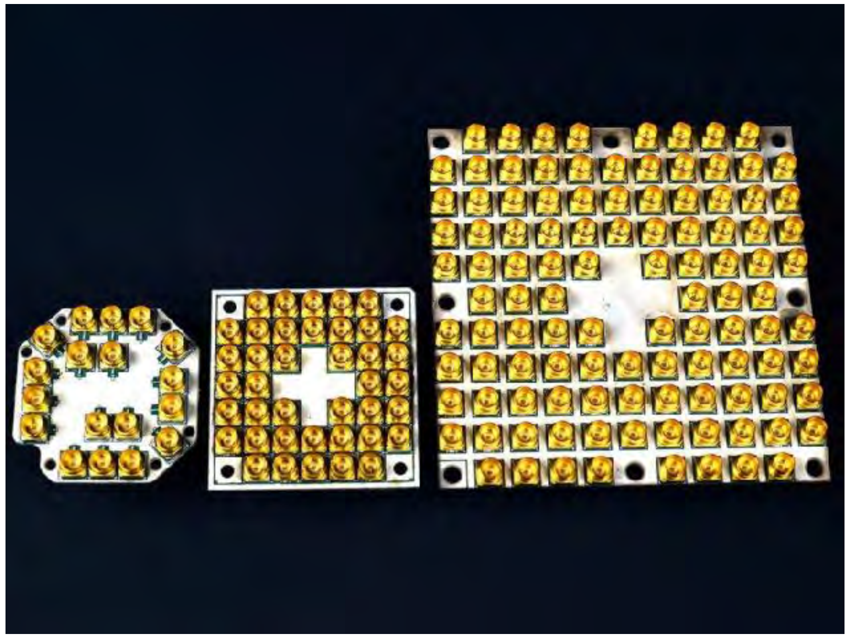
  Google & IBM : 1 000 qubits en 2023, 1 million vers 2030

Langages

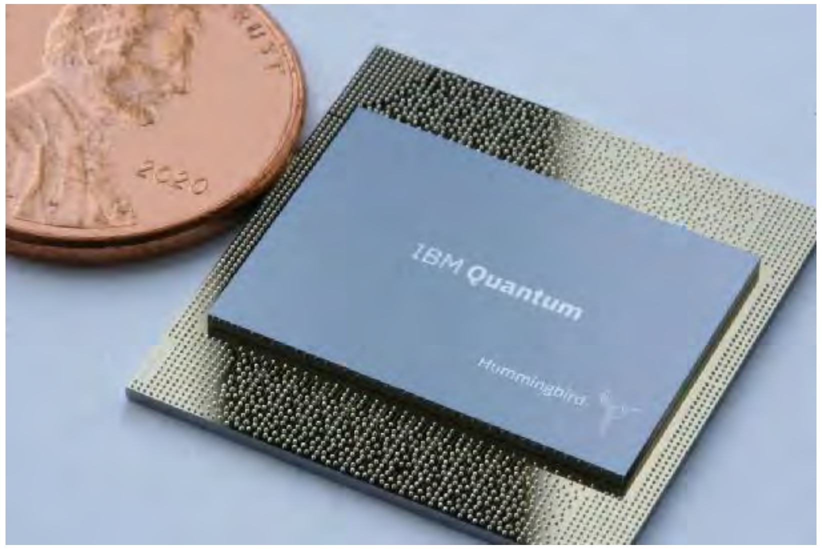
- Cirq (Google), Q# (Microsoft), Qiskit (IBM) (https://qiskit.org), …
- Des langages fonctionnels : Liquid, Quipper, …
- Emulation ou connection dans le cloud !

Les circuits

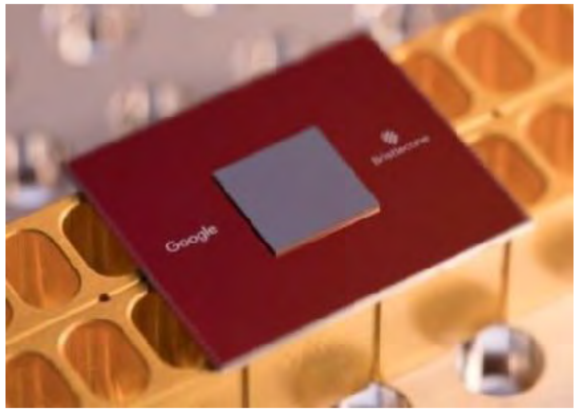
Circuits quantiques ?

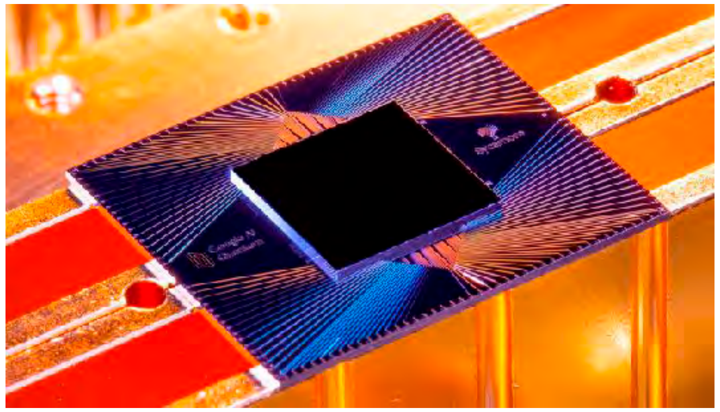


January 2018: Intel's 49-Qubit Chip Shoots for Quantum Supremacy



December 2020: An IBM Quantum Hummingbird r2 Processor (this device has 65 qubits, versus the r1's 53 qubits).





March 2018: Bristlecone, first Google's Quantum Processor with 72-qubit

October 2019: New 54-qubit processor, named "Sycamore", that is comprised of fast, high-fidelity quantum logic gates.

