Fundamentos de lenguajes de programación cuántica

Introducción a la computación cuántica

Alejandro Díaz-Caro

9 al 14 de Febrero de 2015

Enfoque de este apunte

Estas notas están pensadas para estudiantes de grado de computación, no de física. Es por ello que el enfoque que se da es casi puramente matemático, con algún comentario aquí y allá de la física que motiva el formalismo, pero todos los razonamientos se realizan exclusivamente desde el lado de la matemática.

22va Escuela de Verano de Ciencias Informáticas
Universidad Nacional de Río Cuarto

Curso: Fundamentos de lenguajes de programación cuántica
Primeras dos clases: Introducción a la computación cuántica

Alejandro Díaz-Caro
http://diaz-caro.web.unq.edu.ar

Índice general

1. Introducción a la computación cuántica **5**
1.1. Introducción . . . 5
1.2. Preliminares: un poco de álgebra . . . 7
1.2.1. Espacio de Hilbert . . . 7
1.2.2. Productos tensoriales . . . 8
1.2.2.1. Entre matrices y entre vectores . . . 8
1.2.2.2. Entre espacios vectoriales . . . 8
1.2.2.3. Una propiedad llamativa del espacio $E \otimes F$ . . . 9
1.2.3. Notación bra–ket . . . 9
1.2.3.1. Notación bra y ket para vectores . . . 9
1.2.3.2. Notación bra y ket para matrices . . . 11
1.3. Bits cuánticos y operadores . . . 12
1.3.1. Primera intuición en 8 líneas . . . 12
1.3.2. Bits cuánticos . . . 12
1.3.3. Operadores . . . 12
1.4. Teorema del no-clonado . . . 15
1.5. Estados de Bell . . . 16
1.6. Usando los estados de Bell . . . 17
1.6.1. Codificación superdensa . . . 17
1.6.2. Teleportación cuántica . . . 18
1.7. Paralelismo Cuántico . . . 19

2. Algoritmos cuánticos más comunes y aplicación a la criptografía **21**
2.1. Algoritmo de Deutsch . . . 21
2.2. Algoritmo de Deutsch-Jotza . . . 22
2.3. Algoritmo de Búsqueda de Grover . . . 24
2.3.1. Oráculo . . . 25
2.3.2. Inversión sobre el promedio . . . 25
2.3.3. El algoritmo . . . 26
2.3.3.1. Paso 1: Se aplica Hadamard ($H^{\otimes n}$) . . . 26
2.3.3.2. Paso 2: Se aplica el oráculo (U) . . . 26
2.3.3.3. Paso 3: Se aplica la inversión sobre el promedio (G) . . . 27
2.3.4. Cálculo del número óptimo de iteraciones . . . 28
2.4. Aplicación criptográfica . . . 29
2.4.1. One-time pad . . . 29
2.4.2. Criptosistema Cuántico QKD-BB84 . . . 30

Capítulo 1

Introducción a la computación cuántica

> *I feel that a deep understanding of why quantum algorithms work is still lacking. Surely the power of quantum computers has something to do with entanglement, quantum parallelism, and the vastness of Hilbert space, but I think that it should be possible to pinpoint more precisely the true essence of the matter.*
>
> John Preskill [1998]

1.1. Introducción

La computación cuántica, una rama de las ciencias de la computación teórica, tiene su origen en la física, y más precisamente en el físico estadounidense Richard Feynman, quien en 1981 dedicó una charla en el Massachusetts Institute of Technology (MIT) al problema de la simulación de la física cuántica con computadoras clásicas. Sus ya célebres palabras finales resumen su frustración de ese entonces:

> *Y no estoy feliz con todos los análisis que consideran sólo la teoría clásica, porque la naturaleza no es clásica, maldita sea, y si querés hacer una simulación de la naturaleza, mejor que lo hagas cuántico-mecánicamente, y caramba si es un problema maravilloso, porque no parece muy fácil. Gracias.*[1]
>
> (ver, por ejemplo, [Brown, 2001, pp.100])

Esta provocación, lejos de plantear soluciones, abrió las puertas a interrogantes nunca antes concebidos. ¿Qué ganancia se lograría si las computadoras fuesen regidas por las leyes de la mecánica cuántica? Fueron los algoritmos de Grover [1996] y Shor [1997] los cuales despertaron el gran interés desde las ciencias de la computación en este nuevo paradigma. El primero es un algoritmo de búsqueda sobre registros desordenados, el cual provee una ganancia cuadrática de complejidad temporal frente a cualquier algoritmo clásico conocido. El segundo es un algoritmo para la factorización de números, con una ganancia exponencial.

[1] Original: *And I'm not happy with all the analyses that go with just the classical theory, because nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy. Thank you.*

Actualmente existen muchas áreas de investigación dentro de la computación cuántica. Por ejemplo, desde un punto de vista práctico se plantea el problema de construir el hardware de una computadora cuántica. Desde sus orígenes, en las palabras de Feinmann, la idea es que un algoritmo cuántico sea una simulación cuántica en hardware que se comporta de acuerdo a las leyes de la física cuántica. Es decir que un experimento cuántico en un laboratorio, puede considerarse como un algoritmo. O dicho de otro modo: podemos describir el comportamiento de un sistema cuántico a través de un algoritmo. La pregunta es, ¿podemos realizar el experimento cuántico que describe un algoritmo dado? Allí es donde se manifiesta el desafío técnico.
Otra área es la de desarrollar algoritmos que obtengan una ganancia con respecto a su contraparte clásica. En general los algoritmos de Grover y Shor mencionados anteriormente se consideran como los ejemplos canónicos de aceleración obtenida gracias a la computación cuántica. La mayor parte de algoritmos cuánticos son derivados de ellos. La pregunta aquí es ¿existen otros algoritmos que no sean derivados de estos dos ejemplos?
Otra rama de investigación es la del diseño de lenguajes de programación que permitan expresar los algoritmos cuánticos de una manera amigable, y quizá permitiendo descubrir nuevos algoritmos al tener una herramienta de alto nivel para pensarlos.
Desde un punto de vista más fundamental, y como lo expresara Preskill en la cita que abre este capítulo, los fundamentos lógicos detrás de la computación cuántica, siguen siendo un misterio. Si bien existe una lógica cuántica [Birkhoff y von Neumann, 1936], ésta fue propuesta muchos años antes de la computación cuántica, por lo que encontrar la correspondencia entre computación y lógica cuántica no es trivial. Esta área tiene muchas subáreas con metodologías diferentes. En particular, el estudio de semántica de lenguajes de programación sigue este objetivo. En este caso no se persigue el estudio del lenguaje en sí mismo, sino que el objetivo es el estudio de la lógica subyacente. Estudiar la lógica detrás de la computación cuántica implica estudiar la lógica detrás de la física cuántica, lo cual puede tener influencia en el desarrollo de nuevas teorías sobre el mundo que nos rodea.

En este curso nos interesan los dos últimos aspectos: lenguajes de programación que permitan expresar el cómputo cuántico de una manera estructurada y amigable, y el estudio de propiedades de lenguajes que nos acerquen hacia una lógica computacional de la física cuántica.

Estructura del curso y de estos apuntes En este apunte cubrimos los dos primeros días del curso: introducción a la computación cuántica. Esta es una introducción para dos días de curso, para una introducción más extensa, se recomienda el libro de Nielsen y Chuang [2010]. En el resto de este capítulo desarrollaremos los rudimentos básicos de la computación cuántica, desde un enfoque puramente matemático (en contraposición con el enfoque físico). En el Capítulo 2 explicaremos algunos de los algoritmos más conocidos y una aplicación a la criptografía.
Los tres días que seguirán del curso se desarrollarán de la siguiente manera: el tercer día introduciremos el cálculo lambda, el cual nos servirá en el estudio del diseño de lenguajes en los dos días restantes El cuarto día presentaremos el paradigma de lenguajes con control clásico y datos cuánticos. Este paradigma apunta al diseño de lenguajes como herramienta de alto nivel para desarrollar algoritmos cuánticos. Finalmente, el último día

del curso presentaremos el paradigma de lenguajes con control y datos cuánticos, el cual apunta al estudio de la lógica subyacente detrás de la computación cuántica.

1.2. Preliminares: un poco de álgebra

1.2.1. Espacio de Hilbert

TL;DR $\mathbb{C}^n$ con la suma $(+)$ y el producto $(\cdot)$ usuales, y el producto escalar definido por

$$\langle \vec{v}, \vec{w} \rangle = \langle (v_1, v_2, \ldots, v_n), (w_1, w_2, \ldots, w_n) \rangle = \sum_{i=1}^{n} v_i^* \cdot w_i$$

donde v^* es el complejo conjugado de v, es un *espacio de Hilbert*.

En el resto de la sección se define formalmente qué es un espacio de Hilbert.

Definición 1.1 (Producto escalar)**.** Sea E un espacio vectorial sobre el cuerpo $\mathbb{K}$ ($\mathbb{R}$ o $\mathbb{C}$). Un producto escalar (también llamado producto interno) definido sobre E es una función $\langle , \rangle : E \times E \to \mathbb{K}$ que verifica las siguientes propiedades.
Para todo $\vec{u}, \vec{v}, \vec{w} \in E$, $a, b \in \mathbb{K}$, se cumple:

$$\begin{cases} \langle \vec{u}, \vec{u} \rangle \geq 0 \\ \langle \vec{u}, \vec{u} \rangle = 0 \Leftrightarrow \vec{u} = \vec{0}_E \end{cases} \qquad \text{(Definida positiva)}$$

$$\langle \vec{w}, a\vec{u} + b\vec{v} \rangle = a\langle \vec{w}, \vec{u} \rangle + b\langle \vec{w}, \vec{v} \rangle \qquad \text{(Lineal por derecha)}$$

$$\langle a\vec{u} + b\vec{v}, \vec{w} \rangle = a^*\langle \vec{u}, \vec{w} \rangle + b^*\langle \vec{v}, \vec{w} \rangle \qquad \text{(Antilineal por izquierda)}$$

$$\langle \vec{u}, \vec{v} \rangle = \langle \vec{v}, \vec{u} \rangle^* \qquad \text{(Hermítica)}$$

Definición 1.2 (Espacio pre-Hilbert)**.** Un espacio pre-Hilbert es un espacio vectorial sobre $\mathbb{K}$ con producto escalar.

