UNIVERSIDAD DE GRANADA

Facultad de Ciencias

Doble Grado en Física y Matemáticas

Trabajo Fin de Grado

ESTRUCTURAS ALGEBRAICAS EN COMPUTACIÓN CUÁNTICA: DISEÑO DE ALGORITMOS CLÁSICOS DE SIMULACIÓN DE CIRCUITOS CUÁNTICOS

Presentado por:
D. José Alberto Azorín Puche

Curso Académico 2020-2021

Declaración de originalidad

D. José Alberto Azorín Puche

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Grado (TFG), correspondiente al curso académico 2020-2021, es original, entendida esta, en el sentido de que no ha utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a 6 de julio de 2021

Fdo: José Alberto Azorín Puche

Summary

Along the last few decades, we have experienced how technology has become a fundamental part in our daily lives, turning mobile phones and computers into our biggest allies. In the later case, it is not difficult to think about an application in any conceivable task that we may face: computers appear anywher helpping us in varied fields, such as design, numerical calculation, data science, AI or simulations, among infinite others.
All these years have proven an indisputable fact: the greater are computational developments, the greater are the challenges that humans achieve. In 1965, Gordon Moore formulated what we now remember as *Moore's Law*: the number of transistors in a dense integrated circuit doubles about every two years. This statement was observed empirically and still remains verified.
However, higher quantities of components in circuits imply tinier transistors. That is the reason why quantum effects are now starting to appear and interfere in the performance of electronic devices. Science has found a solution for this issue moving to a different computing paradigm.
Quantum computation and quantum information is the study of the information processing tasks that can be accomplished using quantum mechanical systems. This brand-new point of view for computation brings countless phenomena that classical computation does not enjoy. Here are some examples:

- Quantum parallelism: quantum systems with n qubits represent 2^n states at once, whereas classical systems can only go one by one. This is the basis for Deutsch's algorithm and Deutsch-Jozsa algorithm, which exploit this characteristic of quantum computation to accomplish an exponencial number of operations at the same time.

- Entanglement: a set of particles is said to be entangled when there is not an individual description for each of them, but they are described jointly. No matter how far they are from one another, actions over one of them have an impact on the rest. This is a very powerful resource in quantum information that can be used to make improvements on communications, helping create safer channels to transmit messages.

The machines functioning according to quantum computation principles are called quantum computers: they take advantage of quantum systems' coherence to carry out calculations. These devices have very sophisticated ways to get implemented, like ion traps or nuclear magnetic resonance. This is the source of important amount of noise, due to the interaction between the system and the environment.

To help avoid the fatal effects that noise might introduce, Quantum Error Correction (QEC) and fault-tolerant quantum computation were born: quantum computation is able to asume a limited quantity of noise and keep functioning with all the advantages it has to offer. Inside this domain, there exists a relevant class of codes, the stabilizer codes, whose main support is algebraic group theory.

In this project, the reader will get immersed in the basics of quantum computation and the stabilizer formalism, that remains a key tool in contemporary investigations. There is also a big goal we are chasing after: presenting every aspect with the rigour that theorical physics deserve. This is why one can find and introductory section to the mathematical subjects that will apear all along the document.

Once we have got at ease with all these new concepts, we expose two classical algorithms permitting simulate quantum circuits. But wait, quantum computation had promised to prevail over its classical counterpart. Why would we use classical algorithms instead of starting with quantum computers as soon as possible?

As usual, our classical mind does not let us visualize beyond what we already know. Classical computers architects can debug a device by including test conditions, monitoring registers, interrupting processing at intermediate steps, and so on. However, quantum computers get bothered by this kind of resources, given the fact that they destroy coherence. Hence, we need to find a way to design and debug a quantum computer before even trying to implement it.

Quantum architecture is not the only motivation to study classical algorithms to simulate and manipulate quantum circuits. Physicists and chemists can also profit these algorithms to simulate quantum systems before new (expensive) quantum computers come into existence.

We will understand simulation algorithms engendered by Gottesman and Knill [1], and Aaronson and Gottesman [2]. After studying how they work, we will focus on the improvements in terms of efficiency that the second one introduces: the use of symplectic products to check conmutativity relations between operators leads to a decrease of computational cost, jumping from $O(n^3)$ to $O(n^2)$.

This efficiency analysis is necessary when we talk about implementing new algorithms, as computational cost (number of operations executed) translates into time, energy and even space expenses.

In a nutshell, this paper aims to show the reader the wonders of quantum computing and give him/her/them a global perspective of how physicists and computers scientists work, in order to develop this discipline that may change our world in the same way classical computers have changed us in these last few decades.

Índice

1 Introducción

En la sociedad actual, los ordenadores se han impuesto como un compañero esencial en cualquier tarea: se dejan ver en ámbitos clásicos como la ofimática, el cálculo científico o el diseño, pero también han surgido ambientes más exóticos como las simulaciones en ciencia e ingeniería, o la ciencia de datos, en los que juegan un rol fundamental. Todos estos campos tienen algo en común, y es que con el paso del tiempo han experimentado importantes mejoras ligadas al aumento de potencia que ha ofrecido la computación en las últimas décadas.

En 1965, Gordon Moore enunciaba la conodida como *ley de Moore*, que predecía cuál sería la tendencia de mejora computacional para los ordenadores. Ésta dice que cada dos años se duplica el número de transistores en un microprocesador, conllevando esto un incremento de potencia computacional. Sin embargo, al aumentar el número de componentes, el tamaño de los mismos ha de ir haciéndose más y más pequeño, de manera que los efectos cuánticos empiezan a intervenir en el funcionamiento, y que pueden ser perjudiciales de no ser tratados con el cuidado que se merecen.

Ante la amenaza de que la ley de Moore termine por fallar, debido a las limitaciones físicas recién mencionadas, parece buena idea encontrar una solución cambiando totalmente la perspectiva desde la que se resuelven los problemas. Aquí es donde entra en juego la Computación Cuántica.

La Computación Cuántica y la Información Cuántica [1] consisten en el estudio de tareas de procesamiento de la información mediante la utilización de sistemas físicos que se rigen por las leyes de la Mecánica Cuántica. En este trabajo, se pretende sumergir al lector en este fascinante mundo empezando desde sus elementos más básicos, el qubit y las puertas cuánticas. Esto se hace en la sección 3, después de haberlo introducido en 2 en los diferentes aspectos matemáticos a tener en cuenta durante la lectura del texto.

Ya sabemos lo que es la Computación Cuántica, pero ¿cómo se traduce esto en los ordenadores que se desarrollan a partir de ella? Podemos definir un ordenador cuántico como una máquina que usa la coherencia de sistemas cuánticos para acelerar los cálculos que realiza.

Estas mejoras se consiguen por medio de procedimientos físicos muy diversos: trampas de iones, resonancia magnética nuclear, etc. Sin más que mirar la figura 1, podemos imaginar que estos procesos son bastante complicados y que los ordenadores cuánticos que los llevan a cabo son máquinas muy sofisticadas.

El problema que presentan estos aparatos es que son muy sensibles al ruido, que aparece por la interacción con su entorno. No obstante, se ha visto que la Computación Cuántica puede tolerar ciertas cantidades de ruido, sin sacrificar con ello sus ventajas computacionales. Para tal fin se crean los códigos de corrección de errores, de los que se dará un ejemplo introductorio en el apartado 3.4.2.

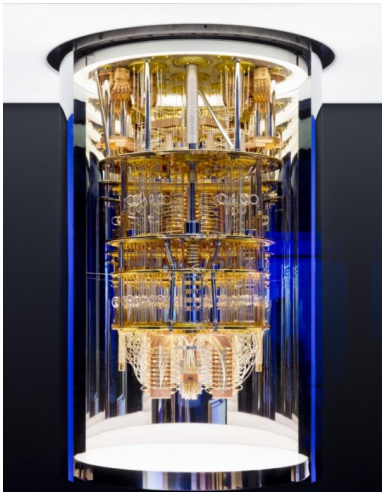


Figura 1: Ordenador Cuántico IBM Q

Dentro de los códigos de corrección de errores aparecen los códigos estabilizadores, que se definen a partir de nociones básicas de teoría de grupos. Hablamos sobre ellos en el epígrafe 3.4. El formalismo que en él se deriva será la piedra angular en el desarrollo de los dos algoritmos explicados en las secciones 4 y 5.

