Calcul et informatique quantique: une introduction formelle

Antoine Groudiev

ENS Ulm

18 Janvier 2024

Plan

Plan

Introduction

Notation de Dirac

Représentation vectorielle

Visualisation avec la sphère de Bloch

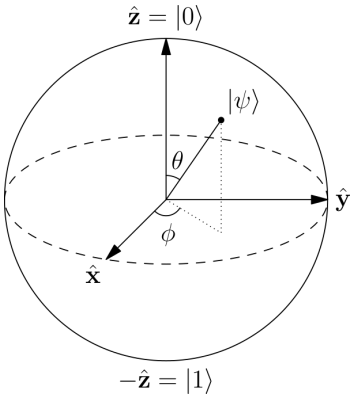

Figure – Sphère de Bloch

Plan

Circuits quantiques

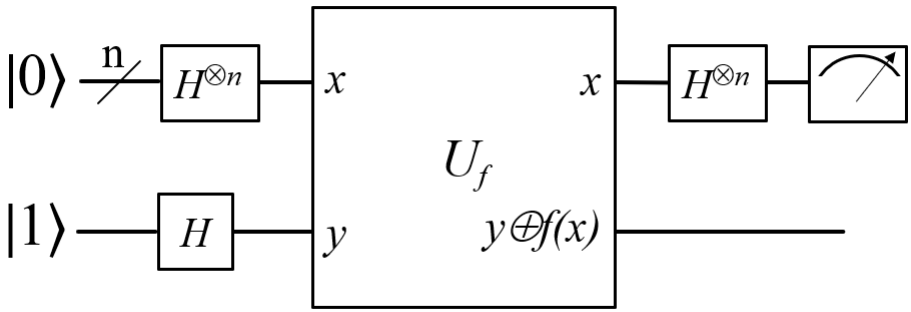

Figure – Un exemple de circuit (Algorithme de Deutsch-Jozsa)

Porte X

Porte Z

Porte de Hadamard

Intrication quantique

Porte *CNOT*

Langage quantique

Retour sur les langages classiques

Soit Σ un alphabet, et $L \subseteq \Sigma^\star$ un langage. L peut être défini alternativement comme un sous-ensemble de $\Sigma^\star$, ou par sa fonction caractéristique χ_L :

$$\chi_L(w) = \begin{cases} 1 & \text{si } w \in L \\ 0 & \text{sinon} \end{cases}$$

Langage quantique

Définition

On peut par analogie définir un *langage quantique* comme une fonction associant des probabilités à des mots :

Définition (Langage quantique)

Un *langage quantique* sur l'alphabet Σ est une fonction f telle que :

$$f : \Sigma^\star \to [0, 1]$$

Remarque

f est un langage classique lorsque $f(\Sigma^\star) \subseteq \{0, 1\}$.

Automate quantique fini

Définition (AQF)

Un *Automate Quantique Fini* $\mathcal{A} = (H, s_{\text{init}}, H_{\text{accept}}, P_{\text{accept}}, \Sigma, \delta)$ constiste en :

- un espace de Hilbert H de dimension n
- un vecteur initial normalisé $s_{\text{init}} \in H$ (i.e. $\|s_{\text{init}}\|^2 = 1$)
- un sous-espace $H_{\text{accept}} \subseteq H$, et un opérateur P_{accept} projettant sur H_{accept}
- un alphabet Σ
- une fonction $\delta : \Sigma \to U_n(\mathbb{C})$, associant à chaque lettre une matrice unitaire U_a (c'est-à-dire $U_a U_a^\dagger = I_n$)

On note $\delta^\star(w = w_1 \cdots w_{|w|}) = \delta(w_{|w|}) \cdots \delta(w_1) = U_{w_{|w|}} \cdots U_{w_1}$. Enfin, le langage reconnu par $\mathcal{A}$ est :

$$f_{\mathcal{A}} : w \mapsto \|P_{\text{accept}} \delta^\star(w) s_{\text{init}}\|^2$$

Langage quantique régulier et propriétés

Définition (LQR)

Un *Langage Quantique Régulier* est un langage quantique reconnu par un automate quantique fini

Théorème (Clôture des LQR par produit)

Soient f, g des LQRs. Alors, le produit fg est un LQR.

Théorème (Clôture des LQR par combinaison linéaire)

Soient f_i des LQRs, et c_i des constantes telles que $\sum_i c_i \leq 1$. Alors, $\sum_i c_i f_i$ est un LQR.