Observación. Todo espacio pre-Hilbert es un espacio vectorial normado con la norma

$$\|\vec{v}\| = \sqrt{\langle \vec{v}, \vec{v} \rangle}$$

Definición 1.3 (Sucesión de Cauchy)**.** Sea $\vec{v}_n$ una sucesión de vectores del espacio E. Si $\|\vec{v}_n - \vec{v}_m\| \to 0$ cuando $n, m \to \infty$, entonces la sucesión $\vec{v}_n$ es una sucesión de Cauchy. (Esto quiere decir que puedo hacer distar entre sí los términos tan poco como quiera).

Observación. Toda sucesión convergente es de Cauchy, pero no toda sucesión de Cauchy es convergente.

Definición 1.4 (Espacio completo)**.** E es completo para la norma $\|\cdot\|$, si y sólo si toda sucesión de Cauchy converge con esa norma.

Definición 1.5 (Espacio de Hilbert)**.** Un espacio pre-Hilbert completo en su norma se denomina espacio de Hilbert.

1.2.2. Productos tensoriales

1.2.2.1. Entre matrices y entre vectores

El producto tensorial $\otimes$, también llamado producto externo, se puede aplicar entre matrices, vectores, espacios vectoriales y muchas otras estructuras. En nuestro caso sólo nos interesan esas tres aplicaciones: entre matrices, entre vectores, el cual se define igual, tomando los vectores como matrices de una sola columna, y entre espacios vectoriales.

Definición 1.6 (Producto tensorial entre matrices)**.** El producto tensorial de dos matrices, P y Q se define como la matriz

$$P \otimes Q = \begin{pmatrix} p_{11}Q & \dots & p_{1m}Q \\ \vdots & & \vdots \\ p_{n1}Q & \dots & p_{nm}Q \end{pmatrix}$$

Ejemplos 1.1.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \otimes \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} & 2\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \\ 3\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} & 4\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 1\begin{pmatrix} 3 \\ 4 \end{pmatrix} \\ 2\begin{pmatrix} 3 \\ 4 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 6 \\ 8 \end{pmatrix}$$

Observación. El producto escalar, o producto interno, entre dos vectores nos da un número. El producto tensorial, o producto externo, entre dos vectores nos da un vector de mayor dimensión.

1.2.2.2. Entre espacios vectoriales

El producto tensorial entre espacios vectoriales se define como el espacio generado por el producto tensorial de todos los vectores de una base del primero con los vectores de una base del segundo.

Definición 1.7 (Producto tensorial entre bases)**.** Sean $B_E = \{\vec{e}_1, \dots, \vec{e}_{\dim(E)}\}$ una base del espacio vectorial E y $B_F = \{\vec{f}_1, \dots, \vec{f}_{\dim(F)}\}$ una base del espacio vectorial F. El producto tensorial entre dichas bases se define como

$$B_E \otimes B_F = \{\vec{e}_1 \otimes \vec{f}_1, \dots, \vec{e}_1 \otimes \vec{f}_{\dim(F)}, \vec{e}_2 \otimes \vec{f}_1, \dots \vec{e}_{\dim(E)} \otimes \vec{f}_{\dim(F)}\}$$

Ejemplo 1.2. $B_1 = \{\vec{v}_1, \vec{v}_2\},\ B_2 = \{\vec{u}_1, \vec{u}_1\}$ entonces

$$B_1 \otimes B_2 = \{\vec{v}_1 \otimes \vec{u}_1, \vec{v}_1 \otimes \vec{u}_2, \vec{v}_2 \otimes \vec{u}_1, \vec{v}_2 \otimes \vec{u}_2\}$$

Definición 1.8 (Producto tensorial entre espacios vectoriales)**.** Sean B_E una base del espacio vectorial E y B_F una base del espacio F. Entonces

$$E \otimes F = \text{Gen}(B_E \otimes B_F)$$

1.2.2.3. Una propiedad llamativa del espacio $E \otimes F$

Existen vectores de $E \otimes F$ que no son producto tensorial entre uno de E y uno de F.

Ejemplo 1.3. Consideremos el espacio $\mathbb{C}^2 \otimes \mathbb{C}^2$. Una base de $\mathbb{C}^2$ es $\{\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}\}$ Por lo tanto

$$\mathbb{C}^2 \otimes \mathbb{C}^2 = \text{Gen}(\left\{ \begin{pmatrix}0\\0\\0\\1\end{pmatrix}, \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \right\}) = \mathbb{C}^4$$

Tomemos $\vec{v} = (\alpha, 0, 0, \beta)^T$, con $\alpha, \beta \neq 0$. Es fácil verificar que $\vec{v} \in \mathbb{C}^4$. Sin embargo, no existen $\vec{v}_1, \vec{v}_2 \in \mathbb{C}^2$ tal que $\vec{v} = \vec{v}_1 \otimes \vec{v}_2$.

Demostración. Supongamos que existen $\vec{v}_1$ y $\vec{v}_2$ tales que $\vec{v}_1 \otimes \vec{v}_2 = \vec{v}$, entonces

$$\begin{pmatrix}a\\b\end{pmatrix} \otimes \begin{pmatrix}c\\d\end{pmatrix} = \begin{pmatrix}ac\\ad\\bc\\bd\end{pmatrix} = \begin{pmatrix}\alpha\\0\\0\\\beta\end{pmatrix} \Rightarrow \begin{cases} ac = \alpha \\ ad = 0 \\ bc = 0 \\ bd = \beta \end{cases}$$

pero este es un sistema que no tiene solución. □

1.2.3. Notación bra–ket

Notación introducida por Paul Dirac [1939] para describir estados cuánticos.

1.2.3.1. Notación bra y ket para vectores

En lugar de escribir los vectores como $\vec{v}$ la notación ket usa $|v\rangle$.
En particular definimos:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \qquad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

Por lo tanto, cualquier vector de $\mathbb{C}^2$ puede escribirse como

$$\begin{pmatrix}\alpha\\\beta\end{pmatrix} = \alpha \begin{pmatrix}1\\0\end{pmatrix} + \beta \begin{pmatrix}0\\1\end{pmatrix} = \alpha|0\rangle + \beta|1\rangle$$

Podemos, por ejemplo, definir vectores como los siguientes

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

y como estos son dos vectores ortogonales (por ende, forman una base), ahora es posible también escribir cualquier vector de $\mathbb{C}^2$ como combinación lineal de $|+\rangle$ y $|-\rangle$.
Por ejemplo:

$$\begin{pmatrix}\alpha\\\beta\end{pmatrix} = \alpha|0\rangle + \beta|1\rangle = \frac{1}{\sqrt{2}}(\alpha + \beta)|+\rangle + \frac{1}{\sqrt{2}}(\alpha - \beta)|-\rangle$$

Observación. Al menos que se indique lo contrario, en el resto del apunte consideraremos el espacio complejo de dimensión N, $\mathbb{C}^N$.

Definición 1.9 (Bra y Ket)**.** Llamamos ket a un vector de la forma

$$|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix}$$

y bra a un vector de la forma

$$\langle\psi| = (\alpha_1^*, \ldots, \alpha_N^*)$$

donde $\alpha_i \in \mathbb{C}$ y α_i^* denota el conjugado de α_i.

Observaciones.

- Haciendo un abuso de notación, podemos escribir vectores como el siguiente:

$$|\alpha_1\psi_1 + \alpha_2\psi_2\rangle = \alpha_1|\psi_1\rangle + \alpha_2|\psi_2\rangle$$

- A partir la definición de bras y kets, llamamos "braket" al producto escalar:

$$\langle\psi|\phi\rangle = (\alpha_1^*, \ldots, \alpha_N^*) \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_N \end{pmatrix} = a \in \mathbb{C}$$

- Recordatorio de álgebra: Una *base ortonormal* de un espacio vectorial normado es una base donde todos los vectores tienen norma 1. Además, en una base, todos los vectores son ortogonales entre sí (es decir, el producto escalar entre ellos es 0). Por lo tanto:

  Dado un conjunto $B = \{|u_1\rangle, \ldots, |u_N\rangle\}$, B es una base ortonormal de $\mathbb{C}^N$ si y sólo si para todo i, j tenemos $\langle u_i|u_j\rangle = \delta_{ij}$, donde δ_{ij} es la delta de Kronecker (igual a 1 si $i = j$, y 0 en otro caso).

- Entonces, todo Ket $|\psi\rangle$ se puede expresar como $|\psi\rangle = \sum_{i=1}^N a_i|u_i\rangle$.

- Si tomamos la base canónica de $\mathbb{C}^N$, con $|u_i\rangle$ el vector i-ésimo de dicha base, podemos calcular la componente i-ésima de un vector cualquiera de la siguiente manera:

$$\langle u_i|\psi\rangle = \langle u_i| \sum_{j=1}^N a_j|u_j\rangle = \sum_{j=1}^N a_j \underbrace{\langle u_i|u_j\rangle}_{\delta_{ij}} = a_i$$

Teorema 1.1. Sea $B = \{|u_1\rangle, \ldots, |u_N\rangle\}$ una base ortonormal, entonces $\sum_{i=1}^N |u_i\rangle\langle u_i| = I$.

Demostración.

$$\left(\sum_{i=1}^{N}|u_i\rangle\langle u_i|\right)|\psi\rangle = \left(\sum_{i=1}^{N}|u_i\rangle\langle u_i|\right)\left(\sum_{j=1}^{N}a_j|u_i\rangle\right)$$
$$= \sum_{i=1}^{N}\sum_{j=1}^{N}a_j|u_i\rangle\underbrace{\langle u_i|u_j\rangle}_{\delta_{ij}} = \sum_{i=1}^{N}a_i|u_i\rangle = |\psi\rangle \qquad \square$$

Observaciones.