Dichos algoritmos nos permiten simular el comportamiento de un ordenador cuántico de manera clásica. Surge ante esto una duda más que comprensible: si tan buenos son los ordenadores cuánticos, ¿para qué pasar por una simulación suya en un ordenador clásico?

Al depurar defectos en los ordenadores clásicos, podemos monitorizar diferentes magnitudes, haciendo todas las medidas que sean necesarias, o interrumpir procesos si se estima necesario, entre otros recursos. Sin embargo, si trabajamos de esta manera con un ordenador cuántico, destruiríamos la coherencia que lo caracteriza. Por consiguiente, es interesante poder diseñar y depurar una computadora cuántica por medio de herramientas clásicas antes de vernos en la necesidad de implementarla físicamente.

Además, la arquitectura de hardware no es la única motivación que podemos encontrar para el diseño de estos algoritmos. La simulación con ordenador de sistemas cuánticos puede tener aplicaciones obvias en Física y en Química, pudiendo estudiarlos sin verse obligados a esperar a la costosa contrucción de los ordenadores cuánticos.

Un último aspecto que se abordará en estas páginas suele aparecer de forma natural al hablar de algoritmos: la eficiencia. Al ejecutar un algoritmo, se consumen recursos como pueden ser tiempo, energía e incluso espacio. Por ese motivo, en la fase de diseño se buscará minimizar el consumo de alguno de ellos (que implicará con frecuencia un ahorro del resto). En los epígrafes 4.3 y 5.3 comparamos las eficiencias de los dos algoritmos estudiados, para ver que la introducción del producto simpléctico consigue reducir de manera notable el coste computacional cuando el número de qubits en el circuito simulado es elevado.

En definitiva, el objetivo de este trabajo no es otro que el de introducir al lector en una rama de la Física y la Computación todavía joven y desconocida a día de hoy para muchos científicos. A través de un par de ejemplos detallados, se intentarán reflejar las posibilidades que nos aporta esta disciplina. No obstante, estas posibilidades son un conjunto muy restringido de todo lo que la Computación Cuántica tiene para sorprendernos.

2 Fundamentos matemáticos

Dado que la Mecánica Cuántica y la Computación son disciplinas muy teóricas y abstractas, comenzamos presentando una serie de conceptos que necesitamos asimilar.

2.1 Teoría de grupos

Comenzamos esta sección de fundamentos con uno de los imprescindibles en la lectura de este trabajo, la teoría de grupos [3]. Daremos algunas definiciones básicas que necesitamos conocer y varias herramientas que servirán para construir los algoritmos que explicamos en las secciones 4 y 5.

Definición 1. *Un **grupo** es un par ordenado $(G, *)$, donde G es un conjunto y $* : G \times G \longrightarrow G$ es una operación binaria que satisface las siguientes condiciones:*

1. *$*$ es asociativa, esto es, $a * (b * c) = (a * b) * c,\ \forall a, b, c \in G$.*
2. *Existe un elemento neutro $e \in G$ tal que $a * e = e * a = a,\ a \in G$.*
3. *Todo elemento $a \in G$ tiene inverso: $\forall a \in G,\ \exists a^1 / a * a^{-1} = a^{-1} * a = e$.*

*Si, además, se cumple que $a * b = b * a,\ \forall a, b \in G$, diremos que $(G, *)$ es un grupo **abeliano** o conmutativo.*

Un ejemplo importante que podemos dar es el grupo de todas las permutaciones sobre un conjunto $X \neq \emptyset$, que notaremos $\mathrm{Perm}(X)$ o S_n, si X es un conjunto finito. Es claro que las tres propiedades de 1 se verifican tomando la composición de permutaciones donde antes poníamos $*$.

Un paso clave en nuestro objetivo de tener algoritmos eficientes pasará por ser capaces de representar un grupo de la manera más compacta posible. Esto lo conseguimos por medio de los conocidos como **generadores**:

Definición 2. *Si A es un subconjunto del grupo G, definimos el **subgrupo de** G **generado por** A como*

$$\langle A \rangle = \bigcap_{A \subseteq H \leq G} H.$$

Si el generador $A = \{a_1, ..., a_n\}$ es un conjunto finito, escribiremos $\langle A \rangle \equiv \langle a_1, ..., a_n \rangle$, y si el subgrupo está generado por más de un conjunto, lo notaremos $\langle A \bigcup B \rangle \equiv \langle A, B \rangle$.

El siguiente resultado revela el ahorro que supone describir un grupo en términos de sus generadores, que más tarde se traducirá en una clara ventaja computacional:

Proposición 1. *Sea G un grupo finito formado por $|G|$ elementos. Entonces G está generado por un conjunto de, máximo, $\log_2 |G|$ elementos.*

Demostración. Supongamos que $g_1, ..., g_l$ es un conjunto de elementos que pertenecen al grupo G y $g \in G$ es otro elemento tal que $g \notin \langle g_1, ..., g_l \rangle$.

Tomamos entonces $f \in \langle g_1, ..., g_l \rangle$, de manera que $fg \notin \langle g_1, ..., g_l \rangle$ (pues de lo contrario $g = f^{-1}fg \in \langle g_1, ..., g_l \rangle$, conduciendo a contradicción). Por tanto, $\forall f \in \langle g_1, ..., g_l \rangle$, hay un elemento fg que está en $\langle g_1, ..., g_l, g \rangle$, pero no en $\langle g_1, ..., g_l \rangle$.

Es por ello que, añadiendo un nuevo elemento al sistema generador de un grupo, lo que hacemos es, como mínimo, doblar el tamaño del grupo generado. De esto se deduce que G ha de tener un conjunto generador de, como máximo, $\log_2 |G|$. □

Otro ejemplo de grupo son los tan conocidos espacios vectoriales, teniendo a la suma como ley de composición interna. Al aprender sobre espacios vectoriales, una de las primeras nociones que nos descubren es que hay conjuntos más pequeños dentro de ellos que mantienen características iguales, los subespacios. Ahora ocurre lo mismo:

Definición 3. *Sea G un grupo. Diremos que $H \subset G$ es un subgrupo de G si H es no vacío y cerrado bajo la operación $*$ y la inversión, esto es, $a, b \in G \Rightarrow a * b, a^{-1}, b^{-1} \in G$. En tal caso, lo notaremos $H \leq G$.*

Si H es un conjunto finito, basta con comprobar que es no vacío y cerrado para $*$ para tener la certeza de que es un subgrupo.

Para ilustrar la idea de subgrupos, vamos a definir varias estructuras que serán de gran utilidad más adelante. En todos los casos, comprobar la defición de subgrupo es sencillo:

Definición 4. *Sea G un grupo y $A \neq \emptyset$ un subconjunto suyo. Definimos el **centralizador** de A en G como el conjunto de elementos de G que conmutan con todos los elementos de A:*

$$C_G(A) = \{g \in G \mid g * a = a * g \ \forall a \in A\}.$$

*Llamaremos **centro** de G a $Z(G) = C_G(G) = \{g \in G \mid g * a = a * g \ \forall a \in G\}$.*
*Por último, si notamos $gAg^{-1} = \{gag^{-1} | a \in A\}$, se define el **normalizador de** A **en** G como*

$$N_G(A) = \{g \in G \mid gAg^{-1} = A\}.$$

Ahora que hemos introducido esta nueva estructura, necesitamos definir una operación que relacione unos grupos con otros:

Definición 5. *Sean G y H dos grupos:*
*Un **homomorfismo** de G en H es una aplicación $f : G \longrightarrow H \ / \ f(xy) = f(x)f(y), \ \forall x, y \in G$. Llamaremos a G y H dominio y codominio de f, respectivamente. Si $H = G$, diremos que f es un **endomorfismo**.*
*Diremos que f es un **isomorfismo** (respectivamente epimorfismo, monomorfismo) si f como aplicación es biyectiva (respectivamente sobreyectiva, inyectiva). Si f es un endomorfismo que es, además, isomorfismo, diremos que se trata de un **automorfismo**.*

Terminamos esta sección hablando sobre otra aplicación que será fundamental cuando hablemos del formalismo estabilizador y del algoritmo que deriva de él.

