Information Quantique

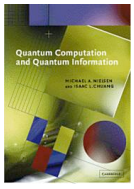
Frédéric Magniez

Cours 2 :
Rappels sur le calcul classique,
circuits quantiques,
algorithmes élémentaires

Un peu de bibliographie

Quantum Computation and Quantum Information

- Auteurs : Michael A. Nielsen, Isaac L. Chuang
- Editeur : Cambridge University Press



Classical and Quantum Computation

- Auteurs : A. Yu. Kitaev, A. H. Shen, M. N. Vyalyi
- Editeur : Amer Mathematical Society
- Collection : Graduate Studies in Mathematics

Lecture Notes for Quantum Computation

- Auteur : John Preskill
- Lien URL : http://www.theory.caltech.edu/~preskill/ph229/

Le calcul ?

Calculabilité

- Que veut dire calculer ?
- Qu'est-ce qu'une machine, un programme ?
- Comment modéliser un ordinateur ?

  indépendamment de la technologie, du langage, du système utilisés

Difficulté d'un problème

Définitions intrinsèques (indépendantes de la technologie, du langage, du système utilisés) pour

- Calculable / Non calculable
- Facile / Intraitable

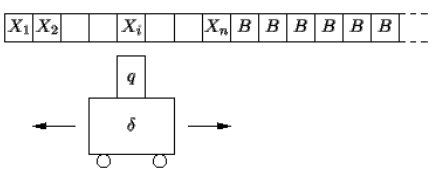
Rappels historiques

- Machine de Turing, calculabilité, universalité : [Turing 1936]
- Proposition : EDVAC (Electronic Discrete VAriable Computer) [von Neumann 1945]
- Premier ordinateur : Mark I [Robinson-Tootill-Williams 1949]

Machine de Turing (MdT)

Principe

- Ruban : juxtaposition de cellules, chaque cellule contient un symbole
- Tête positionnée en face d'une cellule, et dans un état
- A chaque étape/transition, la tête lit le contenu une cellule puis décide de
  1. Modifier le contenu de la cellule
  2. Se déplacer d'une cellule à gauche ou à droite (ou ne pas se déplacer)
  3. Changer d'état
- Entrée : sur le ruban
- Sortie : sur le ruban



Définition

- Ensemble fini de symboles (alphabet avec un caractère blanc B) : Σ
- Ensemble fini d'états (avec un/des état(s) FIN): Q
- Fonction de transition : $\delta : (Q \times \Sigma) \rightarrow (Q \times \Sigma) \times \{-1,0,1\}$

Théorème

- Une MdT quelconque peut être simulée par une MdT sur l'alphabet booléen

Calculabilité

- Une fonction est calculable si il existe une MdT qui la calcule (pour toute entrée)

Temps d'exécution

- Le temps d'exécution d'une MdT sur une entrée donnée est le nombre (éventuellement infini) de transitions qu'elle effectue avant d'être dans un état FIN

Complexité

- La taille d'une entrée est le nombre de symboles qui la composent
- La complexité d'une fonction est le temps d'exécution minimal $T(n)$ d'une MdT qui calcule la fonction sur toute entrée de taille n

Efficacité / facile

- Une MdT est efficace si son temps d'exécution est polynomial en la taille de son entrée, soit en n^c pour une certaine constante $c \geq 0$
- Une fonction est facile s'il existe une MdT efficace qui la calcule

Représentation binaire d'un entier $0 \leq x < 2^{n+1}$

- Encodage en binaire : $x = \sum x_i \, 2^{n-i} \rightarrow \boldsymbol{X} = (x_1, x_2, ..., x_n)$
- Comparaison : $x < y$ ssi $\boldsymbol{X} \lhd \boldsymbol{Y}$ où $\lhd$ est l'ordre lexicographique

  $\boldsymbol{X}$ et $\boldsymbol{Y}$ sont complétés par des 0 à gauche pour avoir la même taille