Langage quantique régulier et propriétés

Théorème (Lemme de pompage quantique)

Si f est un LQR, alors pour tout mot $w \in \Sigma^\star$ et tout $\varepsilon > 0$, il existe $k \in \mathbb{N}^\star$ tel que $\|f(uw^k v) - f(uv)\| < \varepsilon$ pour tout mots u, v. De plus, si l'automate de f est de dimension n, alors il existe une constante c (indépendante de ε) telle que $k < (c\varepsilon)^{-n}$.

Grammaire quantique hors-contexte

Définition (Grammaire[1] Quantique Hors-Contexte)

Une *Grammaire Quantique Hors-Contexte* $G = (V, T, I, P)$ de *dimensionnalité* n consiste en :

- un alphabet V de variables
- un alphabet T de terminaux
- une variable initiale $I \in V$
- un ensemble fini de productions P de la forme $\alpha \to \beta$, où $(\alpha, \beta) \in V \times (T \cup V)^\star$.

1. https://xkcd.com/1090/

À chaque production de P est associée un ensemble d'amplitudes complexes $(c_k(\alpha \to \beta))_{1 \leq k \leq n}$.
On définit l'amplitude d'*une* suite de productions :

$$c_k(\alpha_1 \to \cdots \to \alpha_m = \beta) := \prod_{i=1}^{m-1} c_k(\alpha_i \to \alpha_{i+1})$$

Et l'amplitude d'une dérivation :

$$c_k(\alpha \Rightarrow \beta) := \sum_{\alpha=\alpha_1 \to \cdots \to \alpha_m = \beta} c_k(\alpha_1 \to \cdots \to \alpha_m)$$

Enfin, G *génère* le langage quantique f définit par :

$$f(w) = \sum_{k=1}^{n} \|c_k(I \Rightarrow w)\|^2$$

Automate à pile quantique

Définition (QPDA)

Un *Automate à Pile Quantique* $\mathcal{A} = (H = Q \otimes \Sigma, \Gamma, Q_{\text{acccept}}, s_{\text{init}}, A, \delta)$ consiste en :

- un espace de Hilbertt H des configurations de $\mathcal{A}$, avec $H = Q \otimes \Sigma$ pour Q, Σ
- chaque vecteur de base de Σ est un mot fini de *l'alphabet de pile* Γ
- un état initial s_{init}
- $Q_{\text{acccept}} \subseteq Q$, tel que $H_{\text{acccept}} = Q_{\text{acccept}} \otimes \{\varepsilon\}$
- un alphabet d'entrée A
- une fonction de transition δ, telle que $\forall a \in A$, $\delta(a)$ est un endomorphisme unitaire

Automate à pile quantique

(Suite)

Pour s'assurer du comportement LIFO de la pile, on ajoute les contraintes suivantes : Soient $q_1, q_2 \in Q$ des états de contrôle, et $\sigma_1, \sigma_2 \in \Gamma$ des états de la pile ; l'amplitude de la transition (q_1, σ_1) to (q_2, σ_2) est non-nulle seulement s'il existe $t \in T$ tel que :

$$(\sigma_1 t = \sigma_2) \vee (\sigma_1 = \sigma_2 t) \vee (\sigma_1 = \sigma_2)$$

De plus, les amplitudes de transition peuvent dépendre uniquement de la dernière lettre de σ_1 et de σ_2.
Le langage reconnu par $\mathcal{A}$ est la fonction :

$$f_{\mathcal{A}} : w \mapsto \| P_{\text{accept}} \delta^*(w) s_{\text{init}} \|^2$$

Théorème d'équivalence

Théorème (Équivalence entre GQHC et APQ)

Un langage quantique est généré par une grammaire quantique hors-contexte si et seulement si il est reconnu par un automate à pile quantique.

Machine de Turing quantique

Définition (MTQ)

Une *Machine de Turing Quantique* $M = (H, \Gamma, b, \Sigma, \delta, q_0, Q_{\text{accept}})$ consiste en :

- un espace de Hilbert Q des états
- un autre espace de Hilbert Γ de la bande
- un symbole blanc $\sqcup \in \Gamma$
- un alphabet d'entrée et de sortie Σ
- un état (vecteur) initial $q_0 \in Q$
- un sous-espace $Q_{\text{accept}} \subseteq Q$
- une fonction de transition δ telle que :

$$\delta : \Sigma \times Q \otimes \Gamma \to \Sigma \times Q \otimes \Gamma \times \{L, R\}$$

Plan

Classe BQP (Bounded-error Quantum Polynomial time)

Définition (BQP)

La classe *Bounded-error Quantum Polynomial time* (BQP) est l'ensemble des problèmes de décision qui peuvent être résolus en temps polynomial par une machine de Turing quantique, avec une erreur maximale de $\frac{1}{3}$.