Definición 6. *Sea G un grupo y $X \neq \emptyset$ un conjunto. Una **acción** de G sobre X es una aplicación* $\mathrm{ac} : G \times X \longrightarrow X$, *que verifica las dos condiciones siguientes:*

1. $\mathrm{ac}(e, x) = x, \ \forall x \in X$.
2. $\mathrm{ac}(g, \mathrm{ac}(h, x)) = \mathrm{ac}(gh, x), \ \forall g, h \in G \ \forall x \in X$.

El caso particular en que $\mathrm{ac}(g, x) = gx$ *se conoce como **acción por traslación**, y si* $\mathrm{ac}(g, x) = gxg^{-1}$ *diremos que* ac *es una **acción por conjugación**.*

Además, resulta que dar una acción de G sobre X es equivalente a dar un homomorfismo de grupos de G en $\mathrm{Perm}(X)$. Es decir, aplicando una acción sobre un elemento $x_1 \in X$, estamos convirtiéndolo en otro $x_2 \in X$ de ese mismo conjunto (pudiendo ser él mismo).

2.2 Espacios de Hilbert

Como veremos en el Postulado 1 de la Mecánica Cuántica, el ambiente en que nos vamos a mover para describir los sistemas físicos será el de los espacios de Hilbert complejos. Recordemos que un espacio de Hilbert es un espacio prehilbertiano donde la norma asociada a su producto escalar $\langle\cdot|\cdot\rangle$[1] es completa.

Dicha elección es más que acertada: resultan ser una generalización de los tan conocidos espacios euclídeos reales a espacios complejos y espacios de dimensión infinita. Además, estos espacios garantizan la existencia de una base ortonormal y nos permitirán usar el Teorema espectral de operadores, herramientas esenciales en la formulación de esta rama de la Física. En este apartado, daremos las definiciones y resultados analíticos que aparecerán de forma continua, aunque no necesariamente de forma explícita, a lo largo de todo el trabajo. Todas las demostraciones pueden encontrarse en [4].

Definición 7. *Un **operador lineal**, o simplemente **operador**, es un homomorfismo entre espacios vectoriales normados. Notaremos $L(\mathcal{H})$ al espacio de los operadores lineales continuos de un espacio de Hilbert $\mathcal{H}$ en sí mismo. Un operador $T \in L(\mathcal{H})$ es **inversible** si $T^{-1} \in L(\mathcal{H})$.*

Dado que $L(\mathcal{H})$ es un espacio vectorial, podremos sumar y componer operadores sin que ello suponga un problema. Vayamos descubriendo algunas propiedades y cualidades que pueden presentar los operadores lineales.

Proposición 2. *Para cada $T \in L(\mathcal{H})$ hay un único opertador $T^\dagger \in L(\mathcal{H})$, llamado el **adjunto** de T, verificando que $\langle y|Tx\rangle = \langle T^\dagger y|x\rangle$, $\forall x, y \in \mathcal{H}$. Además, se verifican:*
(a) $(S+\lambda T)^\dagger = S^\dagger + \lambda^* T^\dagger$, $(ST)^\dagger = T^\dagger S^\dagger$ y $(T^\dagger)^\dagger = T$.
(b) T es inversible si y solo si $T^\dagger$ es inversible, en cuyo caso $(T^\dagger)^{-1} = (T^{-}1)^\dagger$.

A partir de la proposición 2, se deducen varias definiciones que van a tener gran relevancia cuando hablemos de observables en la sección 3:

Definición 8. *Un operador $T \in L(\mathcal{H})$ se llama **unitario** cuando $TT^\dagger = T^\dagger T = I$.*
*Se dice que un operador $T \in L(\mathcal{H})$ es **autoadjunto** o **hermitiano** cuando $T^\dagger = T$, esto es, cuando se verifica que $\langle y|Tx\rangle = \langle Ty|x\rangle$ $(x, y \in \mathcal{H})$.*

Una gran ventaja que presentan los operadores, a la hora de trabajar con ellos, es que admiten una **representación matricial**. Sea $\mathcal{H}$ un espacio de Hilbert de dimensión infinita[2] sobre un cuerpo $\mathbb{K}$ y sea $B = \{u_n : n \in \mathbb{N}\}$una base ortonormal. A cada operador $T \in L(\mathcal{H})$ podemos asociar una función $a : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{K}$, dada por

$$a(i,j) = \langle u_i | Tu_j \rangle \quad ((i,j) \in \mathbb{N} \times \mathbb{N}) \tag{2.1}$$

Podemos así pensar en dicha función como una matriz $A = (a(i,j))_{\mathbb{N}\times\mathbb{N}}$, que representa al operador T en la base B. De hecho, T está determinado de forma única por su matriz A. En esta representación, el adjunto viene representado por la matriz traspuesta complejo-conjugada de A: $A^\dagger = (A^*)^\dagger$.

Gracias a esta representación, será frecuente que hablemos indistintamente de un operador y la matriz que lo representa en una cierta base ortonormal.

[1] Lo definimos de forma que sea lineal por la derecha y que sea antilineal por la izquierda.

[2] Lo que hemos hecho para dimensión infinita puede particularizarse al caso finito: sólo hay que restringir $\mathbb{N} \times \mathbb{N}$ al conjunto $\mathcal{N} \times \mathcal{N}$, donde $\mathcal{N} = \{1, ..., d\}$, siendo d la dimensión del espacio.

Avanzamos en el camino hacia el Teorema Espectral definiendo algunos elementos que aparecerán explícitamente en el enunciado:

Sea $\mathcal{H}$ un espacio de Hilbert sobre un cuerpo $\mathbb{K}$ y $T \in L(\mathcal{H})$. Un número $\lambda \in \mathbb{K}$ es un **valor propio** o **autovalor** de T si $\ker(T - \lambda I) \neq \{0\}$. El espacio $E_\lambda = \ker(T - \lambda I)$ se llama **espacio propio** asociado al autovalor λ y sus vectores no nulos son los **vectores propios** o **autovectores** asociados a λ.

En dimensión finita, los valores propios son las raíces de la ecuación característica del operador; de ahí que el teorema fundamental del Álgebra asegure la existencia de valores propios en espacios de Hilbert complejos. Llamaremos **espectro** del operador T a su conjunto de valores propios y lo notaremos $\sigma(T)$.

Una **proyección ortogonal** en un espacio de Hilbert H es un idempotente P tal que $\ker(P) = P(\mathcal{H})^\perp$. Conocemos también un resultado que caracteriza las proyecciones sobre $P(\mathcal{H})$ como un operador idempotente autoadjunto.

La siguiente proposición da sentido a asociar proyecciones ortogonales a los valores propios de un operador:

Proposición 3. *Sea T un operador autoadjunto en un espacio de Hilbert $\mathcal{H}$.*
Si λ, μ son valores propios distintos de T, entonces los espacios propios $E_\lambda = \ker(T - \lambda I)$ y $E_\mu = \ker(T - \mu I)$ son ortogonales. Por tanto, si P_λ y P_μ son las proyecciones ortogonales de $\mathcal{H}$ sobre dichos subespacios se tiene que $P_\lambda P_\mu = 0$.

Damos una última definición que aparece como una delas hipótesis del teorema:

Definición 9. *Un operador **compacto** en un espacio de Banach X es una aplicación lineal $T : X \to X$ tal que $\overline{T(B_X)}$ es compacto en X, donde $B_X = \overline{B}(0, 1)$.*

Vamos a enunciar, al fin, el Teorema Espectral. Éste tiene dos versiones, una para el caso en que $\sigma(T)$ es un conjunto finito y otra para el caso infinito. Enunciaremos únicamente la primera, puesto que es la que se necesitará más adelante.

Teorema 1 (Teorema Espectral para operadores compactos autoadjuntos)**.**
Sea $\mathcal{H}$ un espacio de Hilbert complejo y $T \in L(\mathcal{H})$ un operador compacto autoadjunto. Equivalen las afirmaciones siguientes:

(a) $T(\mathcal{H})$ es de dimensión finita.