  Temps linéaire (en n)
- Addition

  Addition bit à bit avec propagation de retenue : temps linéaire
- Multiplication

  Plus difficile : algorithme naïf quadratique (en n^2)

  Algorithme sophistiqué de multiplication rapide (comme la FFT) en $n \log n$

Restes modulaires d'un entier $0 \leq x < 2^{n+1}$

- $p_1 < p_2 < ... < p_k$: premiers 2 à 2 $\neq$, $p_1 \times p_2 \times ... \times p_k \approx 2^{n+1}$,
- Encodage modulaire : $x \rightarrow \boldsymbol{X} = (x_1, x_2, ..., x_n)$ tq $x_i = x \mod p_i$
- Multiplication : facile
- Comparaison : plus difficile

Exercice 1 : MdT sur alphabet binaire

Ecrire une MdT pour chaque question, qui

- Renverse l'entrée

  Par exemple, qui renverse 0010111 en 1110100
- Reconnaît un palindrome

  Par exemple, 11100100111

  Construire une machine avec 2 états STOP : Accepte et Rejette

Exercice 2 : Problème de l'arrêt

- Peut-on encoder une MdT quelconque sur un alphabet fini ? binaire ?
- Supposons l'existence d'une MdT *H* qui

  Prend en entrée une description d'une MdT *M* et d'une entrée *x*

  Accepte si *M* s'arrête sur l'entrée *x* (pas de boucle infinie)
- A l'aide de *H*, construire une MdT *A* qui sur l'entrée *x* boucle si *H* ne boucle pas sur l'entrée (*x*,*x*), et termine sinon.
- Prouver une contradiction en exécutant *A* sur l'entrée *A*

Théorème

Il existe une MdT universelle qui

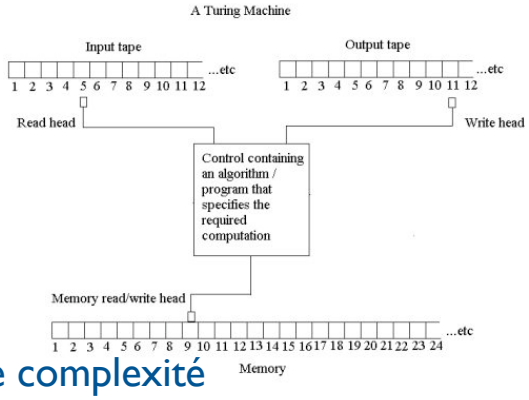
- prend en entrée une description d'une MdT *M* et d'une entrée *x*
- calcule la sortie de *M* et d'une entrée *x* (boucle si *M* boucle)
- en temps compatable à celui de *M* à une constante multiplicative près

Remarques

- Une MdT universelle est un compilateur, l'encodage de *M* étant le langage de programmation
- Tout compilateur de ce type est de même efficacité, à une constante multiplicative près

Machine plus générale

- MdT avec plusieurs rubans
- Théorème. Plusieurs rubans : simulation quadratique
- Exercice. Illustrer le théorème en résolvant l'exercice 1 avec une MdT à 2 rubans



Classes de complexité

- P : fonctions calculables en temps polynomial
- PSPACE : fonctions calculable en espace polynomial (nb de cellules ≠ B)
- NP : fonctions dont la valeur est de taille poly et se vérifie en temps poly

Exercice. Quelles relations d'inclusions existent-ils entre P, PSPACE et NP ?

Millennium Problems

In order to celebrate mathematics in the new millennium, The Clay Mathematics Institute of Cambridge, Massachusetts (CMI) has named seven *Prize Problems*. The Scientific Advisory Board of CMI selected these problems, focusing on important classic questions that have resisted solution over the years. The Board of Directors of CMI designated a $7 million prize fund for the solution to these problems, with $1 million allocated to each. During the Millennium Meeting held on May 24, 2000 at the Collège de France, Timothy Gowers presented a lecture entitled *The Importance of Mathematics*, aimed for the general public, while John Tate and Michael Atiyah spoke on the problems. The CMI invited specialists to formulate each problem.