Un problème Promise-BQP-complet

Définition (Problème APPROX-QCIRCUIT-PROB)

ENTRÉE : α, β tels que $0 \leq \beta < \alpha \leq 1$, et la description d'un circuit quantique C tel que :

- le circuit prend en entrée n qubits
- le circuit contient m portes
- m est polynomial en n
- la probabilité P de mesurer le premier qubit du circuit initialisé à $|0\rangle^{\otimes n}$, est telle que $P \geq \alpha$ ou $P \leq \beta$

SORTIE : Déterminer si $P \geq \alpha$ ou $P \leq \beta$.

Théorème (Complétude)

Tout problème BQP se réduit en APPROX-QCIRCUIT-PROB.

Positionnement par rapport aux classes de complexité classiques

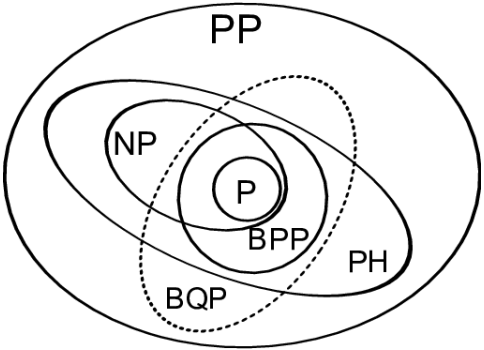

Figure – Inclusions connues et supposées de BQP

Rapport à la thèse de Church-Turing

Thèse (Thèse de Church-Turing)

Une fonction sur les entiers naturels peut être calculée si et seulement si elle est calculable par une machine de Turing.

Thèse (Thèse étendue de Church-Turing)

Une machine de Turing probabiliste peut efficacement simuler tout modèle de calcul réaliste.

Thèse (Thèse quantiquement-étendue de Church-Turing)

Tout système de calcul physique peut être efficacement simulé par une machine de Turing quantique.

Plan

Description du problème

On considère une fonction f fonctionnant sur n bits ou qubits :

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

Cette fonction est supposée être soit *constante*, soit *équilibrée* :

$$|f^{-1}(\{0\})| = n \vee |f^{-1}(\{1\})| = n \vee \left(|f^{-1}(\{0\})| = |f^{-1}(\{1\})| = \frac{n}{2}\right)$$

(C'est à dire qu'elle produit soit que des 0, soit que des 1, soit exactement la moitié de 0 et l'autre moitié de 1.)

Solution classique

Dans le pire cas, un algorithme classique déterministe doit mesurer plus de la moitié des valeurs de f pour les 2^n valeurs entrées possibles, i.e. $2^{n-1}+1$; la meilleure complexité en temps est dès lors exponentielle. (Néanmoins, ce problème peut être résolu avec une probabilité élevée avec un algorithme probabiliste, le problème est donc dans BPP.)

Algorithme de Deutsch

On suppose que f est implémentée par une porte sous la forme :

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

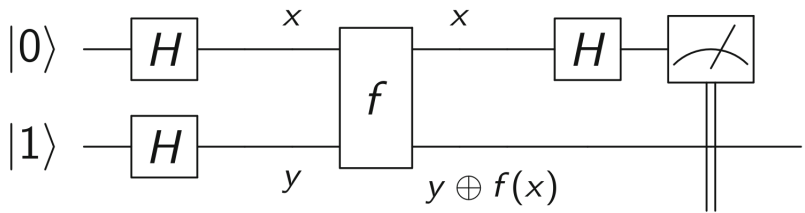

Figure – Circuit de l'algorithme de Deutsch

Algorithme de Deutsch

Cas général (n quelconque)

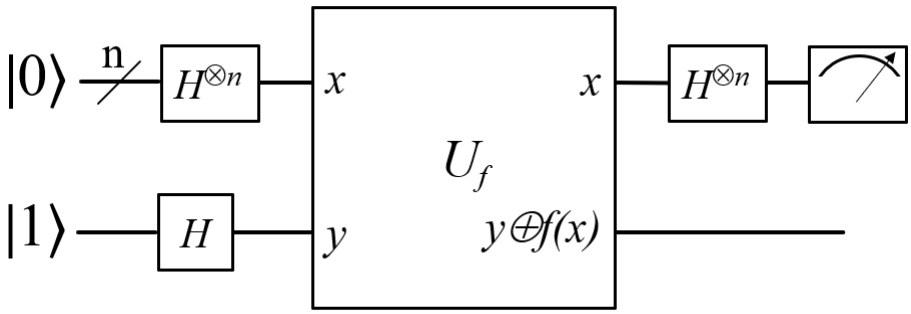

Figure – Circuit de l'algorithme de Deutsch