(b) $\sigma(T)$ es finito, es decir, T tiene un número finito de valores propios distintos λ_i, $1 \leq i \leq N$.

En tal caso, poniendo $E_k = E_{\lambda_k}$, y llamando P_k a la proyección ortogonal de $\mathcal{H}$ sobre E_k, se verifica que

$$\mathcal{H} = \bigoplus_{k=1}^{N} E_k \; ; \quad I = \sum_{k=1}^{N} P_k \; ; \quad T = \sum_{k=1}^{N} \lambda_k P_k \tag{2.2}$$

Además, si B_k es una base ortonormal de E_k, entonces $B = \bigcup_{k=1}^{N} B_k$ es una base ortonormal de $\mathcal{H}$ formada por vectores propios de T. Y si $0 \notin \sigma(T)$ entonces $\mathcal{H}$ es de dimensión finita y T es inversible.

Un operador $T \in L(\mathcal{H})$ se dice **diagonalizable** si existe una base ortogonal de $\mathcal{H}$ formada por vectores propios de T. Se tiene que todo operador autoadjunto es diagonalizable.Esto se puede traducir a las matrices:

Proposición 4. *Una matriz $A \in \mathcal{M}_{d\times d}(\mathbb{K})$ es autoadjunta si, y solo si, $A = UDU^\dagger$, donde U es unitaria y D es diagonal con escalares reales en la diagonal.*

2.2.1 Notación de Dirac

Al comenzar este apartado, notamos el producto escalar por $\langle\cdot|\cdot\rangle$. Esta notación no fue elegida por casualidad, sino que es la que se utiliza en Mecánica Cuántica para trabajar de la forma más cómoda posible. El siguiente resultado, clave en el estudio del Análisis Funcional, nos ayudará a comprender por qué:

Teorema 2 (Riesz-Fréchet)**.** *Sea $\mathcal{H}$ un espacio de Hilbert y $\mathcal{J} : \mathcal{H} \to \mathcal{H}^*$ aplicación que a cada $y \in \mathcal{H}$ hace corresponder el funcional lineal $\mathcal{J}_y : \mathcal{H} \to \mathbb{K}$ definido por*

$$\mathcal{J}_y(x) = \langle y|x\rangle \quad (x \in \mathcal{H}) \tag{2.3}$$

Se verifica que $\mathcal{J}$ es una biyección conjugado-lineal e isométrica de $\mathcal{H}$ sobre $\mathcal{H}^*$.

Este teorema dice que cada vector $y \in \mathcal{H}$ está asociado a un funcional lineal en el espacio dual[3] $\mathcal{H}^*$ que consiste en aplicar el producto escalar por y. Aunque esta notación es clara, Dirac dio con la forma de escribir estos funcionales de forma multiplicativa, y que así trabajar con ellos fuera más directo e intuitivo [5]:

Denotaremos un vector del espacio $\mathcal{H}$ como un ***ket***, $|y\rangle \in \mathcal{H}$, y al correspondiente funcional derivado del Teorema de Riesz-Fréchet lo llamaremos ***bra***, $\langle y| \in \mathcal{H}^*$. Así, la aplicación del funcional y sus propiedades lineales se comprenden de una forma mucho más natural.

$$\mathcal{J}_y(x) \equiv \langle y|\,(|x\rangle) = \langle y|\,|x\rangle \equiv \langle y|x\rangle \tag{2.4}$$

$$\mathcal{J}_y(\alpha x + \beta z) = \alpha\mathcal{J}_y(x) + \beta\mathcal{J}_y(z) \quad \Leftrightarrow \quad \langle y|\,(\alpha\,|x\rangle + \beta\,|z\rangle) = \alpha\,\langle y|x\rangle + \beta\,\langle y|z\rangle \tag{2.5}$$

Aunque $\mathcal{H}$ y $\mathcal{H}^*$ no son el mismo espacio (uno es un espacio de vectores o kets y el otro es un espacio de funcionales o bras), hay una correspondencia biunívoca entre ellos [7]:

Teorema 3. *Dada una base $B = (v_1, ..., v_n)$ de V, existe una única base $B^* = (\phi^1, ..., \phi^n)$ de $V*$ tal que $\phi^i(v_j) = \delta_i j$. Llaremos base dual de B a la base B^*.*

Además, la relación entre un ket y su correspondiente bra se describe de manera muy sencilla:

$$(|y\rangle)^\dagger = \langle y| \qquad (\langle y|)^\dagger = |y\rangle \tag{2.6}$$

Otro elemento curioso de la notación de Dirac es el **producto externo**, un operador de $\mathcal{H}$ en sí mismo que viene dado por la concatenación de un ket y un bra:

$$|x\rangle\langle y| : \mathcal{H} \to \mathcal{H} \text{ tal que } |x\rangle\langle y|\,(|z\rangle) = |x\rangle\,\langle y|z\rangle = \langle y|z\rangle\,|x\rangle \tag{2.7}$$

En 2.4 hablaremos sobre su naturaleza. Un caso particular cuando el vector $|x\rangle$ es un vector de una base ortonormal de $\mathcal{H}$ son los proyectores ortogonales:

$$E = \text{Lin}\,(|x_1\rangle, ..., |x_k\rangle) \quad \Rightarrow \quad P_E = \sum_{j=1}^{k} |x_j\rangle\langle x_j| \text{ (proyector ort. sobre } E) \tag{2.8}$$

Atendiendo a esta definición de proyector es inmediato darse cuenta de que los proyectores son operadores autoadjuntos, sin más que fijarnos en (2.6).

[3] Dado un espacio normado X sobre $\mathbb{K}$, definimos su espacio dual topológico, X^*, como el espacio $L(X, \mathbb{K})$ de todos sus funcionales lineales y continuos.

2.3 Análisis simpléctico

En este breve apartado se introduce una herramienta nueva, ahora desconocida para nosotros, pero que cada vez se utiliza más en diversos ámbitos de la Física y la Computación. Se trata de las **matrices simplécticas** [6]:

Definición 10. *Definida la matriz* $J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \in \mathbb{R}^{2n\times 2n}$, *decimos que una matriz* $M \in \mathbb{R}^{2n\times 2n}$ *es simpléctica si se verifican las siguientes igualdades:*

$$M^T J M = M J M^T = J \tag{2.9}$$

Notaremos el conjunto de todas las matrices simplécticas de dimensión $2n$ *por* $Sp(2n, \mathbb{R})$.

Como $J^T = J^{-1} = -J$, si M es simpléctica también lo será su matriz inversa S^{-1}:

$$(M^{-1})^T J M^{-1} = -(M^{-1})^T J^{-1} M^{-1} = -(M^T)^{-1} J^{-1} M^{-1} = -(M J M^T)^{-1} = -J^{-1} = J$$

$$M^{-1} J (M^{-1})^T = -M^{-1} J^{-1} (M^{-1})^T = -M^{-1} J^{-1} (M^T)^{-1} = -(M^T J M)^{-1} = -J^{-1} = J$$

Además, es trivial reconocer que el producto de matrices simplécticas es, a su vez, una matriz simpléctica. Así, se tiene que $\mathrm{Sp}(2n, \mathbb{R})$ es un grupo (con la multiplicación de matrices como ley de composición interna), aquel llamado **grupo simpléctico**.

Se puede comprobar que la definición presentada anteriormente es redundante, puesto que las igualdades 2.9 son, de hecho, equivalentes:

$$M^T J M = J \quad \Leftrightarrow \quad (M^T J M)^{-1} = J^{-1} \quad \Leftrightarrow \quad -M^{-1} J (M^T)^{-1} = -J \quad \Leftrightarrow \quad J = M J M^T$$

Así, bastará con que se verifique una de las dos para que automáticamente se cumpla la otra, simplificándose la definición 10.

Pasamos a definir otra utilidad que necesitaremos para comprender en la sección 5 cómo funciona el algoritmo: el **producto simpléctico**.