One hundred years earlier, on August 8, 1900, David Hilbert delivered his famous lecture about open mathematical problems at the second International Congress of Mathematicians in Paris. This influenced our decision to announce the millennium problems as the central theme of a Paris meeting.

The rules for the award of the prize have the endorsement of the CMI Scientific Advisory Board and the approval of the Directors. The members of these boards have the responsibility to preserve the nature, the integrity, and the spirit of this prize.

Paris, May 24, 2000

Problem list

- Birch and Swinnerton-Dyer Conjecture
- Hodge Conjecture
- Navier-Stokes Equations
- P vs NP
- Poincaré Conjecture
- Riemann Hypothesis
- Yang-Mills Theory
- Rules
- Millennium Meeting Videos

Thèse de Church-Turing : MdT = Ordinateur

Version originale

- "Every 'function which would naturally be regarded as computable' can be computed by a Turing machine."

Version physique

- "Every function that can be physically computed can be computed by a Turing machine."

Version forte

- Any 'reasonable' model of computation can be efficiently simulated on a probabilistic Turing machine."

  modèle raisonnable : physiquement réalisable (en principe)

  simulation efficace : simulation polynomiale

  MdT probabiliste : MdT dont un des rubans est une suite de bits aléatoires (0/1 avec proba 1/2)

Théorème. MdT *probabiliste* = JAVA/C + *random* + mémoire infinie

Classe probabiliste

- BPP : fonctions calculables en temps polynomial avec une probabilité d'erreur bornée (par ex, <1/3)

  L'erreur peut arbitrairement être réduite à ε par $\log(1/\varepsilon)$ itérations

Test de primalité

Remarque : Primalité est aussi dans P.
Problème ouvert : BPP =? P

Primalité

- Input : entier n
- Output : Accepte si n est premier, Rejette sinon

Définition

- Symbole de Legendre : $L(a,p) = \pm 1$ p nombre premier
  Relation de Legendre : $a^{(p-1)/2} \mod p = L(a,p)$
- Symbole de Jacobi : $J(a,n) = \pm 1$ ou 0
  La relation de Legendre est fausse pour au moins 50% des a
- Les symboles de Legendre et Jacobi sont calculables en $(\log n)^3$

Algorithme de Solovay-Strassen

Repéter k fois

- Choisir a au hasard dans l'intervalle $[1,n-1]$
- Calculer $x = J(a,n)$
- Si $x = 0$ ou $a^{(n-1)/2} \mod n \neq x$ alors Rejette

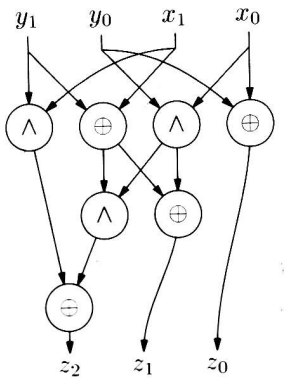
Accepte

Théorème.

L'algo accepte tout nombre premier, et rejette les autres avec proba $\geq 1-(1/2)^k$

Intérêts

- Pratique : Modèle proche des circuits imprimés
- Théorique : Similaire aux formules logiques



Définitions

- Fonction booléenne à n variables : $\{0,1\}^n \rightarrow \{0,1\}$
- Base : ensemble de fonctions booléennes fixées appelées portes (agissant sur ≤ **3** bits)
- Circuit à n bits sur une base A :
  - n variables d'entrées $x_1, ..., x_n$
  - suite d'assignations $y_j = f_j(u_1, ..., u_r)$
    - $f_j \in A$
    - $u_1, ..., u_r$ sont des variables d'entrées ou des variables auxiliaires précédant y_j
  - Le résultat du circuit en est la valeur de la dernière variables auxiliaire
- Extension possible pour des résultats à m bits

Exercice. Que calcule le circuit de la figure ?