Mathématiques

- Analogues des circuits logiques
- Universels pour le calcul
- Adaptés pour décrire des transformations complexes

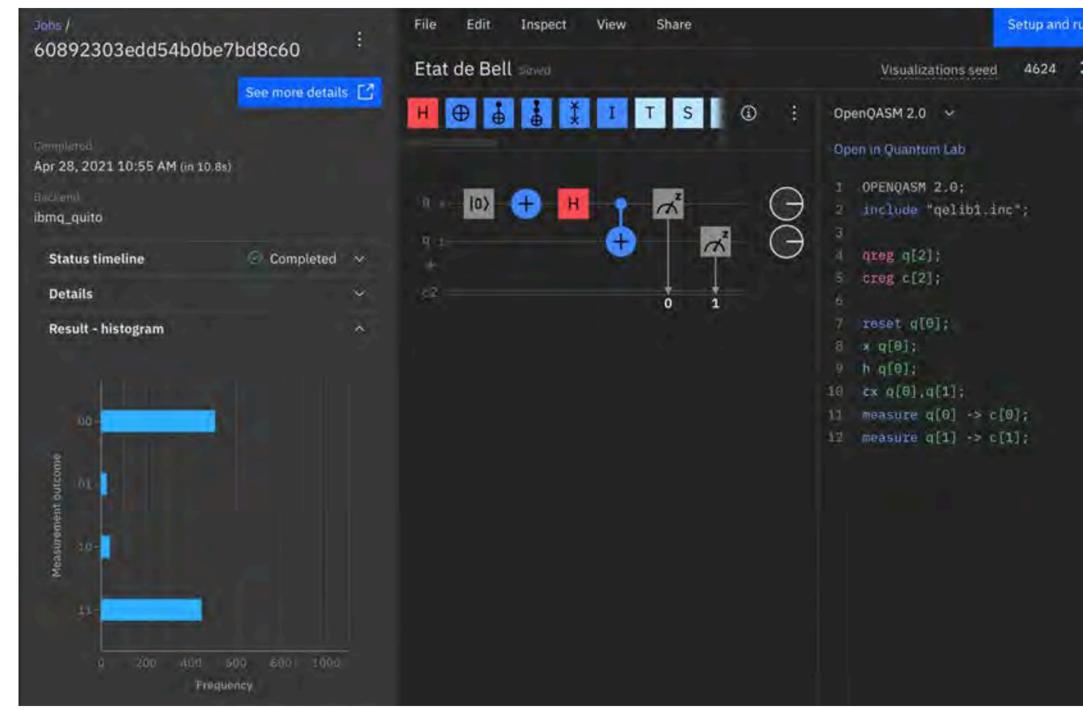
Informatiques

- Adaptés pour l'apprentissage

  https://quantum-computing.ibm.com

- Adaptés pour la programmation

  https://qiskit.org



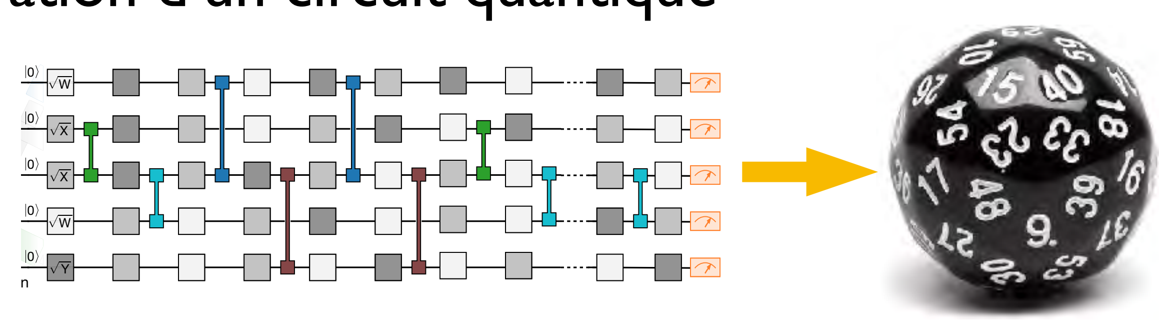
Technologiques

- Proches de certaines expériences

  Dans le cloud ou de "suprématie quantique"