Definición 11. *Diremos que una forma bilineal es una* ***forma simpléctica*** *si es antisimétrica y no degenerada. Un caso especial es el conocido como* ***producto simpléctico*** *(o forma simpléctica estándar)* $(\cdot|\cdot) : \mathbb{R}^{2n} \times \mathbb{R}^{2n} \longrightarrow \mathbb{R}$, *definido por*

$$(z|z') = z'\, J\, z = p \cdot x' - p' \cdot x \tag{2.10}$$

donde $z = (x, p)$ *y* $z' = (x', p')$, *con* $x, x', p, p' \in \mathbb{R}^n$ *y "$\cdot$" denota el producto escalar en* $\mathbb{R}^n$.

En efecto, se verifica que $\sigma(z, z') = -\sigma(z', z)$ (antisimetría) y esto implica que todos los vectores z son *isótropos*, esto es, $\sigma(z, z) = 0$.

2.4 Tensores

En Mecánica Cuántica, espacios tensoriales y producto tensorial son nociones más que necesarias cuando se habla de sistemas compuestos por más de un subsistema, como será nuestro caso. En este apartado, los definiremos y daremos las propiedades que se usarán posteriormente. La redacción de este apartado se apoyará, sobre todo, en [7].

Definición 12. *Sean $X_1, ..., X_k$ y Y espacios vectoriales sobre un cuerpo $\mathbb{K}$. Diremos que una aplicación $M : X_1 \times X_k \to Y$ es **multilineal** si M es lineal en todos sus argumentos:*

$$\begin{aligned} M(x_1, ..., x_{j-1}, ax'_j + bx''_j, x_{j+1}, ..., x_k) = \\ = aM(x_1, ..., x_{j-1}, x'_j, x_{j+1}, ..., x_k) + bM(x_1, ..., x_{j-1}, x''_j, x_{j+1}, ..., x_k) \end{aligned} \tag{2.11}$$

para todos $x_i \in X_i,\ x'_j, x''_j \in X_j,\ 1 \leq j \leq k,\ a, b \in \mathbb{K}$.

Dado un espacio vectorial $V(\mathbb{K})$ aparece un caso particular:

Definición 13. *Un **tensor** r veces covariante y s veces contravariante (de tipo (r, s)) sobre V es una aplicación multilineal*

$$\begin{aligned} T : V \times \overset{(r)}{\cdots} \times V \times V^* \times \overset{(s)}{\cdots} \times V^* &\to \mathbb{K} \\ (u_1, ..., u_r, \phi^1, ..., \phi^s) &\mapsto T(u_1, ..., u_r, \phi^1, ..., \phi^s). \end{aligned} \tag{2.12}$$

Definiendo de forma natural la suma y el producto por escalar de tensores (sumando y multiplicando por escalar las correspondientes imágenes), se tiene que $\mathcal{T}_{r,s}(V)$, el conjunto de tensores de tipo (r, s) sobre $V(\mathbb{K})$, tiene estructura de espacio vectorial.

Es claro que $\mathcal{T}_{1,0} = V^*$ y $\mathcal{T}_{0,1} = V^{**}$. El ambiente en que vamos a trabajar es con V siendo un espacio de Hilbert y de dimensión finita: tanto en un caso como en otro se tiene que podemos identificar V con V^{**} vía un isomorfismo isométrico, de manera que podemos identificar $\mathcal{T}_{0,1}(V) = V$. Dicho con otras palabras, podemos identificar cada vector $v \in V$ con un tensor 1-contravariante

$$v : V^* \to \mathbb{K} \quad , \quad \phi \mapsto \phi(v).$$

Introducimos ahora el producto tensorial, una herramienta que nos permitirá crear tensores nuevos aumentando los índices r o s.

Definición 14. *Sean $T \in \mathcal{T}_{r,s}(V)$ y $T' \in \mathcal{T}_{r',s'}(V)$. Se define el **producto tensorial** de T por T' como $T \otimes T' : V \times \overset{(r+r')}{\cdots} \times V \times V^* \times \overset{(s+s')}{\cdots} \times V^* \to \mathbb{K}$, donde*

$$(T \otimes T')(u_1, ..., u_{r+r'}, \phi^1, ..., \phi^{s+s'}) = T(u_1, ..., u_r, \phi^1, ..., \phi^s) \cdot T'(u_{r+1}, ..., u_{r+r'}, \phi^{s+1}, ..., \phi^{s+s'}).$$

Listamos a continuación algunas propiedades inmediatas deducidas de la definición:

1. $T \otimes T'$ es multilineal y, por tanto, $T \otimes T' \in \mathcal{T}_{r+r',s+s'}(V)$.
2. $\otimes$ es lineal en cada variable:

$$\begin{aligned} (aT_1 + bT_2) \otimes T' = a(T_1 \otimes T') + b(T_2 \otimes T')\,, \quad \forall T_1, T_2 \in \mathcal{T}_{r,s}, \forall T' \in \mathcal{T}_{r',s'} \\ T \otimes (aT'_1 + bT'_2) = a(T \otimes T'_1) + b(T \otimes T'_2)\,, \quad \forall T \in \mathcal{T}_{r,s}, \forall T'_1, T'_2 \in \mathcal{T}_{r',s'}. \end{aligned}$$

3. La operación producto tensorial es asociativa(aunque no conmutativa).

Un ejemplo interesante es el de los **tensores 1-covariantes, 1-contravariantes**. Se trata de los tensores de orden 2 que constituyen el espacio vectorial $\mathcal{T}_{1,1}$ y que se definen a partir de dos tensores $\phi \in V^*$ y $u \in V$ de orden 1. Para ello, se considera su producto tensorial

$$\begin{aligned} \phi \otimes u : V \times V^* &\to \mathbb{K} \\ (v, \psi) &\mapsto \phi(v) \cdot \psi(u). \end{aligned} \tag{2.13}$$

A este espacio pertenecen, pues, los productos externos (2.7) definidos con la notación de Dirac. De nuevo, vemos que esta notación permite trabajar de forma más intuitiva con elementos como este.

Fijando una base $B = \{v_1, ..., v_n\}$ de V, y otra $B^* = \{\varphi^1, ..., \varphi^n\}$ del dual V^*, definimos $B_{1,1} = \{\varphi^1, ..., v_j \,|\, i, j = 1, ..., n\}$, que es una base de $\mathcal{T}_{1,1}$, con dimensión n^2.

Un último comentario sobre estos tensores tan especiales es que guardan una potente relación con los endomorfismos. Mientras que $\mathcal{T}_{2,0}$, $\mathcal{T}_{0,2}$ y $\mathcal{T}_{1,1}$ son todos isomorfos al espacio $\mathrm{End}(V)$ de los endomorfismos de $V(\mathbb{K})$, sólo se tiene para el último un isomorfismo que no depende de las bases.

Teorema 4. *Definido el tensor $T_f(v, \phi) := \phi(f(v))$, se tiene que la aplicación*

$$\begin{aligned} End(V) &\to \mathcal{T}_{1,1}(V) \\ f &\mapsto T_f \end{aligned},$$

es un isomorfismo de espacios vectoriales. Además, para cualquier base B de V, se verifica que

$$M_B(T_f)^4 = M(f, B) \tag{2.14}$$

De este teorema se deduce, pues, que la matriz asociada a un endomorfismo y un tensor de $\mathcal{T}_{1,1}$ pueden ser identificados, sea cual sea la base en la que se trabaje.

Acabamos presentado la forma más general posible de los tensores: los **tensores de tipo** (r, s). Considerando una base $B = \{v_1, ..., v_n\}$ de V y su dual $B^* = \{\varphi^1, ..., \varphi^n\}$ de V^*, se define

$$B_{r,s} = \{\varphi^{i_1} \otimes ... \otimes \varphi^{i_r} \otimes v_{j_1} \otimes ... \otimes v_{j_s} \,|\, i_1, ..., i_r, j_1, ..., j_s = 1, ..., n\}. \tag{2.15}$$

Se puede demostrar que $B_{r,s}$ es una base de $\mathcal{T}_{r,s}(V)$ y que $\dim(\mathcal{T}_{r,s}(V)) = n^{r+s}$.