Note. Une formule est un circuit où chaque var auxi n'est utilisée qu'une seule fois

Définition

- Une base est universelle si elle permet de construire pour toute fonction un circuit la calculant

Exercices

- Montrer que la base des opérateurs booléens OR, AND et NOT est universelle
- Que se passe-t-il si on retire OR ? ou AND ?
- Montrer que la base OR, XOR est universelle.

Théorème

- Il existe un algorithme efficace qui décide si une base est universelle

Taille d'un circuit - Complexité

Définition

- La taille d'un circuit est le nombre de assignations/portes qu'il utilise
- La complexité en circuit d'une fonction est la taille minimale du circuit la calculant

Remarques

- Etant données 2 bases universelles finies, la complexité en circuit d'une fonction diffère au plus d'une constante multiplicative
- Il existe une fonction booléenne à n variables de complexité en circuit c_n

$$1.99^n < c_n < 2.01^n$$

Définition

- P/poly : fonctions F dont les restrictions F_n à $\{0,1\}^n$ ont une complexité en circuit polynomiale, pour tout n

Théorème

- P et BPP sont inclus strictement dans P/poly

Théorème. Une fonction F est dans P ssi (F est dans P/poly et)

- Pour chaque n, un circuit pour F_n peut être construit par une MdT en temps poly(n)

Portes et circuits quantiques

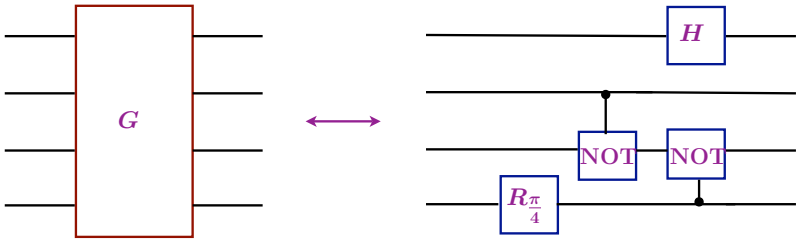
Portes

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

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

Circuit

- Un circuit quantique est la composition de portes (étendues par $\otimes$ Id sur les autres qubits)



$$G = ((I_8 \otimes R_{\frac{\pi}{4}})(I_2 \otimes \text{c-Not})(H \otimes I_2 \otimes \text{c-Not'})$$

Théorème. Familles universelles (au sens approché)

- NOT, H et Toffoli (c-c-NOT) (transformations uniquement réelles)
- NOT, √H et c-NOT
- La porte c-NOT et toutes les portes sur 1-qubit (simulation exacte)

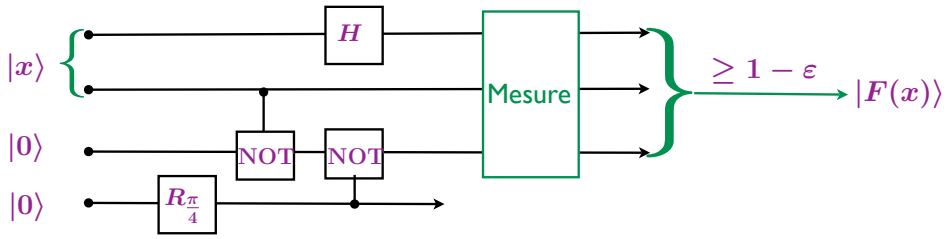
Calculer quantiquement une fonction

Définition

- Un circuit $U = U_L \dots U_2 U_1$ calcule une fonction F avec erreur ε 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 ε par $\log(1/\varepsilon)$ itérations