- Análogamente a los kets, todo bra $\langle\phi|$ puede ser descompuesto como $\langle\phi| = \sum_{i=1}^{N} b_i^*\langle u_i|$.
- Podemos ver que $b_i^* = \langle\phi|u_i\rangle \in \mathbb{C}$ ya que

$$\langle\phi| = \langle\phi|\underbrace{\left[\sum_{i=1}^{N}|u_i\rangle\langle u_i|\right]}_{I} = \sum_{i=1}^{N}\langle\phi|u_i\rangle\langle u_i| \quad \Rightarrow \quad b_i^* = \langle\phi|u_i\rangle$$

Observación. De aquí en más, trabajaremos sólo con los vectores normalizados de $\mathbb{C}^N$ (es decir, vectores cuya norma es 1). Esto es

$$1 = \|\psi\|^2 = \langle\psi|\psi\rangle = \left(\sum_{j=1}^{N}a_j^*\langle u_j|\right)\left(\sum_{i=1}^{N}a_i|u_i\rangle\right) = \sum_{i,j=1}^{N}a_j^*a_i\underbrace{\langle u_j|u_i\rangle}_{\delta_{ij}} = \sum_{i=1}^{N}|a_i|^2 = 1$$

Es decir, trabajamos con vectores cuya suma de los módulos al cuadrado de sus componentes es 1.

1.2.3.2. Notación bra y ket para matrices

Para toda matriz cuadrada de dimensión N a coeficientes complejos A, tenemos la siguiente representación:

$$A = \left(\underbrace{\sum_{i=1}^{N}|u_i\rangle\langle u_i|}_{I}\right)A\left(\underbrace{\sum_{j=1}^{N}|u_j\rangle\langle u_j|}_{I}\right) = \sum_{i=1}^{N}\sum_{j=1}^{N}|u_i\rangle\underbrace{\langle u_i|\overbrace{A|u_j\rangle}^{|vector\rangle}}_{\alpha_{ij}}\langle u_j| = \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{ij}|u_i\rangle\langle u_j|$$

donde α_{ij} es la componente ij de la matriz.
Con esta representación, podemos representar el producto de una matriz por un vector de la siguiente manera:

$$A|\psi\rangle = \left(\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{ij}|u_i\rangle\langle u_j|\right)\left(\sum_{k=1}^{N}a_k|u_k\rangle\right)$$
$$= \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\alpha_{ij}a_k|u_i\rangle\underbrace{\langle u_j|u_k\rangle}_{\delta_{jk}} = \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{ij}a_j|u_i\rangle$$

Es decir, las componentes del vector $A|\psi\rangle$ son $b_i = \sum_{j=1}^{N}\alpha_{ij}a_j$.

1.3. Bits cuánticos y operadores

1.3.1. Primera intuición en 8 líneas

En computación clásica la unidad mínima de información es el bit, el cual puede estar en un estado 0 o 1. Leer un bit es una operación que no conlleva ninguna particularidad. En contraposición, un bit cuántico o qubit puede estar en un estado que sea una superposición de los estados 0 y 1. Un qubit es un vector de $\mathbb{C}^2$, por lo tanto lo podemos representar como $\alpha|0\rangle + \beta|1\rangle$, lo cual representa el estado que es 0 y en 1 a la vez. Leer un qubit en cambio se produce a través de una operación llamada medición, y al medir un qubit, éste colapsa, cambia su estado (dependiendo de la medición puede cambiar por ejemplo a $|0\rangle$ o $|1\rangle$, pero también podría usarse otro operador de medición que lo colapse a otra base).

1.3.2. Bits cuánticos

Definición 1.10 (Qubit)**.** Un qubit o bit cuántico es un vector normalizado del espacio de Hilbert $\mathbb{C}^2$.

Observación. Considerando la base $\{|0\rangle, |1\rangle\}$ de $\mathbb{C}^2$, cualquier qubit puede escribirse como $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, con $|\alpha|^2 + |\beta|^2 = 1$.

Definición 1.11 (n–qubits)**.** Un sistema de n–qubits es un vector del espacio $\mathbb{C}^{2^n} = \bigotimes_{i=1}^{n} \mathbb{C}^2$.

Observaciones.

- En lugar de escribir $|0\rangle \otimes |1\rangle \otimes \cdots \otimes |0\rangle$ escribimos $|01\ldots0\rangle$.
- La base canónica del espacio $\bigotimes_{i=1}^{n} \mathbb{C}^2$ es $\{|0\ldots00\rangle, |0\ldots01\rangle, \ldots, |1\ldots11\rangle\}$.

Un algoritmo cuántico consiste en la evolución (Definición 1.19) de un sistema representado por n–qubits.

1.3.3. Operadores

Definición 1.12 (Operador)**.** Un operador de $\mathbb{C}^N$ es una matriz cuadrada de dimensión N a coeficientes complejos.

Definición 1.13 (Adjunto)**.** El adjunto de un operador A se nota por $A^\dagger$ y se define como el operador transpuesto y conjugado de A. Es decir, si $\alpha_{ij} = \langle u_i|A|u_j\rangle$ son las componentes de A, las componentes de $A^\dagger$ son $\alpha^*_{ji} = \langle u_j|A|u_i\rangle^* = \langle u_i|A^\dagger|u_j\rangle$.

Propiedades. Sean A y B operadores de $\mathbb{C}^N$, $a \in \mathbb{C}$ y $|\psi\rangle \in \mathbb{C}^N$

- $(A^\dagger)^\dagger = A$
- $(aA)^\dagger = a^* A^\dagger$
- $\langle A\psi| = \langle\psi|A^\dagger$
- $(A+B)^\dagger = A^\dagger + B^\dagger$
- $(AB)^\dagger = B^\dagger A^\dagger$

Definición 1.14 (Proyector)**.** A los operadores de la forma $P = |\phi\rangle\langle\phi|$ se les llama proyectores, ya que proyecta ortogonalmente un ket $|\psi\rangle$ cualquiera sobre el ket $|\phi\rangle$:

$$P|\psi\rangle = |\phi\rangle \underbrace{\langle\phi|\psi\rangle}_{a\in\mathbb{C}} = a|\phi\rangle$$

Ejemplo 1.4. Tomemos la base $\{|0\rangle, |1\rangle\}$, con $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$ y $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$. Un vector $|\psi\rangle$ cualquiera puede escribirse como $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Por lo tanto

$$|0\rangle\langle 0|\psi\rangle = |0\rangle\langle 0|(\alpha|0\rangle + \beta|1\rangle) = |0\rangle(\alpha \underbrace{\langle 0|0\rangle}_{1} + \beta \underbrace{\langle 1|0\rangle}_{0}) = \alpha|0\rangle$$

Definición 1.15 (Operador hermítico)**.** Un operador A es hermítico si $A = A^\dagger$.

Observación. Si es hermítico, su diagonal debe ser real, ya que $\alpha_{ij} = \alpha^*_{ji}$, por lo tanto $\alpha_{ii} = \alpha^*_{ii}$.

Definición 1.16 (Operador unitario)**.** Un operador U es unitario si $U^\dagger U = UU^\dagger = I$, o lo que es lo mismo $U^\dagger = U^{-1}$.

Propiedades. Para cualquier operador U unitario vale:

- U preserva el producto interno: $\langle U\phi|U\psi\rangle = \langle\phi|U^\dagger U|\psi\rangle = \langle\phi|\psi\rangle$
- U^{-1} es unitario.
- Si $\{|\psi_1\rangle, \ldots, |\psi_N\rangle\}$ es base ortonormal, entonces $\{U|\psi_1\rangle, \ldots, U|\psi_N\rangle\}$ también lo es.

Definición 1.17 (Operador de medición)**.** Un conjunto de proyectores $\{M_1, \ldots, M_k\}$ se dice que es un operador de medición si satisface

$$\sum_{i=1}^{k} M_i M_i^\dagger = I$$

Definición 1.18 (Compuertas cuánticas)**.** A los operadores unitarios y hermíticos se les llama compuertas cuánticas, como analogía a las compuertas lógicas de la computación clásica, ya que serán esos los que se utilizan para realizar el cómputo.

Definición 1.19 (Evolución)**.** Se dice que un sistema representado por un ket $|\psi\rangle$ evoluciona al sistema $|\phi\rangle$, cuando se realiza una de las siguientes operaciones:

- Se premultiplica por una compuerta cuántica U:

$$|\phi\rangle = U|\psi\rangle$$

- Se aplica un operador de medición $M = \{M_1, \ldots, M_k\}$ de la siguiente manera:

$$|\phi\rangle = \frac{M_i|\psi\rangle}{\sqrt{\langle\psi|M_i^\dagger M_i|\psi\rangle}} \text{ para algún } 1 \leq i \leq k$$

  La elección del M_i no se conoce de antemano, sólo se conoce la probabilidad para cada i, la cual viene dada por la siguiente ley:

$$p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle$$

Observaciones.

- Usaremos también la notación $|\psi\rangle \xrightarrow{U} |\phi\rangle$ o $|\psi\rangle \xrightarrow{M} |\phi\rangle$ para indicar que el ket $|\psi\rangle$ evoluciona al ket $|\phi\rangle$.
- Cuando se quiera hacer evolucionar sólo un qubit de un sistema de n-qubits, digamos el qubit i, se premultiplica tensorialmente $i-1$ veces y se postmultiplica $n-i-1$ veces la compuerta a aplicar por la matriz identidad. Ejemplo: U aplicada al segundo qubit de un sistema de 2–qubits, será la compuerta $I \otimes U$.