- L'expérience de Google - 2019

  Statistiques provenant de l'observation d'un circuit quantique

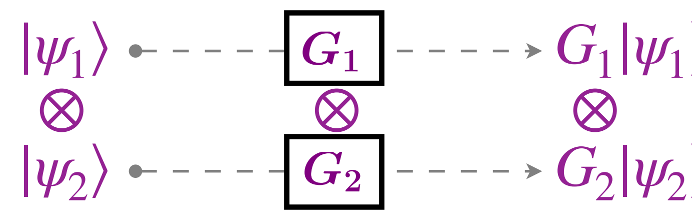


Portes

- Une porte quantique est une transformation unitaire sur au plus 3 qubits

  $G \in \mathcal{U}(2^k), \quad k = 1,2,3$

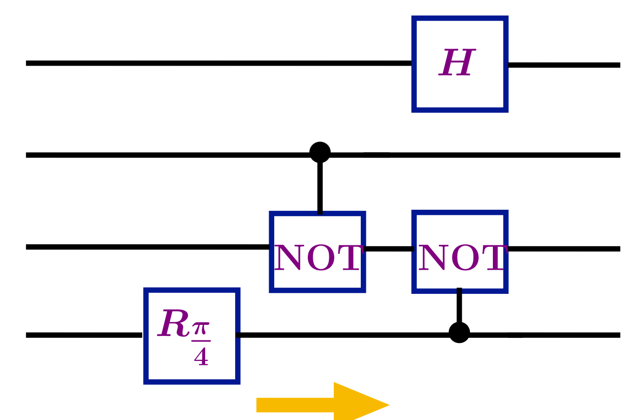
Composition et produit tensoriel



Circuit

- Un circuit quantique

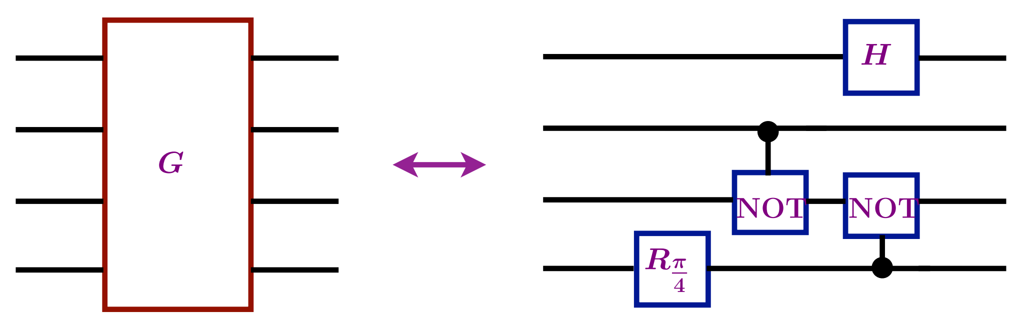
  est la composition de portes (étendues par ⊗ Id)
- Complexités : taille (nb de portes) et profondeur



Sens de composition

Théorème

- Toute transformation unitaire peut se décomposer en un circuit de portes agissant sur au plus 2 qubits



Théorème

- La porte c-NOT et toutes les portes sur 1-qubit permettent de réaliser toute transformation unitaires (avec au plus $O(n4^n)$ portes et des qubits auxiliaires)

Remarque

- Famille finie de porte → Relâcher la notion d'universalité

Théorème

- Les familles suivantes permettent d'approcher toute transformation unitaire

  H, $R_{\pi/4}$ et c-NOT (base tolérante aux erreurs)

  NOT, $\sqrt{H}$ et c-NOT

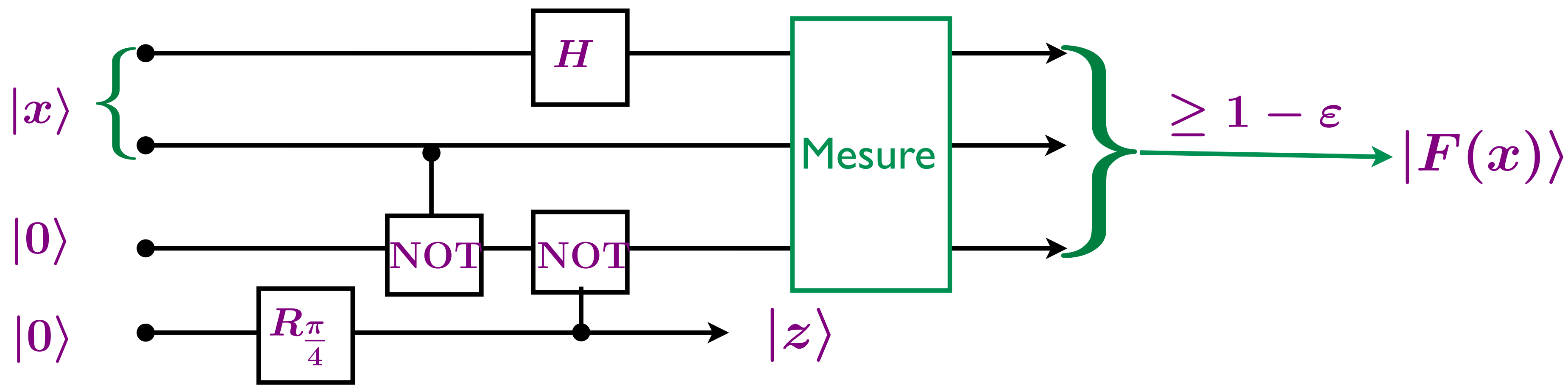
  NOT, H et Toffoli (c-c-NOT) (transformations réelles uniquement)

- Le nombre de portes nécessaire ne dépend pas de la famille universelle (à une constante multiplicative près)
- La plupart des transformations unitaires nécessitent $\Omega(2^n \log(1/\varepsilon)/\log n)$ portes pour être approchées à distance ε (norme d'opérateur)

Définition

- Un circuit $U=U_L U_{L-1}\ldots U_2 U_1$ calcule une fonction f avec erreur $\boldsymbol{\varepsilon}$ si

  pour toute entrée x : $\sum_z |\langle F(x), z|U|x,0^k\rangle|^2 \geq 1-\varepsilon$



- La taille d'un circuit est le nombre de portes utilisées pour le réaliser
- La complexité approchée (resp. exacte) d'une fonction est la taille minimale du circuit qui la calcule avec erreur **1/3** (resp. 0)

Remarques

- La complexité d'une fonction ne dépend pas du choix de base universelle
- L'erreur peut arbitrairement être réduite à $\boldsymbol{\varepsilon}$ par $\log(1/\varepsilon)$ itérations suivies d'un vote majoritaire (classique)
- Peut s'étendre à des systèmes non binaires

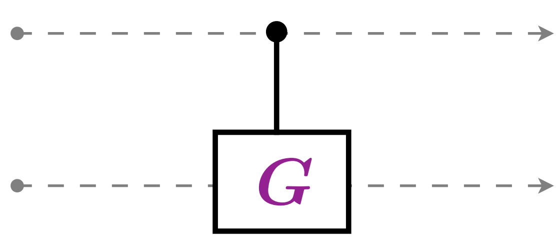
Théorème : Mesure retardée

- Une fonction calculable par un circuit avec des mesures intermédiaires l'est aussi par un circuit comparable avec uniquement une mesure à la fin.