Complexité

Définition

- Un algorithme quantique pour le calcul d'une fonction $F : \{0,1\}^* \to \{0,1\}^*$ est un algorithme classique qui calcule une famille de circuits $(C_n)_{n\in\mathbb{N}}$ telle que C_n calcule avec erreur $\varepsilon < 1/3$ la fonction F restreinte aux entrées de $\{0,1\}^n$
- La complexité en temps $T(n)$ d'un algorithme quantique est la taille du circuit C_n **PLUS** le temps qu'il faut pour décrire le circuit C_n avec précision $O(1/|C_n|)$

Remarques

- En règle générale la description du circuit est négligeable
- Les amplitudes des portes sont donc calculables !
- La complexité *ne dépend pas* du choix des portes

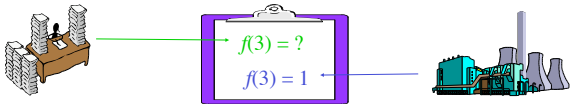
Classes de complexité

- Fonctions (problèmes) calculable (résoluble) en temps polynomial en la taille de l'entrée
  - déterministe : P
  - probabiliste avec erreur $\varepsilon < 1/3$: BPP
  - quantique avec erreur $\varepsilon < 1/3$: BQP

$$\mathrm{P} \subseteq \mathrm{BPP} \subseteq \mathrm{BQP}$$

Problème

- Entrée : $f : \{0,1\}^n \to \{0,1\}$ soit constante, soit balancée
- Sortie : 0 ssi f est constante
- Contrainte : f est une boîte noire



Complexité en requêtes

- Déterministe : $1 + 2^{n-1}$
- Quantique : 1

Cas $n = 1$

- Problème équivalent à décider si $f(0) = f(1)$ pour f quelconque

STOP $x \mapsto f(x)$ n'est pas nécessairement réversible !

Implémentation de f

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

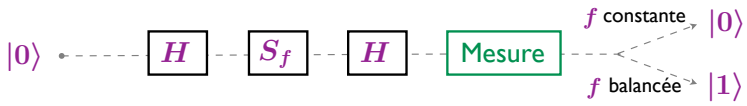
Porte de Hadamard : lame demi-onde à 22,5°

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

Circuit quantique

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

Analyse ($n = 1$)



Initialisation : $|0\rangle$

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

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

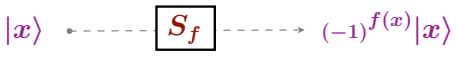
Interférences : $\frac{1}{2}((-1)^{f(0)}(|0\rangle + |1\rangle) + (-1)^{f(1)}(|0\rangle - |1\rangle))$

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

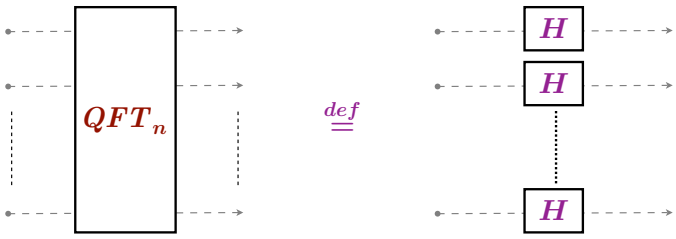
Dans ce cas la supériorité du quantique ne vient pas de l'enchevêtrement, mais des interférences constructives et destructives.

Solution quantique (cas général)

Implémentation de f



Transformée de Fourier quantique



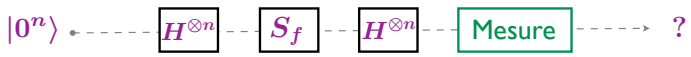
Exercice

- Vérifier que

$$QFT_n|x\rangle = \frac{1}{2^{n/2}} \sum_y (-1)^{x \cdot y}|y\rangle \quad \text{avec} \quad x \cdot y = \sum_i x_i y_i \mod 2$$