Ejemplo 1.5. Consideramos el operador medición de $\{M_0, M_1\}$ con

$$M_0 = |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad M_1 = |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Podemos verificar que $M_0 M_0^\dagger + M_1 M_1^\dagger = M_0 + M_1 = I$, y por lo tanto es un operador de medición.
Sea $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, entonces, la probabilidad de que el proyector que se aplique sea M_0 es

$$\begin{aligned} p(0) &= \langle\psi|M_0^\dagger M_0|\psi\rangle \\ &= (\alpha^*\langle 0| + \beta^*|1\rangle) M_0 (\alpha|0\rangle + \beta|1\rangle) \\ &= |\alpha|^2 \underbrace{\langle 0|\overbrace{M_0|0\rangle}^{|0\rangle}}_{1} + \alpha^*\beta \underbrace{\overbrace{\langle 0|M_0}^{\langle 0|}|1\rangle}_{0} + \alpha\beta^* \underbrace{\langle 1|\overbrace{M_0|0\rangle}^{|0\rangle}}_{0} + |\beta|^2 \underbrace{\langle 1|\overbrace{M_0|1\rangle}^{|0\rangle}}_{0} \\ &= |\alpha|^2 \end{aligned}$$

Análogamente, $p(1) = \langle\psi|M_1^\dagger M_1|\psi\rangle = \cdots = |\beta|^2$.
Dado que el vector está normalizado, $p(0) + p(1) = |\alpha|^2 + |\beta|^2 = \|\psi\| = 1$.
Luego de aplicar este operador de medición, la evolución es la siguiente. Si se aplicó el proyector M_0, el sistema queda en el siguiente estado:

$$\frac{M_0|\psi\rangle}{\sqrt{\langle\psi|M_0^\dagger M_0|\psi\rangle}} = \frac{M_0|\psi\rangle}{\sqrt{p(0)}} = \frac{\alpha}{|\alpha|}|0\rangle$$

Este estado está normalizado ya que $\left|\frac{\alpha}{|\alpha|}\right|^2 = \frac{|\alpha|^2}{|\alpha|^2} = 1$.

Análogamente si se aplicó M_1 se obtiene $\frac{M_1|\psi\rangle}{\sqrt{p(1)}} = \frac{\beta}{|\beta|}|1\rangle$.

Definición 1.20 (Compuertas más comunes y operadores de Pauli)**.** Las compuertas cuánticas más importantes, por su utilidad en el diseño de algoritmos, son las siguientes:

- La compuerta H de Hadamard:

$$\begin{aligned} H|0\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \qquad \text{donde:} \quad H = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

- La identidad I:

$$\begin{matrix} I|0\rangle = |0\rangle \\ I|1\rangle = |1\rangle \end{matrix} \quad \text{donde:} \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- La negación X:

$$\begin{matrix} X|0\rangle = |1\rangle \\ X|1\rangle = |0\rangle \end{matrix} \quad \text{donde:} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

- El cambio de fase Z:

$$\begin{matrix} Z|0\rangle = |0\rangle \\ Z|1\rangle = -|1\rangle \end{matrix} \quad \text{donde:} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- La No-controlada $CNOT$:

$$\begin{matrix} CNOT|0x\rangle = |0x\rangle \\ CNOT|1x\rangle = |1\rangle \otimes X|x\rangle \end{matrix} \quad \text{donde:} \quad CNOT = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix}$$

En particular, las matrices I, X, iXZ y Z son las llamadas *matrices de Pauli* en honor a Wolfgang Pauli

1.4. Teorema del no-clonado

El teorema de no-clonado [Wootters y Zurek, 1982] dice que es imposible hacer una máquina universal de copiado. Es decir: no podemos copiar un qubit arbitrario, ya que no existe ningún método que pueda copiarlo sin saber su estado preciso, y como la medición cambia el qubit, no podemos saber su estado preciso. En consecuencia, no podemos copiar un qubit arbitrario.

Teorema 1.2 (No-cloning)**.** No existe ninguna compuerta cuántica U tal que para algún $|\phi\rangle \in \mathbb{C}^N$ y $\forall|\psi\rangle \in \mathbb{C}^N$ se cumpla $U|\psi\phi\rangle = |\psi\psi\rangle$.

Demostración. Supongamos que existe la operación U de la cual se habla en el teorema, entonces, dados cualesquiera $|\psi\rangle, |\phi\rangle \in \mathbb{C}^N$, se cumple

$$\begin{aligned} U|\psi\phi\rangle &= |\psi\psi\rangle \\ U|\varphi\phi\rangle &= |\varphi\varphi\rangle \end{aligned}$$

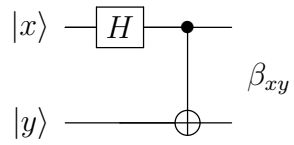
Por lo tanto, $\langle U\psi\phi|U\varphi\phi\rangle = \langle\psi\psi|\varphi\varphi\rangle$. Sin embargo, por un lado

$$\langle U\psi\phi|U\phi\varphi\rangle = \langle\psi\phi|U^\dagger U|\varphi\phi\rangle = \langle\psi\phi|\varphi\phi\rangle = \langle\psi|\varphi\rangle\langle\phi|\phi\rangle = \langle\psi|\varphi\rangle$$

Mientras por el otro $\langle\psi\psi|\varphi\varphi\rangle = \langle\psi|\varphi\rangle\langle\psi|\varphi\rangle = \langle\psi|\varphi\rangle^2$
Pero si $\langle\psi|\phi\rangle = \langle\psi|\phi\rangle^2$, entonces $\langle\psi|\phi\rangle = 0$ o $\langle\psi|\phi\rangle = 1$, lo cual es imposible: 0 implica que los dos vectores tomados al azar son ortogonales, y 1 que son iguales. □

1.5. Estados de Bell

Consideremos el siguiente *circuito* cuántico



Es decir, partiendo del estado inicial $|xy\rangle$, se aplica H al primer qubit. Luego se aplica $CNOT$ a ambos, donde el primero es el de control (marcado con el punto negro). En otras palabras, este circuito representa la siguiente ecuación:

$$\beta_{xy} = CNOT(H \otimes I)|xy\rangle$$

Las posibles salidas de este circuito, cuando x e y varían entre 0 y 1 son las siguientes:

$$|00\rangle \xrightarrow{H(1)} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \xrightarrow{CNOT(1,2)} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \beta_{00}$$
$$|01\rangle \xrightarrow{H(1)} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|1\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle) \xrightarrow{CNOT(1,2)} \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = \beta_{01}$$
$$|10\rangle \xrightarrow{H(1)} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |10\rangle) \xrightarrow{CNOT(1,2)} \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = \beta_{10}$$
$$|11\rangle \xrightarrow{H(1)} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|1\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |11\rangle) \xrightarrow{CNOT(1,2)} \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = \beta_{11}$$

Observación. $\beta_{00} = (X \otimes I)\beta_{01} = (Z \otimes I)\beta_{10} = (XZ \otimes I)\beta_{11}$.

A estos cuatro estados se les llama *Estados de Bell*, en honor a John S. Bell. Estos son estados *entrelazados*, es decir, estados que no pueden representarse como el producto tensorial de dos estados individuales (ver Sección 1.2.2.3).

A los estados entrelazados también se les llama estados EPR por Einstein, Podolsky, y Rosen [1935] quienes en detectaron, en pleno auge de las formulaciones de la teoría cuántica, que existía una acción a distancia que parecía no razonable. Por muchos años se llamó la "paradoja EPR". Lo que determinaron es que cuando se tiene un par entrelazado (físicamente el estado representa por ejemplo el *spin* en un par de electrones, o la polarización de un par de fotones), sucede que cuando se colapsa un estado del par, el segundo también colapsará, incluso cuando físicamente se encuentren a años luz de distancia. Con el tiempo se demostró experimentalmente que esto es exactamente lo que sucede, y por lo tanto no hay paradoja. También se demuestra que esto no contradice la teoría de la relatividad (que entre otras cosas determina que nada puede viajar a mayor velocidad que la luz, ni siquiera la información), ya que no hay trasmisión de información en este colapso a distancia.

Matemáticamente la acción de medir un estado de un par se ve con el siguiente ejemplo:

Ejemplo 1.6. Consideremos el siguiente operador de medición: $M = \{M_0, M_1\}$ donde $M_0 = |0\rangle\langle 0|$ y $M_1 = |1\rangle\langle 1|$.

Aplicando este operador al primer qubit del estado β_{00}, se obtiene uno de los siguientes resultados:

- Si se aplica el proyector M_0 (el cual lo expresamos como $M_0 \otimes I$ para que se aplique M_0 al primer qubit y la identidad al segundo), el estado resultante será

$$\frac{(M_0 \otimes I)\beta_{00}}{\sqrt{p(0)}} = \frac{(|00\rangle\langle 00| + |01\rangle\langle 01|)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)}{\sqrt{\frac{1}{\sqrt{2}}(\langle 00| + \langle 11|)(|00\rangle\langle 00| + |01\rangle\langle 01|)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)}}$$
$$= \frac{\frac{1}{\sqrt{2}}(|00\rangle\langle 00|00\rangle)}{\sqrt{\frac{1}{2}\langle 00|00\rangle\langle 00|00\rangle}} = |00\rangle$$

  Análogamente, si se aplica M_1 se obtiene $|11\rangle$

Es decir, al medir el primer qubit del estado entrelazado β_{00}, se obntiene $|00\rangle$ o $|11\rangle$, es decir que ambos qubits colapsan.

1.6. Usando los estados de Bell

Como se mencionó en la sección anterior, el colapso de un par entrelazado no transmite información (y por eso no viola la teoría de la relatividad), sin embargo, es posible utilizar dicho colapso como canal de comunicación, el cual necesita también de un canal clásico para terminar la transmisión (y por ende, el canal clásico implica todas las limitaciones impuestas por la relatividad).
El algoritmo cuántico descripto en la sección 1.6.1, descripto por primera vez por Bennett y Wiesner [1992], permite transmitir dos bit clásicos, enviando sólo un bit cuántico, utilizando un par entrelazado como canal de comunicación. Es llamado "codificación superdensa" ya que se trata de codificar dos bits de información en un bit cuántico, o dicho de otro modo: dos bits de información en el estado de una partícula cuántica.
El algoritmo descripto en la sección 1.6.2, descripto por primera vez por Bennett, Brassard, Crépeau, Jozsa, Peres, y Wootters [1993], permite enviar un bit cuántico enviando dos bit clásicos y utilizando un par entrelazado como canal de comunicación. Es llamado "teleportación cuántica" ya que se trata de mover el valor de un bit cuántico (recordemos que un bit cuántico no puede ser copiado (ver Teorema 1.2)) a otro bit cuántico, o dicho de otro modo: se trata de teletransportar el estado de una partícula a una nueva partícula, destruyendo la primera.

1.6.1. Codificación superdensa

El objetivo de esta técnica es transmitir 2 bits clásicos enviando tan sólo 1 qubit.
Los pasos a seguir por el emisor (a quien llamaremos "Alice") y el receptor (a quien llamaremos "Bob") son los siguientes.