    Au prix de garder ce registre quantique…

Outil : Le "IF THEN ELSE" quantique

- Opération contrôlée



$$\mathbf{c}\text{–}G|0b\rangle = |0b\rangle$$

$$\mathbf{c}\text{–}G|1b\rangle = |1\rangle G|b\rangle$$

Théorème : Mesure implicite

- Inversement, tout registre qui n'est plus utilisé autrement que pour contrôler des opérations peut être supposé observé

Premiers algorithmes quantiques

Problème à Oracle

- L'oracle possède

  $f : \{0,1\} \to \{0,1\}$
- Question possible

  Que vaut $f(0)$? $f(1)$?
- Tâche

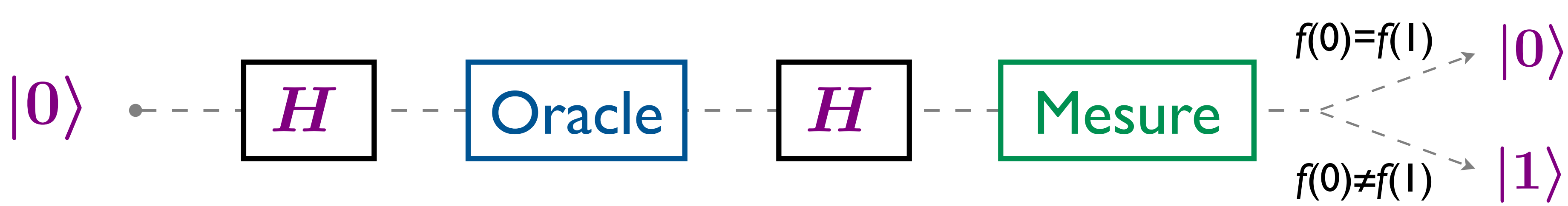
  Décider si $f(0)=f(1)$

Classiquement

- 2 questions sont nécessaires

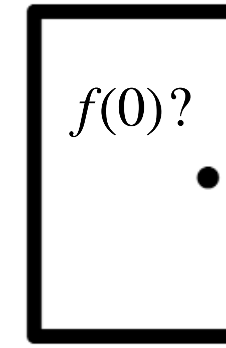
Quantiquement

- 1 seule question en superposition suffit
- Solution :

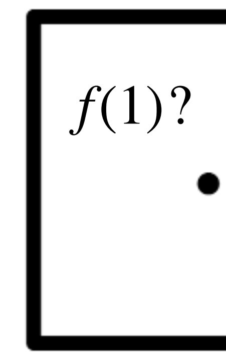


Prêtresse de Delphes
John Collier, 1891

Porte 0



Porte 1



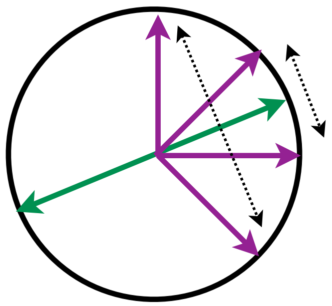
$x \longmapsto f(x)$ peut ne pas être réversible !

Implémentation quantique de *f*

$$|b\rangle \dashrightarrow \boxed{S_f} \dashrightarrow (-1)^{f(b)}|b\rangle$$

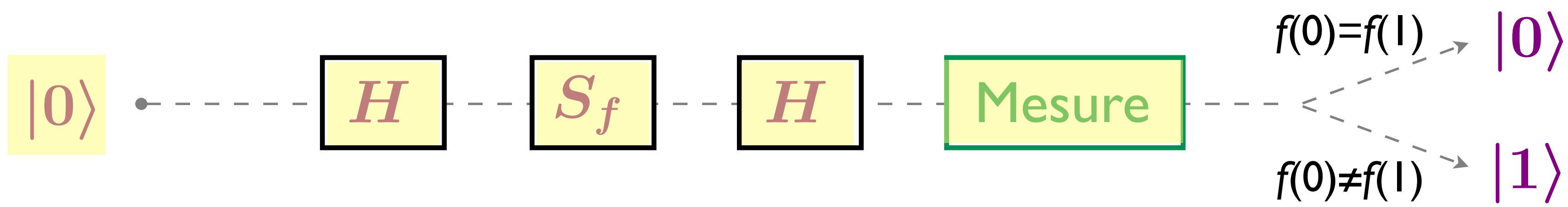
Porte de Hadamard

$$|b\rangle \dashrightarrow \boxed{H} \dashrightarrow \frac{1}{\sqrt{2}}(|0\rangle + (-1)^{b}|1\rangle)$$



Circuit quantique

$$|0\rangle \dashrightarrow \boxed{H} \dashrightarrow \boxed{S_f} \dashrightarrow \boxed{H} \dashrightarrow \boxed{\text{Mesure}} \dashrightarrow$$



Initialisation : $|0\rangle$

Parallélisation : $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

Evaluation de f : $\frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)$

Interférences : $\frac{1}{2}((-1)^{f(0)}(|0\rangle + |1\rangle) + (-1)^{f(1)}(|0\rangle - |1\rangle))$

Etat final : $\frac{1}{2}(((-1)^{f(0)} + (-1)^{f(1)})|0\rangle + ((-1)^{f(0)} - (-1)^{f(1)})|1\rangle)$

Problème de Deutsch-Jozsa 1992

Problème à Oracle

- L'oracle possède

  $f : \{0,1\}^n \to \{0,1\}$ constante ou équilibrée

- Question possible

  Que vaut f(x) ?