2.4.1 Producto de Kronecker

Ahora que ya sabemos que los tensores de orden 2 pueden ser representados por medio de matrices, vamos a revelar una forma muy cómoda de operar productos tensoriales de matrices. Un desarrollo más profundo puede encontrarse en [8].

Sea un conjunto de endomorfimos $a^{(j)} : V \longrightarrow V$, con $j = 1, ..., s$. Cada uno de ellos estará asociado a una matriz (que notaremos $A^{(j)} = M(a^{(j)}, B)$), ¿cómo hacer que actúen todos ellos sobre vectores del espacio $\mathcal{T}_{0,s}(V)$? La operación que estamos buscando se llama **producto de Kronecker** y viene dada por

$$\mathbf{A} := \bigotimes_{j=1}^{n} A^{(j)} : \mathcal{T}_{0,s}(V) \longrightarrow \mathcal{T}_{0,s}(V),$$

donde

$$\bigotimes_{j=1}^{n} v^{(j)} \in \mathcal{T}_{0,s}(V) \quad \longmapsto \quad \mathbf{A}\left(\bigotimes_{j=1}^{n} v^{(j)}\right) = \bigotimes_{j=1}^{n} \left(A^{(j)} v^{(j)}\right) \in \mathcal{T}_{0,s}(V). \tag{2.16}$$

[4] Esta matriz se define coordenada a coordenada como $\left(M_B(T_f)\right)_{ij} = \left(T_f(v_j, \varphi^i)\right)_{ij}$

Obsérvese que **A** sigue siendo la matriz que representa un endomorfismo, por lo que seguirá estando asociada a tensores de orden 2. Sin embargo, la base sobre la que actúa es otra: **A** actúa sobre vectores de $\mathcal{T}_{0,s}(V)$, que tiene dimensión n^s, por lo que **A** será una matriz cuadrada de orden n^s.

También hay que señalar que hemos definido el producto de Kronecker sobre elementos que son productos tensoriales de vectores, no lo hemos definido para elementos cualesquiera de $\mathcal{T}_{0,s}(V)$. Así, para poder calcular el efecto de **A** sobre cualquier vector de $\mathcal{T}_{0,s}(V)$, habrá que usar las propiedades de linealidad que dimos tras la definición 14.

La forma en que se definen los productos tensoriales y el producto de Kronecker nos permiten dar el producto tensorial de dos matrices de una forma muy sencilla. Dadas dos matrices cuadradas $A, B \in \mathcal{M}_n(\mathbb{C})$, su producto tensorial se calcula como

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \dots & a_{nn}B \end{pmatrix} \tag{2.17}$$

Una vez hemos aprendido lo más básico sobre producto tensorial, concluimos dando varios resultados sobre él que necesitaremos más adelante. Todos ellos atenderán a matrices cuadradas, que será lo más útil para nosotros, pero tienen una generalización a otras dimensiones y también un formalismo tensorial que ignoraremos.

Proposición 5. *Sean A y B matrices cuadradas en $\mathcal{M}_n(\mathbb{C})$. Entonces:*

1. $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$, $(A \otimes B)^T = A^T \otimes B^T$ y $(A \otimes B)^* = A^* \otimes B^*$.
2. $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$.
3. $(AC) \otimes (BD) = (A \otimes B)(C \otimes D)$, *donde* $C, D \in \mathcal{M}_n(\mathbb{C})$.

Como consecuencia inmediata de la segunda afirmación se deduce el siguiente

Corolario 1. *Si U y V son matrices unitarias, entonces su producto tensorial $U \otimes V$ también es una matriz unitaria.*

Demostración. Como U y V son unitarias se tiene que $U^{-1} = U^\dagger$ y $V^{-1} = V^\dagger$. Usando las propiedades 2 y 3, tenemos $(U \otimes V)^{-1} = U^{-1} \otimes V^{-1} = U^\dagger \otimes V^\dagger = (U \otimes V)^\dagger$. □

Presentamos ahora un teorema cuya demostración puede encontrarse en [9]:

Teorema 5. *Sean $A \in \mathcal{M}_{n_1}(\mathbb{C})$ y $B \in \mathcal{M}_{n_2}(\mathbb{C})$ dos matrices cuadradas. Si $\{\lambda_i,\ i = 1, ..., n_1\}$ y $\{\mu_j,\ j = 1, ..., n_2\}$ son los espectros de autovalores de A y B, respectivamente, entonces el conjunto de autovalores de $A \otimes B$ es el conjunto $\{\lambda_i \mu_j \,|\, i = 1, ..., n_1,\ j = 1, ..., n_2\}$.*

Además, el vector propio de $A \otimes B$ asociado al autovalor $\lambda_i \mu_j$ es el producto tensorial de los vectores propios asociados a λ_i y μ_j, para cada par (i, j).

De este teorema, se deduce un interesante resultado:

Corolario 2. *Sean $A \in \mathcal{M}_{n_1}(\mathbb{C})$ y $B \in \mathcal{M}_{n_2}(\mathbb{C})$. Se tiene entonces que*

$$\mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A)\mathrm{Tr}(B) = \mathrm{Tr}(B \otimes A).$$

Demostración. La traza de una matriz es igual a la suma de sus valores propios. Siguiendo las hipótesis del teorema 5 se tiene entonces que:

$$\mathrm{Tr}(A \otimes B) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \lambda_i \mu_j = \sum_{i=1}^{n_1} \lambda_i \sum_{j=1}^{n_2} \mu_j = \mathrm{Tr}(A)\mathrm{Tr}(B).$$ □

3 Introducción a la Computación Cuántica

Ahora que conocemos los conceptos matemáticos que aparecerán incesantemente, abordamos los aspectos más esenciales de la Computación Cuántica, sin los cuales no seríamos capaces de comprender los algoritmos que se presentarán posteriormente.

3.1 Postulados de la Mecánica Cuántica

Todo el formalismo de la Mecánica Cuántica puede describirse a partir de unos pocos postulados que resultan de la experimentación, y su validez se asume debido a que toda la teoría que deriva de ellos está en total consonancia con la Naturaleza. Estos postulados [10], además de servir como puente para entender la relevancia de la sección 2, suponen la base para contruir el formalismo nmatemático para describir los procesos físicos.

Postulado 1 Estado de un sistema

Un sistema físico está asociado a un espacio de Hilbert complejo y separable, y un estado puro del sistema (en el instante t) viene descrito por un vector unitario, representado por un ket $|\psi(t)\rangle$ de dicho espacio. [5]

Postulado 2 Observables y operadores

Todo observable de un sistema físico (posición, momento, potencial, etc.) se representa por un operador lineal autoadjunto actuando en el espacio de Hilbert asociado, cuyos vectores propios forman una base completa del mismo.

Postulado 3 Medida de los observables

La medida de un observable A (sobre un cierto estado $|\psi(t)\rangle$) se representa por la acción del operador asociado sobre el correspondiente vector unitario. Por tanto, los únicos valores posibles son los valores propios $\{a_n\}$ (que son reales) del operador. Si el resultado de la observación es a_k, el estado colapsa inmediatamente, proyectándose sobre el subespacio propio asociado a a_k, generado por los vectores propios $\left\{\left|a_k^{(i)}\right\rangle\right\}$:

$$A\,|\psi(t)\rangle = a_k \left|a_k^{(i)}\right\rangle \quad \Rightarrow \quad |\psi(t')\rangle = \frac{P_{A,a_k}\,|\psi(t)\rangle}{\|P_{A,a_k}\,|\psi(t)\rangle\,\|} \tag{3.1}$$

donde $P_{A,a_k} = \sum_i \left|a_k^{(i)}\right\rangle \left\langle a_k^{(i)}\right|$, siendo discreto el espectro de A.

Postulado 4 Resultado probabilístico de la medida

Si el observable A tiene un espectro discreto, la probabilidad al medirlo de obtener el autovalor a_k (posiblemente degenerado) viene dado por

$$p(a_k) = \frac{\sum_i \left|\left\langle a_k^{(i)}\middle|\psi\right\rangle\right|^2}{\langle\psi|\psi\rangle} \tag{3.2}$$

[5] Se deduce el **Principio de Superposición**: cualquier superposición de estados puros es, a su vez, un estado puro del sistema, entendiéndose superposición como una combinación lineal $\sum_n a_n|\psi_n(t)\rangle$ con coeficientes complejos tales que $\sum_n |a_n|^2 = 1$.