1. Alice y Bob preparan un estado β_{00}.

2. Alice se queda con el primer qubit del par y Bob se lleva el segundo. Podemos considerar que estos dos pasos son la preparación del canal cuántico.

Observación. El estado entrelazado no se puede separar en el sentido de que no puede considerarse matemáticamente como un qubit multiplicado tensorialmente por otro qubit. Debemos considerarlos como un vector del espacio $\mathbb{C}^2 \otimes \mathbb{C}^2$, es decir, un vector de dimensión 4. Pero físicamente son un par de electrones, o fotones (u otra partícula elemental), las cuales sí pueden ser separadas físicamente (más allá de que no es trivial el problema experimental que representa manipular dichas partículas sin que interaccionen con el ambiente).

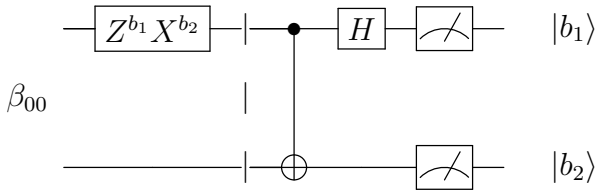
3. Alice aplica una transformación a su qubit, de acuerdo a los bits que quiere enviar: $Z^{b_1}X^{b_2}$, donde $C^0 = I$ y $C^1 = C$.

4. Alice envía su qubit a Bob.

5. Bob aplica CNOT a los dos elementos del par y luego Hadamard al primero.

6. Bob realiza una medición.

El circuito completo queda de la siguiente manera



donde la línea punteada determina el paso 4, en el que Alice envía su qubit a Bob.

Ejemplo 1.7. Se quiere enviar los bits 11. Por lo tanto se aplica $(ZX \otimes I)$ a β_{00}, con lo que se obtiene β_{11} (en general, la aplicación de la compuerta $Z^{b_1}X^{b_2}$ cambia el estado β_{00} a $\beta_{b_1 b_2}$):

$$\begin{aligned}(ZX \otimes I)\beta_{00} &= (Z \otimes I)\left((X \otimes I)\beta_{00}\right) \\ &= (Z \otimes I)\left((X \otimes I)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right) \\ &= (Z \otimes I)\left(\frac{1}{\sqrt{2}}(|10\rangle + |01\rangle)\right) \\ &= \frac{1}{\sqrt{2}}(-|10\rangle + |01\rangle) = \beta_{11}\end{aligned}$$

El resto del circuito (a partir de la línea punteada vertical) es el circuito inverso al de Bell, y como toda compuerta unitaria es tal que $U = U^{-1}$, aplicando el circuito inverso al de Bell se obtiene los estados iniciales. En este caso, $|11\rangle$.

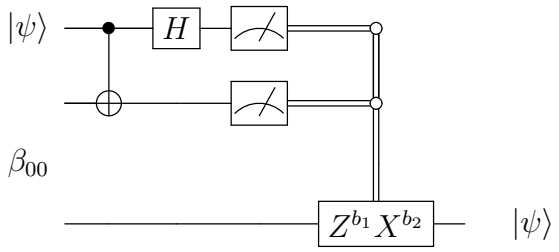
1.6.2. Teleportación cuántica

El objetivo de esta técnica es transmitir un qubit mediante el envío de dos bits clásicos. Los pasos a seguir por Alice y Bob son los siguientes.

1. Alice y Bob preparan un estado β_{00}.

2. Alice se queda con el primer qubit del par y Bob se lleva el segundo.
3. Alice aplica CNOT entre el qubit a transmitir y el primero del par β_{00}, y luego Hadamard al primero.
4. Alice realiza una medición sobre los dos qubits en su posesión y envía el resultado de la medición (2 bits clásicos) a Bob.
5. Bob aplica una transformación sobre su qubit, de acuerdo a los bits recibidos: $Z^{b_1}X^{b_2}$.

El circuito completo queda de la siguiente manera



donde $|\psi\rangle$ es el qubit a transmitir (o "teleportar").

Ejemplo 1.8. Se quiere transmitir el qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, entonces

$$\begin{aligned}
|\psi\rangle \otimes \beta_{00} &= (\alpha|0\rangle + \beta|1\rangle)\left(\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right) \\
&= \frac{1}{\sqrt{2}}(\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle)) \\
&\xrightarrow{CNOT(1,2)} \frac{1}{\sqrt{2}}(\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle)) \\
&\xrightarrow{H(1)} \frac{1}{\sqrt{2}}\left(\alpha\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)\right) \\
&= \frac{1}{2}[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)] \\
&= \frac{1}{2}\sum_{b_1=0}^{1}\sum_{b_2=0}^{1}|b_1 b_2\rangle(X^{b_2}Z^{b_1})|\psi\rangle
\end{aligned}$$

Por lo tanto, aplicando $Z^{b_1}X^{b_2}$, Bob obtendrá el estado original $|\psi\rangle$. (Nótese que para toda compuerta U, $U = U^{-1}$).

Observación. Si se quiere escribir la compuerta —$\boxed{Z^{b_1}X^{b_2}}$— como dos dos compuertas, debe escribirse —$\boxed{X^{b_2}}$—$\boxed{Z^{b_1}}$—, ya que en $Z^{b_1}X^{b_2}$ primero se aplica la compuerta X^{b_2} y luego Z^{b_1}.

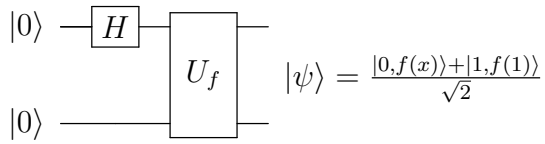
1.7. Paralelismo Cuántico

Consideremos una función $f : \{0,1\} \to \{0,1\}$. Clásicamente para obtener todos los resultados posibles de esta función, es necesario evaluarla tantas veces como sea el cardinal del dominio (2 en este caso, una evaluación para la entrada 0, y otra para la entrada 1).

Esta es una función que toma un bit y devuelve un bit. Si fuese un bit cuántico, sería posible evaluar la función en una superposición de 0 y 1 (por ejemplo $\frac{1}{2}(|0\rangle + |1\rangle)$), lo cual nos daría como resultado una superposición de f aplicada a 0 y a 1.
El método es el siguiente. Primero se debe construir una matriz unitaria U_f de $\mathbb{C}^4$ que calcule la función, de la siguiente manera:

$$U_f|x, 0\rangle = |x, f(x)\rangle$$

En realidad, aunque vamos a usar la definición que acabamos de dar, se debe definir también qué sucede cuando el segundo qubit es $|1\rangle$, por lo que esta compuerta se define más generalmente como $U_f|x, y\rangle = |x, y \oplus f(x)\rangle$, donde $\oplus$ es la suma módulo 2.
Lo que se pretende es aplicar f a todas las entradas posibles, por lo que primero se aplicará Hadamard al $|0\rangle$, a fin de obtener una superposición, y luego se aplicará la compuerta U_f. El circuito es el siguiente:



Es decir:

$$|00\rangle \xrightarrow{H(1)} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \xrightarrow{U_f} \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle)$$

La salida de este circuito es un estado que es superposición de todos los resultados posibles de la aplicación de la función f. Y la compuerta U_f fue utilizada una sola vez. El problema ahora pasa porque el resultado es una superposición de todos los resultados posibles, y al querer leer el resultado (es decir, al medirlo), éste colapsará a uno de los dos. El problema de los algoritmos cuánticos pasa por utilizar la superposición de manera inteligente para aprovechar el paralelismo, pero obteniendo el resultado buscado y no una superposición de resultados sin utilidad.
En el siguiente capítulo mostraremos algunos de los algoritmos que, haciendo uso del paralelismo, consiguen resultados óptimos.

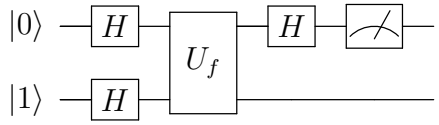
Capítulo 2

Algoritmos cuánticos más comunes y aplicación a la criptografía

En este capítulo veremos algunos de los algoritmos cuánticos más conocidos. En particular, los algoritmos de Deutsch [1985] y de Deutsch y Jozsa [1992], que pueden considerarse como los primeros algoritmos cuánticos que hacen uso del paralelismo (ver Sección 1.7). El algoritmo de Grover [1996], que es uno de los que motivó que los investigadores en computación se interesaran en el área. No se incluye el algoritmo de Shor [1997], el otro importante algoritmo que motivó a investigadores en computación a adentrarse en el área, ya que requeriría de herramientas matemáticas que escapan a los objetivos de este curso. Finalmente, el último ejemplo es una aplicación directa de la física cuántica en criptografía, diseñado por Bennett y Brassard [1984], la cual no sigue el esquema de los otros algoritmos cuánticos presentados, pero es también el puntapié de un área de investigación activa dentro de la computación cuántica.

2.1. Algoritmo de Deutsch

El objetivo de este algoritmo es saber si una función que toma un bit y devuelve un bit, es constante o no.
El algoritmo se resume en el siguiente circuito



Observación. U_f es la compuerta definida en la Sección 1.7, la cual consideraremos que existe sin dar más detalles de su construcción.

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle$$

Las primeras dos compuertas Hadamard, aplicadas a $|0\rangle$ y $|1\rangle$, producen lo siguiente:

$$|01\rangle \xrightarrow{H(1,2)} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |x\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}(|x, 0\rangle - |x, 0\rangle) \quad (2.1)$$

donde $|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ es una abreviación introducida por comodidad.