- Tâche

  Décider si f est constante ou équilibrée

Solutions

- $1 + 2^{n-1}$ questions classiques sont nécessaires
- 1 question en superposition suffit

  mais il faut *n* qubits…

- Mais $\log(1/\varepsilon)$ questions probabilistes suffisent pour garantir une erreur d'au plus ε

Prêtresse de Delphes
John Collier, 1891

exemple : $n = 4$

$f(0000) \stackrel{?}{=} 0$
$f(0001) \stackrel{?}{=} 0$
$f(0010) \stackrel{?}{=} 0/1$
$f(0011) \stackrel{?}{=} 0$
$f(0100) \stackrel{?}{=} 0/1$
$f(0101) \stackrel{?}{=} 0/1$
$f(0110) \stackrel{?}{=} 0/1$
$f(0111) \stackrel{?}{=} 0$
$f(1000) \stackrel{?}{=} 0$
$f(1001) \stackrel{?}{=} 0/1$
$f(1010) \stackrel{?}{=} 0$
$f(1011) \stackrel{?}{=} 0/1$
$f(1100) \stackrel{?}{=} 0/1$
$f(1101) \stackrel{?}{=} 0$
$f(1110) \stackrel{?}{=} 0/1$
$f(1111) \stackrel{?}{=} 0$

Implémentation quantique de *f*

$$|x\rangle \longrightarrow \boxed{S_f} \longrightarrow (-1)^{f(x)}|x\rangle$$

Transformée de Fourier quantique

$$QFT_n \equiv \begin{array}{c} \longrightarrow \boxed{H} \longrightarrow \\ \longrightarrow \boxed{H} \longrightarrow \\ \vdots \\ \longrightarrow \boxed{H} \longrightarrow \end{array}$$

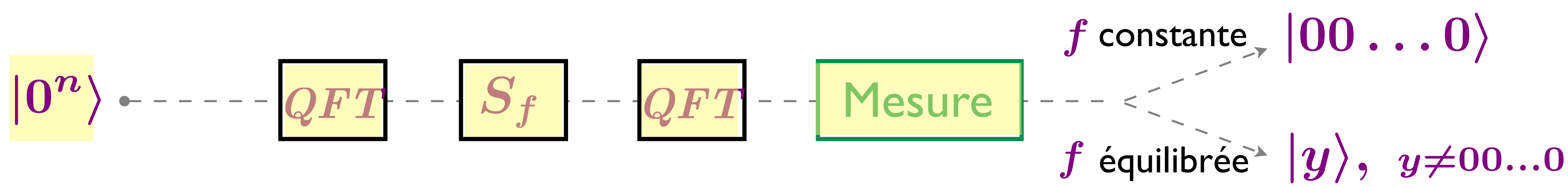
$$|b\rangle \longrightarrow \boxed{H} \longrightarrow \tfrac{1}{\sqrt{2}}(|0\rangle + (-1)^b|1\rangle)$$

$$QFT_n|x\rangle = \frac{1}{2^{n/2}} \sum_y (-1)^{x \cdot y}|y\rangle$$

avec $x \cdot y = \sum_i x_i y_i \mod 2$

Circuit quantique

$$|0^n\rangle \longrightarrow \boxed{QFT} \longrightarrow \boxed{S_f} \longrightarrow \boxed{QFT} \longrightarrow \boxed{\text{Mesure}} \longrightarrow ?$$



Initialisation : $|00\ldots0\rangle$

Parallélisation : $\frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} |x\rangle$

Evaluation de f : $\frac{1}{2^{n/2}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle$

Interférences : $\frac{1}{2^n} \sum_{x,y\in\{0,1\}^n} (-1)^{f(x)+x\cdot y}|y\rangle$

Etat final : $\left(\frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}\right)|00\ldots0\rangle + \sum_{y\neq 00\ldots0} \alpha_y|y\rangle$

Problème

- Oracle : $f : \{0,1\}^n \to \{0,1\}$ une fonction

  telle que $f(x) = a \cdot x = a_1 x_1 \oplus a_2 x_2 \oplus \ldots \oplus a_n x_n$

  pour une valeur $a \in \{0,1\}^n$ fixée mais inconnue

- Sortie : a

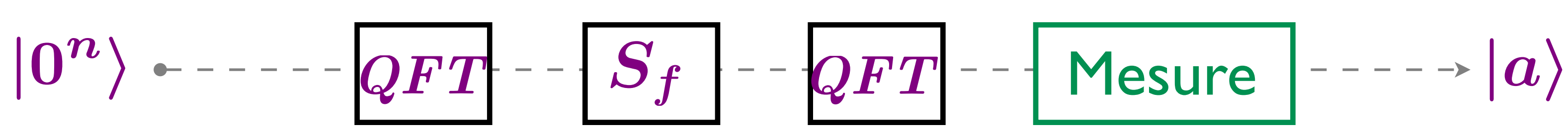
Nombre d'évaluations de *f*

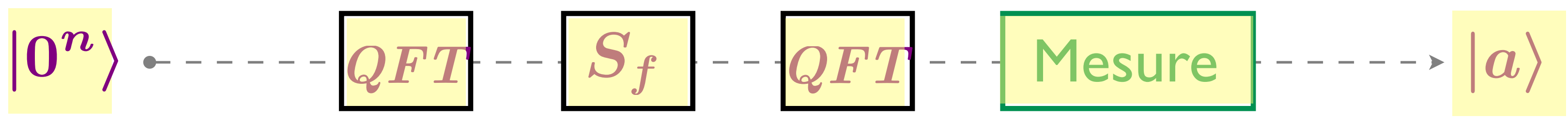
- Probabiliste : n

  Requêtes $f(0^{i-1}10^{n-i})=a_i$, pour $i=1,2,\ldots,n$

- Quantique : 1

Circuit quantique





Initialisation : $|00\ldots 0\rangle$

Parallélisation : $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} |x\rangle$

Evaluation de f : $\frac{1}{2^{n/2}} \sum_{x \in \{0,1\}^n} (-1)^{a \cdot x} |x\rangle = QFT|a\rangle$

Interférences : $QFT^2|a\rangle$

Etat final : $|a\rangle$

Du classique au quantique

Transformation

- Transformer le calcul classique en circuit logique réversible

  AND, NOT, OR sont universels pour le calcul logique

  Toffoli = c-c-NOT est universel pour le calcul logique réversible
- Un circuit réversible est aussi un circuit quantique

  (puisqu'il permute les états "classiques")

Problèmes

- Cas des calculs probabilistes

  Simuler l'aléa avec des qubits $H|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$
- Quel est le coût de la transformation ?