Postulado 5 Evolución temporal de un sistema

La evolución temporal de un estado $|\psi(t)\rangle$ está regida por la ecuación de Schrödinger independiente del tiempo:

$$i\hbar\frac{\partial|\psi(t)\rangle}{\partial t} = \mathcal{H}|\psi(t)\rangle \ , \tag{3.3}$$

donde $\mathcal{H}$ es el operador hamiltoniano, que describe la energía total del sistema.

3.1.1 El operador densidad

El formalismo de la Mecánica Cuántica, tal y como lo hemos presentado, deriva predicciones estadísticas sobre un conjunto (una colección) de sistemas físicos idénticamente preparados, todos ellos caracterizados por un mismo estado $|\psi\rangle$. Por ejemplo, imaginemos un haz de átomos de plata, todos con el mismo estado de espín, como el que resulta del experimento de Stern-Gerlach. En este caso, el haz está polarizado; sin embargo, sabemos que previo al experimento, la orientación de los espines no está polarizada, tenemos un conjunto completamente aleatorio de estados.

¿Cómo tratar, pues, la situación en que haya distintos estados para los sistemas de la preparación? En tal caso, se definen los **estados mezcla** como una colección de sistemas dentro de la cual, una fracción de los elementos presentan un estado $|\psi_1\rangle$ y la fracción restante presenta otro, $|\psi_2\rangle$. Matemáticamente, el estado mezcla se describe por medio de la conocida como **matriz de paridad**[6] :

$$\rho = \omega_1|\psi_1\rangle\langle\psi_1| + \omega_2|\psi_2\rangle\langle\psi_2| \qquad \text{donde } \ \omega_1+\omega_2 = 1. \tag{3.4}$$

Los pesos ω_1 y ω_2 serían las proporciones (expresadas en tanto por uno) de cada uno de los estados presentes en la mezcla. Obsérve que en caso de que sólo hubiera un estado, el operador densidad $\rho = |\psi_1\rangle\langle\psi_1|$ estaría refieriéndose a un estado puro. Este operador es autoadjunto y cumple la condición de normalización $\mathrm{Tr}(\rho) = 1$.

Ahora que contamos con esta nueva herramienta para describir una situación más general que la anterior, es necesaria una reformulación de los postulados 3 y 4 recogidos en la página anterior:

Postulado 3′ Si un sistema físico está en un estado mezcla descrito por una matriz de densidad ρ y medimos un observable A, obteniéndose el valor propio a, el sistema se transforma en un estado mezcla descrito por la matriz de densidad

$$\rho_{A,a} = \frac{P_{A,a}\rho P_{A,a}}{\mathrm{Tr}(\rho P_{A,a})} \tag{3.5}$$

Postulado 4′ Si un sistema físico se encuentra en un estado descrito por una matriz de densidad ρ, entonces la probabilidad de obtener el valor propio a de un observable A es

$$p_a = \mathrm{Tr}(\rho P_{A,a}) \tag{3.6}$$

[6] Para un estado mezcla donde aparecen k estados distintos, definiríamos la matriz de paridad como $\rho = \sum_{i=1}^{k}\omega_i|\psi_i\rangle\langle\psi_i|$, con la ligadura $\sum_{i=1}^{k}\omega_i = 1$.

3.2 Qubits y puertas cuánticas

A continuación, pasamos a describir con detalle los que serán los elementos esenciales de la Computación Cuántica: el qubit y las diferentes puertas cuánticas. Gracias a ellos, es posible realizar una analogía muy clara entre circuitos eléctricos clásicos y los circuitos cuánticos que trataremos de simular.

3.2.1 El qubit: la unidad básica de información

Es conocido que la forma más simple de almacenar información en un ordenador clásico es el **bit**, que toma valores binarios, esto es, que solo puede encontrarse en dos estados (0 y 1). Además, dichos estados son físicamente realizables con componentes muy sencillos (interruptor abierto/cerrado, bombilla encendida/apagada, etc.) y, de ahí que, para observar en qué estado se halla, baste con observar el bit.

La estructura análoga en Computación Cuántica es el **qubit**, o bit cuántico, dado por un sistema cuántico que presenta dos estados básicos $|0\rangle$ y $|1\rangle$ (pensemos, por ejemplo, en un átomo que puede estar en su estado energético fundamental o en un estado excitado, o en un electrón con dos posibles estados de espín), de los que hay que destacar que conforman un sistema ortonormal.

La clave de esto es que, mientras el bit clásico (llamémoslo b) es un elemento $b \in \{0,1\} \cong \mathbb{Z}_2$, el qubit es un vector unitario en un espacio de Hilbert $\mathcal{H}$ complejo y separable, como rezaba el Postulado 1. Así, el qubit puede encontrarse en uno de los estados básicos o una superposición $|\psi\rangle \in \mathcal{H}$ cualquiera de ellos:

$$|\psi\rangle = \alpha_1 |0\rangle + \alpha_2 |1\rangle \quad / \quad |\alpha_1|^2 + |\alpha_2|^2 = 1 \tag{3.7}$$

Nace así, de forma natural, el modelo matemático que nos permitirá trabajar con este nuevo elemento: los vectores $|0\rangle$ y $|1\rangle$ constituyen una base del espacio de Hilbert asociado al sistema, que en nuestro caso será $\mathcal{H} = \mathbb{C}^2$.

A partir de ahora, identificaremos la base $\mathcal{B}_1 = \{|0\rangle, |1\rangle\}$ con la base canónica, $\{(1,0),(0,1)\}$, de manera que podremos escribir los estados en superposición (3.7) en la forma (α_1, α_2).

Conviene, antes de continuar, presentar una segunda base de $\mathbb{C}^2$ que también es relevante y conocida. Se trata de aquella formada por los "estados equiprobables":

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \overset{\mathcal{B}_1}{=} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \quad , \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \overset{\mathcal{B}_1}{=} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \tag{3.8}$$

En conclusión, los qubits pueden existir en todo un continuo de estados entre $|0\rangle$ y $|1\rangle$ hasta el instante en que son observados, cuando colapsan, pero nunca podremos conocer (por medio de la observación) las amplitudes que los describen.

3.2.2 Operando sobre los qubits: puertas y medida

Una vez hemos presentado la forma de representar la información, es inevitable preguntarse cómo podemos manipularla: dado que los estados son vectores de $\mathbb{C}^2$, las operaciones que emplearemos con ellos vienen dadas por matrices de $\mathbb{C}^{2\times 2}$. A estas operaciones

las llamaremos **puertas cuánticas**, volviendo al paralelismo con la computación clásica y sus puertas lógicas.

Sin embargo, no todas las matrices son válidas: al aplicar una puerta cuántica sobre un qubit, se obtiene un nuevo estado, de forma que la ligadura $|\alpha_1|^2 + |\alpha_2|^2 = 1$ ha de cumplirse antes y después de la aplicación. Como la norma ha de conservarse (los vectores son unitarios), las puertas cuánticas sólo podrán venir dadas por matrices unitarias[7]. Se tiene además el recíproco: toda matriz unitaria puede actuar como una puerta cuántica (en particular, la identidad).