La aplicación de U_f sobre el estado (2.1) produce el siguiente estado:

$$\begin{aligned} & U_f(\frac{1}{\sqrt{2}}(|x,0\rangle - |x,1\rangle)) \\ & = \frac{1}{\sqrt{2}}\left(U_f|x,0\rangle - U_f|x,1\rangle\right) \\ & = \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|0,f(0)\rangle + |1,f(1)\rangle) - \frac{1}{\sqrt{2}}(|0,1\oplus f(0)\rangle + |1,1\oplus f(1)\rangle)\right) \\ & = \frac{1}{2}\left(|0,f(0)\rangle + |1,f(1)\rangle - |0,1\oplus f(0)\rangle - |1,1\oplus f(1)\rangle\right) \end{aligned} \tag{2.2}$$

Si $f(0) \neq f(1)$, (2.2) es igual a

$$\pm\frac{1}{2}\left(|00\rangle + |11\rangle - |01\rangle - |10\rangle\right) = \pm\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

en cambio si $f(0) = f(1)$,

$$\pm\frac{1}{2}\left(|00\rangle + |10\rangle - |01\rangle - |11\rangle\right) = \pm\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Es decir, el primer qubit es $\pm|-\rangle$, si $f(0) \neq f(1)$ y $\pm|+\rangle$ si $f(0) = f(1)$. Aplicando Hadamard al primer qubit, obtenemos $|1\rangle$ si éste era $|-\rangle$ y $|0\rangle$ si éste era $|+\rangle$.

$$\text{Si } f(0) \neq f(1), \text{ aplicando Hadamard se obtiene } \pm|1\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$
$$\text{Si } f(0) = f(1), \text{ aplicando Hadamard se obtiene } \pm|0\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

es decir, aplicando Hadamard, se obtiene

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

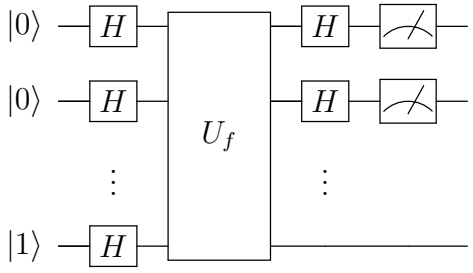
Dado que el primer qubit es $|0\rangle$ o $|1\rangle$, podemos medirlo y nos dará con probabilidad 1 el valor 1 si f es constante y con probabilidad 1 el valor 0 si f no lo es.

Observación. Este algoritmo hace uso del paralelismo, ya que la evaluación de la función se realiza una vez sobre el estado en superposición de 0 y 1. El algoritmo clásico equivalente haría dos evaluaciones de la función y una comparación.

2.2. Algoritmo de Deutsch-Jotza

Este algoritmo es una generalización del anterior. Dada una función que toma n bits y devuelve uno, el algoritmo permite distinguir si la función es constante o balanceada (o sea, con la mitad de las entradas devuelve 0 y con la otra mitad 1). Sólo se distinguen esos dos casos, el algoritmo no es útil para otro tipo de funciones.

El circuito es el siguiente:



La entrada de este algoritmo son $n+1$ qubits: $|0\rangle^{\otimes n}|1\rangle = |0\dots01\rangle$.
Aplicando las $n+1$ compuertas Hadamard sobre la entrada, se obtiene

$$\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)^{\otimes n}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) = \sum_{\overline{x}\in\{0,1\}^n}\frac{|\overline{x}\rangle}{\sqrt{2^n}}\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \tag{2.3}$$

La compuerta U_f que se utiliza es una generalización del caso anterior definida por

$$U_f|\overline{x},y\rangle = |\overline{x},y\oplus f(\overline{x})\rangle$$

donde $\overline{x}$ son cadenas de n bits.
Es decir

$$U_f|\overline{x},0\rangle = |\overline{x},f(\overline{x})\rangle \qquad\qquad U_f|\overline{x},1\rangle = |\overline{x},1\oplus f(\overline{x})\rangle$$

Por lo tanto, aplicando U_f sobre el estado (2.3) se obtiene

$$\begin{aligned}
U_f\left(\sum_{\overline{x}\in\{0,1\}^n}\frac{|\overline{x}\rangle}{\sqrt{2^n}}\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]\right) &= \sum_{\overline{x}\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}U_f|\overline{x}\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \\
&= \sum_{\overline{x}\in\{0,1\}^n}\frac{1}{\sqrt{2^{n+1}}}\left(U_f|\overline{x},0\rangle - U_f|\overline{x},1\rangle\right) \\
&= \sum_{\overline{x}\in\{0,1\}^n}\frac{1}{\sqrt{2^{n+1}}}\left(|\overline{x},f(\overline{x})\rangle - |\overline{x},1\oplus f(\overline{x})\rangle\right) \\
&= \sum_{\overline{x}\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}|\overline{x}\rangle\left(\frac{|f(\overline{x})\rangle-|1\oplus f(\overline{x})\rangle}{\sqrt{2}}\right)
\end{aligned} \tag{2.4}$$

Para simplificar la notación, la compuerta Hadamard puede expresarse como sigue

$$\left.\begin{array}{l} H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) \\ H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\end{array}\right\} \Rightarrow H|x\rangle = \frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}}(-1)^{xy}|y\rangle$$

De la misma manera, es posible generalizar la aplicación de H a n qubits como sigue:

$$\begin{aligned}
H^{\otimes n}|x_1\dots x_n\rangle &= \left(\frac{1}{\sqrt{2}}\sum_{z_1\in\{0,1\}}(-1)^{x_1z_1}|z_1\rangle\right)\cdots\left(\frac{1}{\sqrt{2}}\sum_{z_n\in\{0,1\}}(-1)^{x_nz_n}|z_n\rangle\right) \\
&= \frac{1}{\sqrt{2^n}}\sum_{\overline{z}\in\{0,1\}^n}(-1)^{\overline{x}\cdot\overline{z}}|\overline{z}\rangle
\end{aligned}$$

donde $\overline{x}\cdot\overline{z} = x_1z_1+\ldots+x_nz_n$.

Con esta notación, se aplica Hadamard a los primeros n qubits del estado (2.4) (es decir, al ket $|\overline{x}\rangle$), obteniendo

$$\begin{aligned}\sum_{\overline{x}\in\{0,1\}^n} \frac{1}{\sqrt{2^n}} \left(\frac{1}{\sqrt{2^n}} \sum_{\overline{z}\in\{0,1\}^n} (-1)^{\overline{x}\cdot\overline{z}} |\overline{z}\rangle \right) \left(\frac{|f(\overline{x})\rangle - |1 \oplus f(\overline{x})\rangle}{\sqrt{2}} \right) \\ = \sum_{\overline{x}\in\{0,1\}^n} \sum_{\overline{z}\in\{0,1\}^n} \frac{(-1)^{\overline{x}\cdot\overline{z}} |\overline{z}\rangle}{2^n} \left(\frac{|f(\overline{x})\rangle - |1 \oplus f(\overline{x})\rangle}{\sqrt{2}} \right)\end{aligned} \tag{2.5}$$

Casos:

- Si f es constante, el estado (2.5) es

  $$\pm \sum_{\overline{x}\in\{0,1\}^n} \sum_{\overline{z}\in\{0,1\}^n} \frac{(-1)^{\overline{x}\cdot\overline{z}} |\overline{z}\rangle}{2^n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

  Cuando $\overline{z} = 0$, los primeros n qubits son

  $$\pm \sum_{\overline{x}\in\{0,1\}^n} \frac{|0\rangle^{\otimes n}}{2^n} = \pm \frac{2^n}{2^n} |0\rangle^{\otimes n} = \pm |0\rangle^{\otimes n}$$

  Por lo tanto, dado que este vector tiene norma 1, el resto de los términos de la suma deben anularse, debido a que el resultado tiene que ser forzosamente un vector normalizado. Por lo tanto, cuando f es constante, el estado (2.5) es

  $$\pm |0\rangle^{\otimes n} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]$$

  Es decir, midiendo los primeros n qubits se obtiene $0 \ldots 0$ en este caso.

- Si f es balanceada (50 % de las veces devuelve 0 y 50 % devuelve 1), entonces para $\overline{z} = 0$

  $$\sum_{\overline{x}\in\{0,1\}^n} \frac{|0\rangle^{\otimes n}}{2^n} \left(\frac{|f(\overline{x})\rangle - |1 \oplus f(\overline{x})\rangle}{\sqrt{2}} \right) = \sum_{\overline{x}\in\{0,1\}^n} (-1)^{\overline{x}} \frac{|0\rangle^{\otimes n}}{2^n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) = 0$$

  Es decir que los primeros n qubits no incluyen al qubit $|0\rangle^{\otimes n}$, y por lo tanto, al medir los primeros n qubits no se puede obtener $0 \ldots 0$ en este caso.

Conclusión: Si se obtiene $|0\rangle^{\otimes n}$ a la salida de la medición, la función es constante, en otro caso la función es balanceada.

2.3. Algoritmo de Búsqueda de Grover

Antes de analizar este algoritmo, son necesarias algunas compuertas extras: la compuerta *Oráculo* (Sección 2.3.1), y la compuerta de *inversión sobre el promedio* (Sección 2.3.2).

2.3.1. Oráculo

Dada una función de un bit en un bit f, la compuerta U_f definida en la Sección 1.7, es $U_f|x,y\rangle = |x, y \oplus f(x)\rangle$.
Si se elije $y = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, entonces

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

Dado que U_f no modifica el estado y, es posible omitirlo y tomarlo como parte de la definición de la compuerta. Entonces, definimos la compuerta

$$U|x\rangle = (-1)^{f(x)}|x\rangle$$

a la cual se le llama *Oráculo*.