Exercice 1

- Montrer que le circuit suivant résout le problème



Exercice 2

- Montrer que le même circuit permet de trouver f avec la promesse que

$$f(x) = a \cdot x$$

- Remarque : on montre que pour ce problème, la complexité probabiliste en requêtes est en $\Omega(2^{n/2})$

Circuit réversible

- Un circuit logique est réversible s'il n'utilise que des portes réversibles
- Un circuit réversible est aussi un circuit quantique
  (car il permute les éléments de la base classique)

Notation : $f : \{0,1\}^n \to \{0,1\}^m$

$$f_\oplus : \{0,1\}^{n+m} \to \{0,1\}^{n+m} \qquad f_\oplus(x,y) = (x, y \oplus f(x))$$

Théorème

- Toute fonction F calculable par un circuit logique de taille L est aussi calculable par un circuit réversible de taille $O(L)$

  porte f ➡ porte réversible $f_\oplus$ + c-NOT

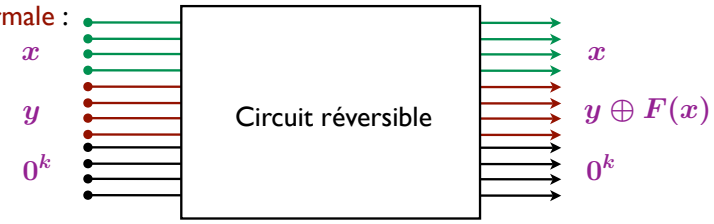
Remarque

- La porte Toffoli (c-c-NOT) est universelle pour le calcul réversible

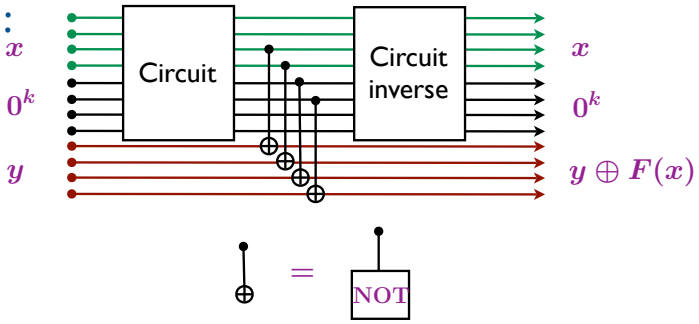
Forme normale d'un calcul réversible

Théorème

- Dans le théorème précédent on peut demander que le circuit calcule $F_\oplus$ et que les bits auxiliaires reviennent à 0, *i.e.* que le circuit soit en forme normale :



Preuve :



Calcul réversible vs calcul quantique

Corollaire

- Si *F* a une complexité classique *L* alors sa complexité quantique est en *O*(*L*)
- *F* et c-*F* ont des complexités classiques (resp. quantiques) équivalentes

Théorème

- La porte Toffoli (avec la porte NOT pour générer des bits à 1) est universelle pour le calcul réversible

$$T(a,b,c) = (a,b,c \oplus (a \wedge b))$$

- La porte Toffoli (avec NOT...) et la porte de Hadamard sont universelles pour le calcul quantique
- La porte c-NOT et la porte √H (avec NOT...) sont universelles pour le calcul quantique (avec amplitudes réelles)

Exercice

- Montrer comment implémenter $S_F|x\rangle = (-1)^{F(x)}|x\rangle$, lorsque F est à valeurs booléennes, en utilisant $U_F|x,y\rangle = |F_\oplus(x,y)\rangle = |x, y \oplus F(x)\rangle$

Théorème

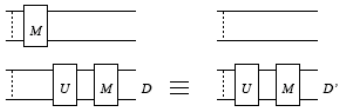
- Une fonction calculable par un circuit avec des mesures intermédiaires l'est aussi par un circuit comparable avec uniquement une mesure à la fin.

Exercice

- Montrer le théorème pour les cas suivants :

  Faire un raisonnement à l'aide de matrices densités bien choisies

  Mesure implicite



  Mesure de contrôle