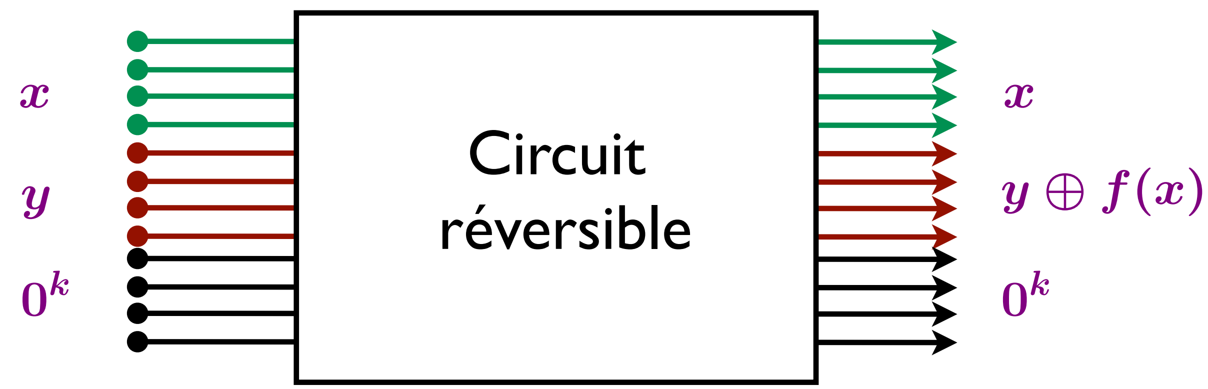
  Si une fonction *f* peut être calculée par un circuit logique de taille *L* et de profondeur *d*, alors *f* peut aussi l'être par un circuit réversible de taille O(*L*) et de profondeur O(*d*)

  <u>Mais avec O(*L*) bits supplémentaires</u>

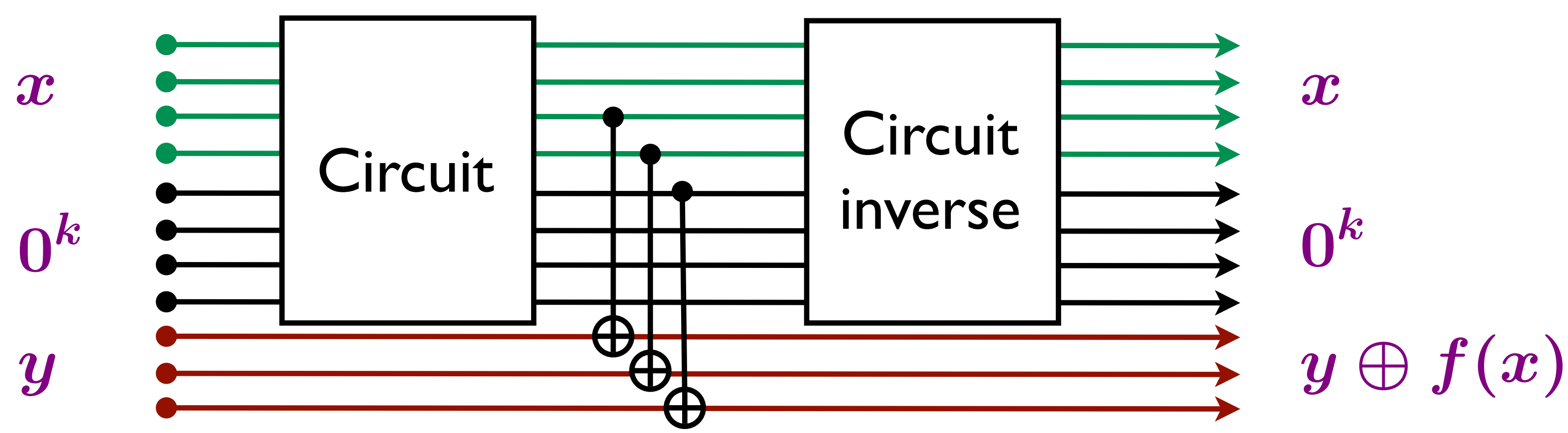
  Peut être optimisé [Bennett 1973]
- Forme normale : réinitialiser les bits supplémentaires utilisés

Théorème

- Dans le théorème précédent on peut demander que le circuit soit en forme normale :



Preuve



Remarque

- Il en est de même pour tout calcul quantique !