2.3.2. Inversión sobre el promedio

Sea el estado $|\phi\rangle = {}^1/\sqrt{2^n}\sum_{\overline{x}\in\{0,1\}^n}|\overline{x}\rangle$. Definimos la compuerta de *Inversión sobre el promedio* como $G = 2|\phi\rangle\langle\phi| - I$. Es decir

$$\begin{aligned} G &= 2|\phi\rangle\langle\phi| - I \\ &= 2\begin{pmatrix} \frac{1}{\sqrt{2^n}} \\ \vdots \\ \frac{1}{\sqrt{2^n}} \end{pmatrix}_{2^n}\left(\frac{1}{\sqrt{2^n}}\cdots\frac{1}{\sqrt{2^n}}\right)_{2^n} - I \\ &= \begin{pmatrix} \frac{2}{2^n}-1 & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \frac{2}{2^n} & \frac{2}{2^n}-1 & \cdots & \frac{2}{2^n} \\ \vdots & \vdots & & \vdots \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n}-1 \end{pmatrix}_{2^n\times 2^n} \end{aligned}$$

La aplicación de G sobre un estado cualquiera $|\psi\rangle = \sum_{\overline{x}\in\{0,1\}^n} a_{\overline{x}}|\overline{x}\rangle$ es la siguiente

$$\begin{array}{c|c} G|\psi\rangle & \begin{pmatrix} a_0 \\ \vdots \\ a_{2^n-1} \end{pmatrix} \\ \hline \begin{pmatrix} \frac{2}{2^n}-1 & \cdots & \frac{2}{2^n} \\ \vdots & & \vdots \\ \frac{2}{2^n} & \cdots & \frac{2}{2^n}-1 \end{pmatrix} & \begin{pmatrix} \left(\sum_{\overline{x}\in\{0,1\}^n} \frac{2a_{\overline{x}}}{2^n}\right) - a_0 \\ \vdots \\ \left(\sum_{\overline{x}\in\{0,1\}^n} \frac{2a_{\overline{x}}}{2^n}\right) - a_{2^n-1} \end{pmatrix} \end{array}$$

Es decir:

$$G\left(\sum_{\overline{x}\in\{0,1\}^n} a_{\overline{x}}|\overline{x}\rangle\right) = \sum_{\overline{x}\in\{0,1\}^n}\left[\left(\sum_{\overline{y}\in\{0,1\}^n}\frac{2a_{\overline{y}}}{2^n}\right) - a_{\overline{x}}\right]|\overline{x}\rangle = \sum_{\overline{x}\in\{0,1\}^n}(2A - a_{\overline{x}})|\overline{x}\rangle$$

donde A es el promedio de los $a_{\overline{x}}$.

2.3.3. El algoritmo

El algoritmo de Grover es un algoritmo de búsqueda sobre una lista desordenada. Suponemos una lista de tamaño N, con $N = 2^n$ (observar que siempre es posible aumentar la lista con datos irrelevantes para cumplir la condición sobre N). Los índices de la lista son $\overline{x} \in 0, 1^n$, es decir $\overline{x} = 0 \ldots 2^n - 1$.
El objetivo del algoritmo es localizar el $\overline{x}_0$ tal que $f(\overline{x}_0) = 1$, para una función booleana f dada.
El input del circuito es $|0\rangle^{\otimes n}$.

2.3.3.1. Paso 1: Se aplica Hadamard ($H^{\otimes n}$)

El primer paso es generar una superposición en todos los qubits.

$$|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}}\sum_{\overline{x}\in\{0,1\}^n}|\overline{x}\rangle \tag{2.6}$$

Este estado es una superposición de todos los elementos de la lista. La idea del algoritmo es subir la probabilidad de que al medir este estado obtengamos el elemento $\overline{x}_0$.

2.3.3.2. Paso 2: Se aplica el oráculo (U)

Aplicar el oráculo es el equivalente a aplicar la función booleana f sobre la superposición.

$$(2.6) \xrightarrow{U} \frac{1}{\sqrt{2^n}}\sum_{\overline{x}\in\{0,1\}^n}(-1)^{f(\overline{x})}|\overline{x}\rangle \tag{2.7}$$

2.3.3.3. Paso 3: Se aplica la inversión sobre el promedio (G)

$$
\begin{aligned}
(2.7) &= \sum_{\overline{x}\in\{0,1\}^n} \underbrace{\left[\frac{(-1)^{f(\overline{x})}}{\sqrt{2^n}}\right]}_{a_{\overline{x}}} |\overline{x}\rangle \\
&\xrightarrow{G} \sum_{\overline{x}\in\{0,1\}^n} (2A - a_{\overline{x}})|\overline{x}\rangle \\
&= \sum_{\overline{x}\in\{0,1\}^n} \left[\left(2\sum_{\overline{y}\in\{0,1\}^n} \frac{(-1)^{f(\overline{y})}}{2^n\sqrt{2^n}}\right) - \frac{(-1)^{f(\overline{x})}}{\sqrt{2^n}}\right] |\overline{x}\rangle \\
&= \sum_{\overline{x}\in\{0,1\}^n} \left[\left(2\sum_{\substack{\overline{y}\in\{0,1\}^n \\ \overline{y}\neq\overline{x}}} \frac{(-1)^{f(\overline{y})}}{2^n\sqrt{2^n}}\right) + \frac{2(-1)^{f(\overline{x})}}{2^n\sqrt{2^n}} - \frac{(-1)^{f(\overline{x})}}{\sqrt{2^n}}\right] |\overline{x}\rangle \\
&= \sum_{\overline{x}\in\{0,1\}^n} \left[\left(2\sum_{\substack{\overline{y}\in\{0,1\}^n \\ \overline{y}\neq\overline{x}}} \frac{(-1)^{f(\overline{y})}}{2^n\sqrt{2^n}}\right) + \frac{2-2^n}{2^n\sqrt{2^n}}(-1)^{f(\overline{x})}\right] |\overline{x}\rangle
\end{aligned}
\tag{2.8}
$$

En el estado (2.8), el término $\overline{x} = \overline{x}_0$, con $f(\overline{x}_0) = 1$, el cual estamos buscando es el siguiente:

$$
\begin{aligned}
\left[\left(2\sum_{\substack{\overline{y}\in\{0,1\}^n \\ \overline{y}\neq\overline{x}_0}} \frac{1}{2^n\sqrt{2^n}}\right) + \frac{2^n-2}{2^n\sqrt{2^n}}\right] |\overline{x}_0\rangle &= \left[\frac{2}{2^n\sqrt{2^n}}(2^n-1) + \frac{2^n-2}{2^n\sqrt{2^n}}\right] |\overline{x}_0\rangle \\
&= \left[\frac{2^{n+1}+2^n-4}{2^n\sqrt{2^n}}\right] |\overline{x}_0\rangle
\end{aligned}
$$

mientras que los otros términos, donde $\overline{x} \neq \overline{x}_0$, son

$$
\left[\left(2\sum_{\substack{\overline{y}\in\{0,1\}^n \\ \overline{y}\neq\overline{x}_0 \\ \overline{y}\neq\overline{x}}} \frac{1}{2^n\sqrt{2^n}}\right) + \frac{2(-1)}{2^n\sqrt{2^n}} + \frac{2-2^n}{2^n\sqrt{2^n}}\right] |\overline{x}\rangle = \left[\frac{2^{n+1}-2^n-4}{2^n\sqrt{2^n}}\right] |\overline{x}\rangle
$$

El algoritmo ha cambiado las amplitudes del estado, aumentando la amplitud del estado $\overline{x}_0$ y disminuyendo las otras.
Repitiendo este proceso (pasos 2 y 3) se va subiendo la amplitud del estado que se quiere encontrar y disminuyendo las otras. Sin embargo es cíclico: pasado cierto número de repeticiones, esa amplitud vuelve a decrecer. En la Sección 2.3.4 se calcula el número optimo de repeticiones para obtener la amplitud máxima. Cuando la amplitud es máxima, se realiza una medición, obteniendo el estado $\overline{x}_0$ con la máxima probabilidad. En la Sección 2.3.4 se muestra que la probabilidad de error tiene cota máxima en $1/2^n$.

Ejemplo

Sea una lista de $2^4 = 16$ elementos, de los que sólo uno, $\overline{x}_0$, verifica la propiedad $f(\overline{x}_0) = 1$. El algoritmo comienza por tomar el estado $|0\rangle^{\otimes 4}$ y aplicar $H^{\otimes 4}$ obteniendo,

$$\frac{1}{4}\sum_{\overline{x}\in\{0,1\}^4} |\overline{x}\rangle$$

Inicialmente todas las amplitudes son iguales a $1/4$. Se aplica el oráculo y se obtiene

$$\frac{1}{4}\sum_{\overline{x}\in\{0,1\}^4} (-1)^{f(\overline{x})}|\overline{x}\rangle$$

Luego se aplica la inversión sobre el promedio, y la nueva amplitud del estado $\overline{x}_0$ será

$$\frac{2^5 + 2^4 - 4}{2^4\sqrt{2^4}} = \frac{11}{16} = 0{,}6875$$

y para el resto de los $\overline{x}$ la amplitud será

$$\frac{2^5 - 2^4 - 4}{2^4\sqrt{2^4}} = \frac{3}{16} = 0{,}1875$$

Con las sucesivas repeticiones de la aplicación del oráculo y la inversión sobre el promedio, se obtienen las siguientes amplitudes:

Repetición	Amplitud de $\overline{x}_0$	Amplitud de $\overline{x} \neq \overline{x}_0$	Probabilidad de error
1	0.6875	0.1875	0.527
2	0.953125	0.078125	0.092
3	0.98046875	-0.05078125	0.039

A partir de la iteración 4 la probabilidad de error comienza a subir, por lo tanto el número óptimo de iteraciones es 3, con una probabilidad de error de 0,039.

2.3.4. Cálculo del número óptimo de iteraciones

Luego de k iteraciones $\overline{x}_0$ tendrá una amplitud b_k y el resto tendrán todos una amplitud m_k. Es decir, el estado será

$$b_k|\overline{x}_0\rangle + m_k \sum_{\substack{\overline{x}\in\{0,1\}^n \\ \overline{x}\neq\overline{x}_0}} |\overline{x}\rangle$$

En cada iteración se aplica el oráculo U, el cual cambia el signo de b_k, y luego G. Es posible definir recursivamente las amplitudes en la repetición k:

$$\begin{aligned} m_0 = b_0 &= \frac{1}{\sqrt{2^n}} \\ m_{k+1} &= 2A_k - m_k \\ b_{k+1} &= 2A_k + b_k \end{aligned} \qquad \text{donde} \qquad A_k = \frac{(2^n - 1)m_k - b_k}{2^n}$$

Las fórmulas cerradas para estas recursiones son

$$m_k = \frac{1}{\sqrt{2^n - 1}} \cos((2k+1)\gamma)$$
$$b_k = \text{sen}((2k+1)\gamma)$$

donde $\cos(\gamma) = \sqrt{\frac{2^n - 1}{2^n}}$ y $\text{sen}(\gamma) = \sqrt{\frac{1}{2^n}}$.
Para conseguir la mínima probabilidad de error, se debe minimizar $|m_k|$. Notar que $m_k = 0$ si y sólo si $(2k+1)\gamma = \frac{\pi}{2}$, es decir, si $k = \frac{\pi}{4\gamma} - \frac{1}{2}$.
Sin embargo, dado que k es el número de repeticiones, debe ser entero, por lo tanto, el número óptimo de iteraciones es

$$\tilde{k} = \left\lfloor \frac{\pi}{4\gamma} \right\rfloor$$

Para calcular una cota de la probabilidad de error, observar primero que que $|k - \tilde{k}| \leq \frac{1}{2}$, entonces

$$|\frac{\pi}{2} - (2\tilde{k}+1)\gamma| = |(2k+1)\gamma - (2\tilde{k}+1)\gamma| = |2\gamma(k - \tilde{k})| \leq \gamma$$

Con esto, podemos determinar que la probabilidad de error luego de $\tilde{k}$ iteraciones es

$$(2^n - 1)(m_k)^2 = cos^2((2\tilde{k}+1)\gamma) = sen^2(\frac{\pi}{2} - (2\tilde{k}+1)\gamma) \leq sen^2(\gamma) = \frac{1}{2^n}$$

En el ejemplo anterior

$$\tilde{k} = \left\lfloor \frac{\pi}{4asen(\sqrt{\frac{1}{16}})} \right\rfloor = 3$$

y la probabilidad de error es $0{,}039 \leq \frac{1}{2^4} = 0{,}0625$.

