Calcul et informatique quantique: une introduction formelle

Antoine Groudiev

ENS Ulm

18 Janvier 2024

Plan

Plan

Introduction

On manipule non pas des *bits*, mais des *qubits*.

Définition (Superposition quantique)

$$|\Psi\rangle = \alpha|\Psi\rangle + \beta|\Psi\rangle$$

où $\alpha, \beta \in \mathbb{C}$ sont appelés les *amplitudes d'états*.

Remarque (Condition de normalisation)

$$|\alpha|^2 + |\beta|^2 = 1$$

Circuits quantiques

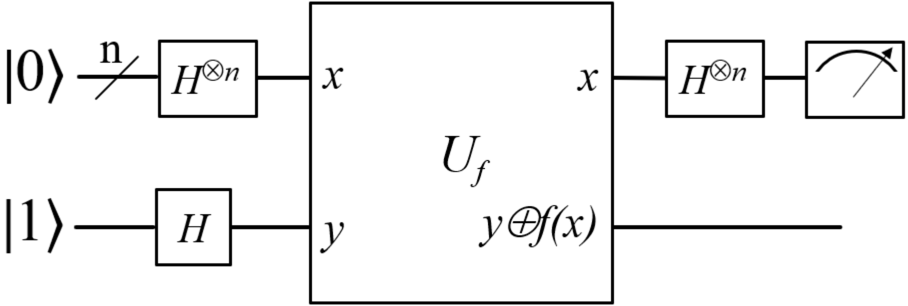


Figure – Un exemple de circuit (Algorithme de Deutsch-Jozsa)

Porte X

$$X : \alpha|0\rangle + \beta|1\rangle \mapsto \beta|0\rangle + \alpha|1\rangle$$

Sous forme de matrice :

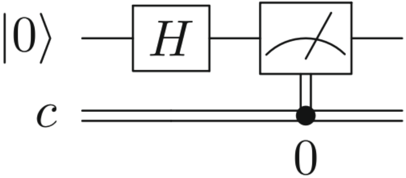
$$X \cong \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Porte de Hadamard

$$H \cong \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Résultat direct :

$$\begin{cases} H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{cases}$$



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* $\mathcal{A} = (H, s_{\text{init}}, H_{\text{accept}}, P_{\text{accept}}, \Sigma, \delta)$ consiste 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 :

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

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 $\frac{1}{3}$.

Positionnement par rapport aux classes de complexité classiques

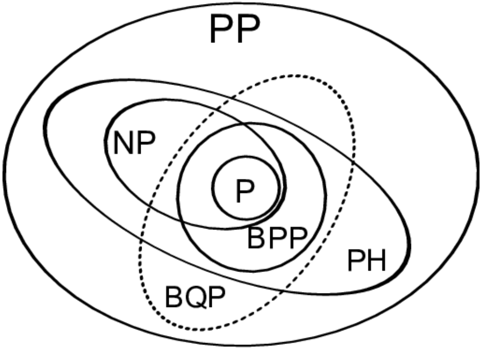


Figure – Inclusions connues et supposées de BQP

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 :

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

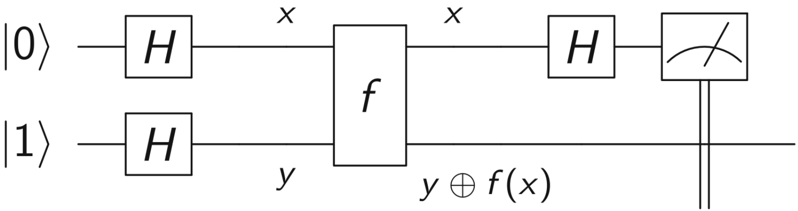


Figure – Circuit de l'algorithme de Deutsch

Algorithme de Deutsch

L'état du premier qubit est finalement :

$$\frac{1}{2}\left[\left(1+(-1)^{f(0)\oplus f(1)}\right)|0\rangle+\left(1-(-1)^{f(0)\oplus f(1)}\right)|1\rangle\right]$$

qui vaut $|0\rangle$ si et seulement $f(0) \oplus f(1) = 0$.

Cas général (n quelconque)

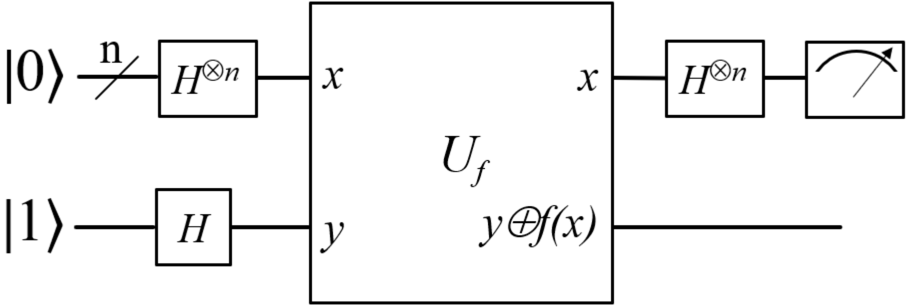


Figure – Circuit de l'algorithme de Deutsch