Deux formes normales

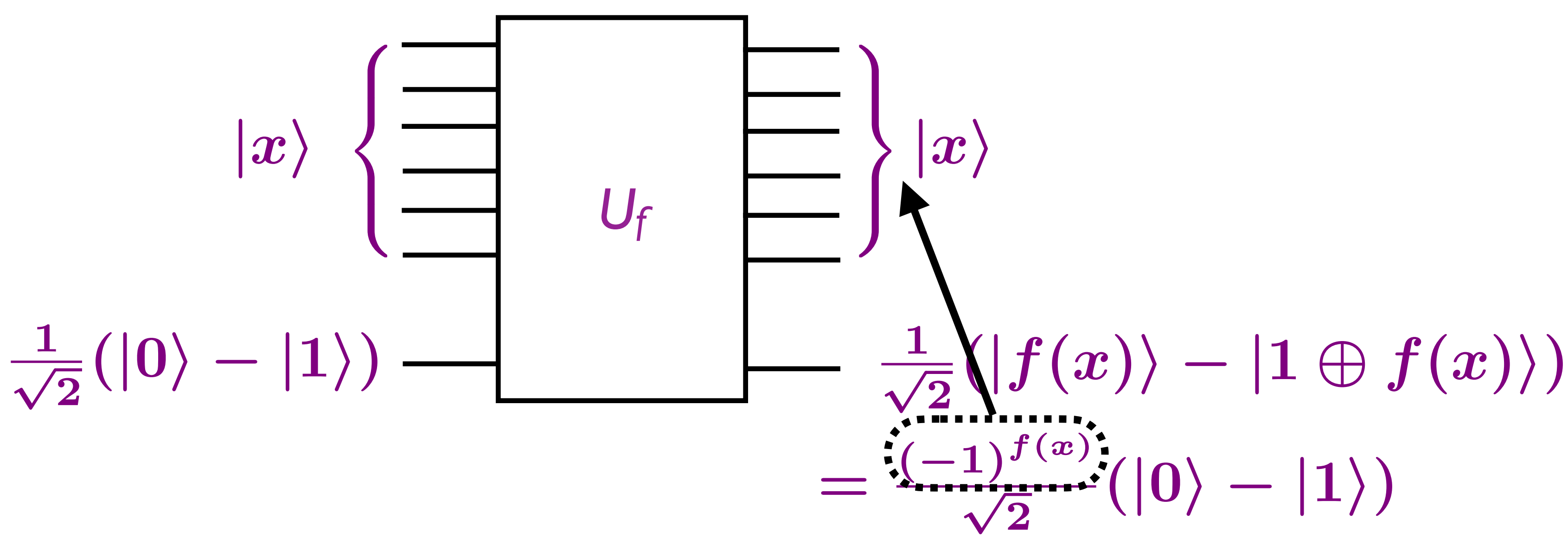
- Fonction booléenne : $f : \{0,1\}^n \longrightarrow \{0,1\}$
- Forme normale 1 : $U_f : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$

  $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$

- Forme normale 2 : $S_f : |x\rangle \mapsto (-1)^{f(x)}|x\rangle$

Réaliser S_f avec U_f

- Observons : $U_f\left(|\psi\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = (S_f|\psi\rangle) \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$



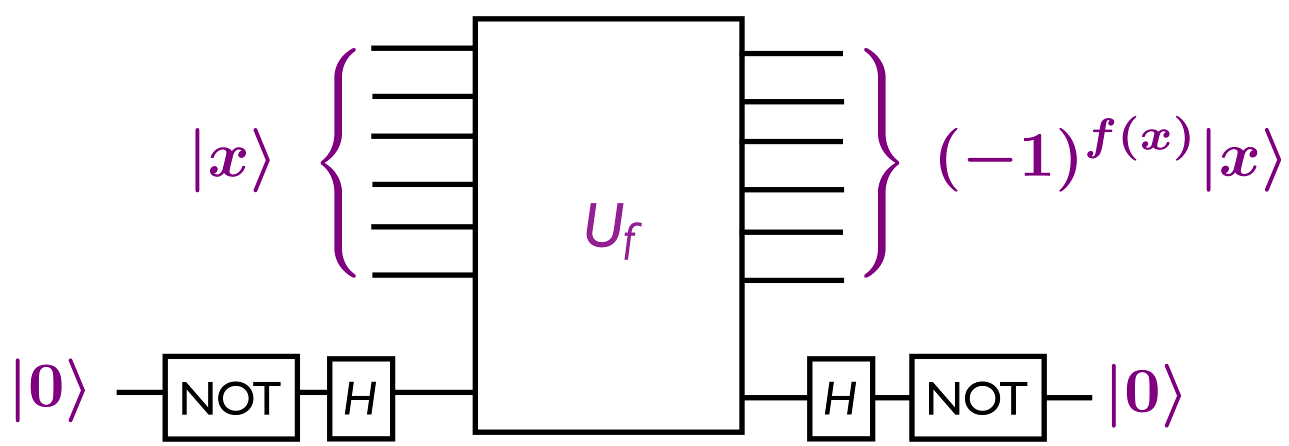
Deux formes normales

- Fonction booléenne : $f : \{0,1\}^n \longrightarrow \{0,1\}$
- Forme normale 1 : $U_f : |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$
  $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$
- Forme normale 2 : $S_f : |x\rangle \mapsto (-1)^{f(x)}|x\rangle$

Réaliser S_f avec U_f

- Observons : $U_f\left(|\psi\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = (S_f|\psi\rangle) \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$

Ou encore :



Des circuits quantiques aux algorithmes quantiques

Notion d'uniformité

- Pour être utilisable, <u>un circuit doit être codé par un objet fini</u>
- Une famille de circuits (C_n) est un algorithme si

  elle est définie par un algorithme classique qui

    prenant en entrée n

    fournit la suite des portes de C_n ("description" de C_n)

Complexité

- Complexité en temps d'une famille uniforme de circuit

  Temps pour décrire C_n + Temps pour simuler C_n
- Complexité en espace

  Espace pour décrire C_n + Espace pour simuler C_n

Pourquoi s'arrêter là ?