2.4. Aplicación criptográfica

2.4.1. One-time pad

Este es un método de criptografía clásica [Vernam, 1926] que consiste en compartir una secuencia de bits (clave) del largo del mensaje a transmitir y aplicar la operación reversible XOR para cifrar y descifrar. (Ver Figura 2.1). Las claves deben ser secretas y no deben ser reutilizadas.
Este método es 100 % seguro: un 0 en el mensaje encriptado puede significar un 0 en el mensaje original y un 0 en la clave, o un 1 en el mensaje y un 1 en la clave. Lo mismo sucede con un 1 en el mensaje encriptado. Es decir que adivinar la clave tiene la misma probabilidad que adivinar el mensaje original. La única debilidad de este método es la predistribución de claves, ya que el canal que se use para distribuirla podría ser vulnerado.
El método cuántico que se describe a continuación, QKD-BB84 (por *Quantum Key Distribution* de Bennett y Brassard [1984]), es justamente un método para la distribución segura de claves.

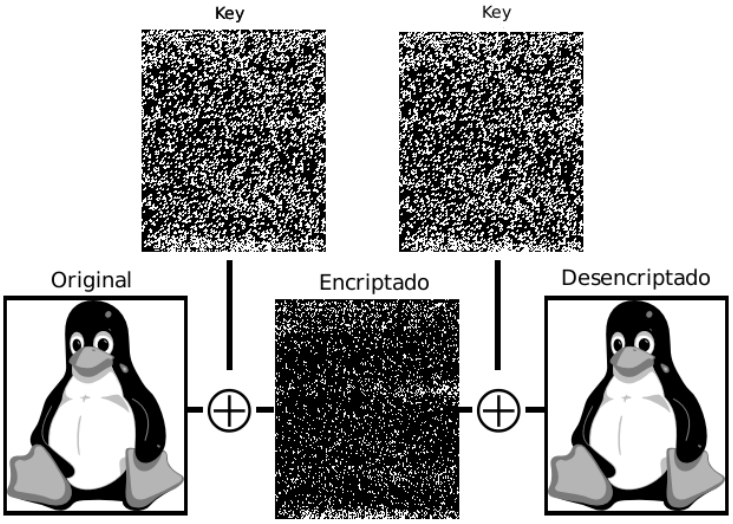


Figura 2.1: One-Time pad

2.4.2. Criptosistema Cuántico QKD-BB84

La idea es transmitir una clave binaria por un canal inseguro.
Para transmitir el bit 0, Alice (el emisor) puede elegir, al azar, la base $\{|0\rangle, |1\rangle\}$ (a la que llamaremos esquema $+$) y considerar $0 \equiv |0\rangle$, o la base $\{|-\rangle, |+\rangle\}$ (a la que llamaremos esquema $\times$) y considerar $0 \equiv |-\rangle$. Análogamente al bit 1 lo codificamos como $|1\rangle$ en el esquema $+$ o como $|+\rangle$ en el esquema $\times$.
Bob realizará una medición sobre el estado recibido eligiendo al azar entre el esquema $+$ y el esquema $\times$. Ver ejemplo en Figura 2.2. El paso final es intercambiar información (por un canal abierto) de los esquemas utilizados, y sólo conservar los bits producidos usando el mismo esquema.

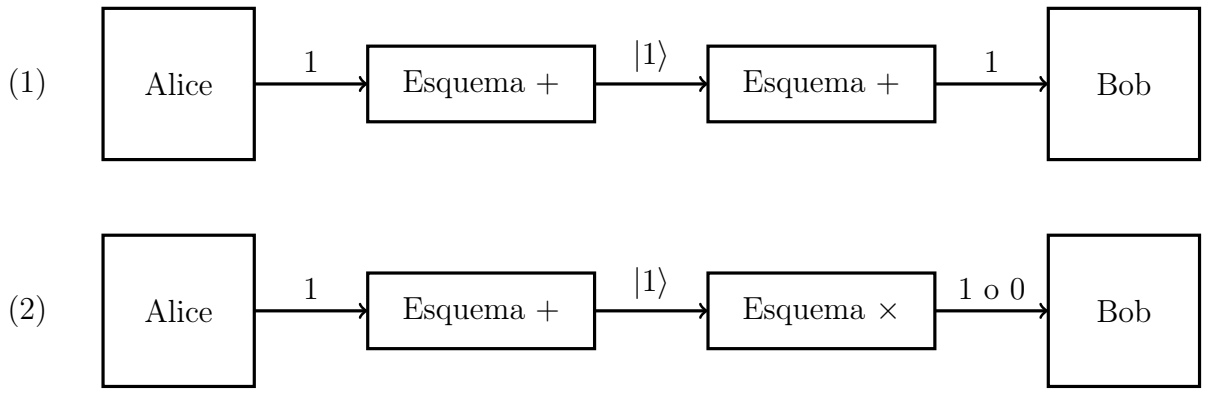


Figura 2.2: Ejemplo: (1) Alice transmite un 1 codificado mediante el esquema $+$ y Bob elije al azar el esquema $+$ obteniendo un 1 (2) si Bob elige el esquema $\times$ obtiene 0 ó 1 con probabilidad $1/2$.

El algoritmo paso a paso:

1. Alice comienza a transmitir una secuencia de 0 y 1, elegidos aleatoriamente, alternando los esquemas $+$ y $\times$ también aleatoriamente.

2. Bob recibe la secuencia y va alternando las mediciones entre los esquemas + y × aleatoriamente.

3. Alice le transmite a Bob la sucesión de esquemas empleada.

4. Bob le informa a Alice en qué casos utilizó el mismo esquema.

5. Usando solamente los bits de los esquemas idénticos a dos puntas, ambos han definido una sucesión aleatoria de bits que servirá como one-time pad de encriptación para transmisiones futuras por cualquier canal.

Esquemas de Alice	×	+	+	×	×	+	×	+
Valores de Alice	$\|-\rangle$	$\|0\rangle$	$\|0\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$	$\|-\rangle$	$\|1\rangle$
Esquemas de Bob	+	×	+	×	+	+	×	×
Valores de Bob	$\|0\rangle$	$\|+\rangle$	$\|0\rangle$	$\|+\rangle$	$\|1\rangle$	$\|0\rangle$	$\|-\rangle$	$\|-\rangle$
Coincidencias			√	√		√	√	
Clave			0	1		0	0	

6. Alice y Bob intercambian hashes de las claves (en bloques) para aceptarla o descartarla.

Inviolabilidad Este protocolo es, en teoría, inviolable. Supongamos que Cliff espía el canal de comunicación entre Alice y Bob e intenta recuperar la clave. Cliff está en la misma situación que Bob y no conoce cuál esquema es el correcto, + o ×. Por lo tanto elige al azar y se equivocará, en promedio, la mitad de las veces.
En el paso 5 Alice y Bob se ponen de acuerdo en cuáles valores tomar en cuenta (las coincidencias de la secuencia de esquemas). Esta información no le es útil a Cliff porque sólo en la mitad de las veces habrá usado el detector correcto, de manera que mal interpretará sus valores finales.
Además el QKD brinda el método para que Alice y Bob puedan detectar el potencial espionaje de Cliff:
Imaginemos que Alice envió un 0 con el esquema × (es decir, el qubit $|-\rangle$). Si Cliff usa el esquema +, colapsará el qubit a $|0\rangle$ o $|1\rangle$. Si Bob usa el esquema × y mide $|-\rangle$ coincide con lo enviado por Alice, pero si mide $|+\rangle$ Alice y Bob descubrirán esa discrepancia durante el intercambio de hashes, por lo tanto descartarán el bloque.

Bibliografía

Charles Bennett y Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. En *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, págs. 175–179. 1984.

Charles Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, y William Wootters. Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels. *Physical Review Letters*, 70(13):1895–1899, 1993.

Charles Bennett y Stephen Wiesner. Communication via one- and two-particle operators on Einstein–Podolsky–Rosen states. *Physical Review Letters*, 69(20):2881–2884, 1992.

Garret Birkhoff y John von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37(4):823–843, 1936.

Julian Brown. *The Quest for the Quantum Computer*. Touchstone, 2001.

David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

David Deutsch y Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 439(1907):553–558, 1992.

Paul A. M. Dirac. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35(03):416–418, 1939.

Albert Einstein, Boris Podolsky, y Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 44(10):777–780, 1935.

Lov K. Grover. A fast quantum mechanical algorithm for database search. En *Proceedings of the 28th Annual ACM Symposium on Theory of computing*, STOC-96, págs. 212–219. ACM, 1996.

Michael Nielsen y Isaac Chuang. *Quantum Computation and Quantum Information. 10th Anniversary Edition*. Cambridge University Press., 2010.

John Preskill. Quantum computing: pro and con. *Proceedings of the Royal Society of London A*, 454:469–486, 1998.

Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

Gilbert S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Transactions of the American Institute of Electrical Engineers*, XLV:295–301, 1926.

William K. Wootters y Wojciech .H. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, 1982.