Ressources

- Mémoire classique (en fait inutile mais pratique !) : en bits ou autre
- Mémoire quantique : en qubits ou autre

Opérations quantiques

- Préparer un qubit/registre quantique dans un état “classique”
- Appliquer une porte (ou un circuit) quantique définie “classiquement”
- Mesurer dans la base “classique”

Contrôle

- Par un algorithme classique (uniformité) qui peut alterner
  - calcul classique
  - opérations quantiques définies par sa mémoire classique

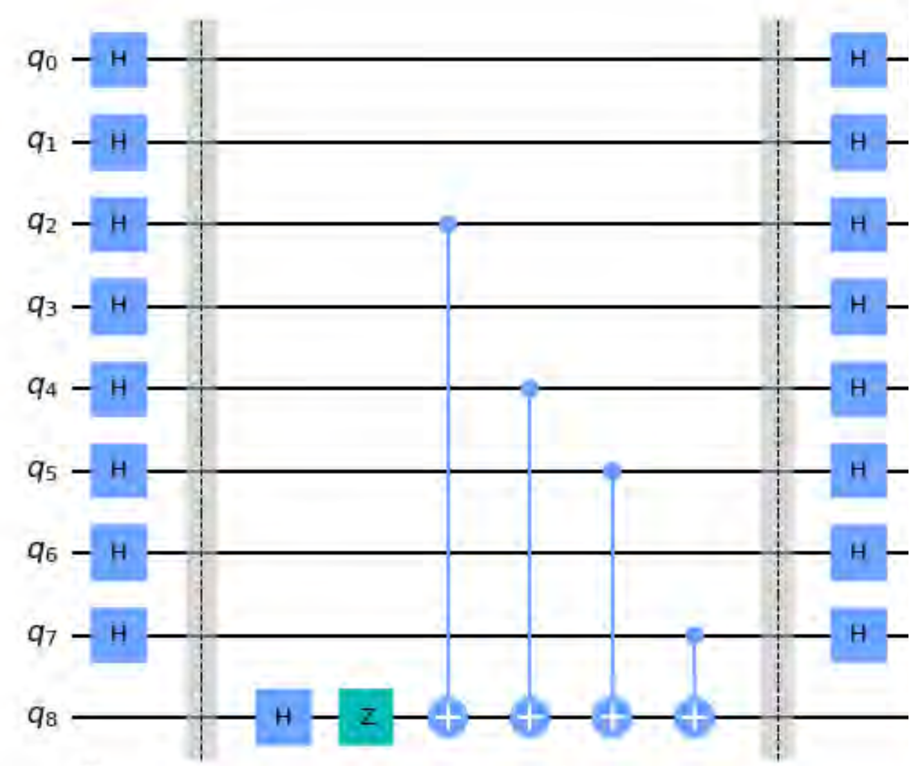
Exemple avec l'algorithme de Bernstein-Vazirani

https://qiskit.org/textbook/ch-algorithms/bernstein-vazirani.html

Le programme Qiskit

```
# We need a circuit with n qubits, plus one auxiliary qubit
# Also need n classical bits to write the output to
bv_circuit = QuantumCircuit(n+1, n)

# put auxiliary in state (|0> - |1>)/√2
bv_circuit.h(n)
bv_circuit.z(n)

# Apply Hadamard gates before querying the oracle Uf
for i in range(n):
    bv_circuit.h(i)

# Apply the oracle defined by another programme
bv_circuit += oracle

#Apply Hadamard gates after querying the oracle
for i in range(n):
    bv_circuit.h(i)

# Measurement
for i in range(n):
    bv_circuit.measure(i, i)
```

```
# hidden string
a = "10110100"
# reverse a to fit qiskit's qubit ordering
a = a[::-1]
n = len(a)
oracle = QuantumCircuit(n+1)

# Place cNOT-gates
for i in range(n):
    if a[i] == '1':
        oracle.cx(i, n)
```



Bernstein-Vazirani

- Problème

  Entrée : Oracle $f : \{0,1\}^n \to \{0,1\}$ tq $f(x) = a \cdot x$

  Sortie : a

- Solution

BV	Requêtes	Temps
Classique	n	$\Omega(n)$
Quantique	1	$O(n)$

Version récursive : *Recursive Fourier Sampling*

- Exemple à 2 niveaux

  Entrée : Oracles $f, g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ tq pour tout x, y

  $f(x, y) = a_x \cdot y$ avec $a_x \in \{0,1\}^n$

  $g(x, a_x) = b \cdot x$ avec $b \in \{0,1\}^n$ (et $g(x, y) = 0$ si $y \neq a_x$)

  Sortie : b

- Solution :

  Simuler la fonction $F(x) = g(x, a_x)$

  Utiliser l'algorithme précédent

2 niveaux	Requêtes	Temps
Classique	n^2	$\Omega(n^2)$
Quantique	≤ 3	$O(n)$

Simulation d'un oracle

- Objectif : simuler S_F avec $F(x) = g(x, a_x) = b \cdot x$
- Etat initial :

  $|x,0\rangle$

1. Appliquer le circuit de BV sur 2e registre avec l'oracle pour f (à x fixé)

   $|x, a_x\rangle$

2. Appeler l'oracle g

   $(-1)^{g(x,a_x)}|x, a_x\rangle = (-1)^{b\cdot x}|x, a_x\rangle$

3. Inverser le circuit de BV à l'étape 1

   $(-1)^{b\cdot x}|x,0\rangle$

Bilan

- 2 niveaux

  Requêtes : 3 - Temps : $2n+2\times 2n=6n$

- Extension à plusieurs niveaux

  Classique : $T(i+1) \geq nT(i)$

  Quantique : $T(i+1) \leq 2T(i) + 2n$

log n niveaux	Requêtes	Temps
Classique	$n^{\log n}$	$\Omega(n^{\log n})$
Quantique	$\leq n$	$O(n^2)$

- Réfutation de la thèse de Church-Turing quantitative (avec oracle) !

Références en lien avec le cours

Modèles, complexité, portes universelles

- Quantum Complexity Theory

  E. Bernstein, U. Vazirani. SIAM J. Comput., 26(5), 1997

- Classical and Quantum Computation

  A. Yu. Kitaev, A. H. Shen, M. N. Vyalyi. American Mathematical Society

Programmation

- Circuits

  https://quantum-computing.ibm.com

- Qiskit

  https://qiskit.org

Séminaire du cours !

- Langages graphiques pour programmer et raisonner en informatique quantique

  avec Simon Perdrix, CNRS, Nancy