A continuación, se presentan las puertas cuánticas[8] (sobre un qubit) que jugarán algún papel dentro del algoritmo con el que vamos a trabajar:

Matrices de Pauli

Estas tres matrices, muy empleadas en Mecánica Cuántica, suponen los cimientos sobre los que se construye el formalismo estabilizador, que estudiaremos más adelante. Son las siguientes:

$$X \equiv \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad , \quad Y \equiv \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad , \quad Z \equiv \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.9}$$

Las puertas X y Z gozan de gran relevancia y tienen una interpretación muy sencilla de comprender:

X es el equivalente cuántico de la puerta lógica NOT: convierte $|0\rangle$ en $|1\rangle$ y viceversa. Sus valores propios son +1 y -1, siendo sus autovectores asociados los estados equiprobables, $|+\rangle$ y $|-\rangle$, respectivamente. Resumiendo:

$$X \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} \alpha_2 \\ \alpha_1 \end{pmatrix} \quad , \quad \begin{matrix} X|+\rangle = +|+\rangle \\ X|-\rangle = -|-\rangle \end{matrix} \tag{3.10}$$

Por su parte, lo que hace Z es invertir el signo de la amplitud asociada al vector $|1\rangle$, dejando invariante la amplitud del otro. Sus vectores propios son los estados básicos, $|0\rangle$ y $|1\rangle$, correspondientes a los valores propios +1 y -1. En definitiva,

$$Z \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ -\alpha_2 \end{pmatrix} \quad , \quad \begin{matrix} Z|0\rangle = +|0\rangle \\ Z|1\rangle = -|1\rangle \end{matrix} \quad . \tag{3.11}$$

Puerta Hadamard

Se trata de una matriz que convierte la base de los estados básicos en la base de los estados equiprobables (3.8). Como H es su propia inversa, resulta que también transforma los estados equiprobables en los estados básicos:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad , \quad \begin{matrix} H|0\rangle = |+\rangle & H|+\rangle = |0\rangle \\ H|1\rangle = |-\rangle & H|-\rangle = |1\rangle \end{matrix} \tag{3.12}$$

Puerta de fase

[7] Si U es una puerta cuántica y $U|\psi\rangle = |\psi'\rangle$, se tiene $1 = \langle\psi'|\psi'\rangle = \langle\psi|U^\dagger U|\psi\rangle = \langle\psi|\psi\rangle \Rightarrow U^\dagger U = I$.

[8] Físicamente, estas puertas se implementan buscando el hamiltoniano $\mathcal{H}$ (ver Postulado 5) que consiga modificar el sistema como nos interesa: cambiar su nivel de energía, invertir su espín, etc.

Se trata de un caso particular de las denominadas puertas de desplazamiento de fase, R_ϕ. Estas puertas dejan invariante el estado base $|0\rangle$, pero añaden una cierta fase a $|1\rangle$. No modifican la probabilidad de medir un estado básico o el otro. La puerta de fase P, aquella que nos interesa, es la que aporta una fase de $\phi = \frac{\pi}{2}$:

$$R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} \quad \Rightarrow \quad P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad , \quad \begin{aligned} P|0\rangle &= |0\rangle \\ P|1\rangle &= i|1\rangle \end{aligned} \tag{3.13}$$

Otros ejemplo de puertas de desplazamiento de fase serían la puerta $R_{\frac{\pi}{4}}$ (que llamamos puerta $\frac{\pi}{8}$), o la puerta Z, que resulta ser $Z = R_\pi$) .

Medida del qubit

Después de haber manipulado un qubit, aplicando sobre él las puertas cuánticas convenientes y relacionándolo con otros qubits del circuito, decidimos que queremos medir su estado. Este proceso se representará con una puerta de medida.

Como consecuencia del Postulado 3 y el Postulado 4, cuando observemos un qubit, éste colapsará al estado $|0\rangle$ con probabilidad $|\alpha_1|^2$ o al estado $|1\rangle$ con probabilidad $|\alpha_2|^2$.

Las "puertas" que se emplean para medir los qubits son, por tanto, los proyectores asociados a cada uno de los estados básicos. Sin embargo, la medida no podrá ser considerada como puerta cuántica, ya que es singular y no es unitaria. Pero, ¿qué problema hay con que sea singular? Dado que una puerta viene dada por una matriz unitaria U, su inversa será su matriz adjunta $U^\dagger$; así, cualquier puerta cuántica está representada por una matriz que ha de ser forzosamente reversible. De hecho, esta es una de las grandes ventajas respecto de la Computación Clásica que se introducen con la Computación Cuántica: una vez se ha realizado una operación sobre el qubit, podemos deshacer nuestros pasos sin más que aplicar la misma operación invertida.

3.2.3 Extensión a sistemas de varios qubits

Comencemos pensando lo que ocurre con bits clásicos. Si tomamos el caso en que sólo tenemos 2 bits (cada uno de ellos con dos estados posibles $\{0, 1\}$), entonces el sistema conjunto podrá adoptar cualquiera de las 4 combinaciones distintas que podemos formar con ellos : 00, 01, 10, 11 (dos interruptores abiertos, uno abierto y otro cerrado, etc.).

Volviendo a los bits cuánticos, el planteamiento es el mismo: un sistema de 2 qubits podrá encontrarse en los estados básicos $|00\rangle$, $|01\rangle$, $|10\rangle$ y $|11\rangle$. Sin embargo, el Principio de Superposición vuelve a abrir una brecha entre bits clásicos y cuánticos: si el sistema puede hallarse en cualquiera de los estados básicos, también puede estar en una superposición de ellos.

$$|\psi\rangle = \alpha_1|00\rangle + \alpha_2|01\rangle + \alpha_3|10\rangle + \alpha_4|11\rangle \quad / \quad |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 + |\alpha_4|^2 = 1$$

Generalicemos esto para un sistema de n qubits:

Los posibles estados básicos en los que podemos encontrar el sistema se corresponden con todas las combinaciones de longitud n posibles que podemos crear a partir de valores binarios. En el caso clásico, los estados son elementos de $\{0, 1\}^n \cong \mathbb{Z}^n$.

Para el caso cuántico, la formulación matemática es algo más compleja: cada estado básico del sistema de n qubits vendrá dado por el producto tensorial de los diferentes

estados básicos de las componentes que lo conforman. Por ejemplo, tomando $n = 6$, un posible estado básico sería $|001011\rangle = |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle$. Se deduce, por tanto, que los estados básicos de un sistema de n qubits (y por tanto cualquier superposición de ellos) serán elementos del espacio vectorial complejo de dimensión 2^n obtenido como el producto tensorial $\mathcal{T}_{0,n}(\mathbb{C}^2) = \mathbb{C}^2 \otimes \overset{n}{\ldots} \otimes \mathbb{C}^2 \equiv \left(\mathbb{C}^2\right)^{\otimes n}$.

Siguiendo (2.15), sabemos que podemos obtener la base de $\mathcal{T}_{0,n}(\mathbb{C}^2) = \left(\mathbb{C}^2\right)^{\otimes n}$ calculando los productos tensoriales de las 2^n combinaciones posibles de estados básicos. Ordenaremos la base siguiendo la denominada "ordenación canónica": si empezamos a contar desde 0, cada estado se halla en la posición que le corresponde por su traducción de código binario al sistema decimal.

Lo comprenderemos mejor con un ejemplo. Un sistema de 3 qubits se representa con el espacio 8-dimensional $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$. Lo describiremos con la siguiente base (ordenada):

$$\{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |101\rangle, |111\rangle\}$$

Así, un estado cualquiera de este sistema vendrá expresado como

$$|\psi_3\rangle = \alpha_1 |000\rangle + \alpha_2 |001\rangle + \alpha_3 |010\rangle + \alpha_4 |011\rangle + \alpha_5 |100\rangle + \alpha_6 |101\rangle + \alpha_7 |110\rangle + \alpha_8 |111\rangle ,$$

donde los cuadrados de los módulos de los coeficientes suman 1.

Es aquí donde se puede empezar a atisbar otra de las bondades de la Computación Cuántica frente al modelo clásico: con n bits clásicos podemos representar un sólo estado, mientras que si empleamos n qubits, podemos almacenar 2^n estados diferentes a la vez.

Puertas de varios qubits

Al igual que antes, sentimos la necesidad de aprender a manipular los qubits con las distintas puertas cuánticas. Ahora, es normal preguntarse si hay alguna manera de relacionar los qubits que componen el sistema, haciendo que el estado de unos afecten a lo que ocurre con otros.

En Computación Clásica, son conocidas puertas lógicas como AND, OR, XOR, NAND y NOR, que toman dos bits de entrada y devuelven uno de salida aplicando una cierta operación lógica sobre los valores que reciben. El prototipo de puerta cuántica para varios qubits es **cNOT** (o ***controlled*-NOT**, también llamada *controlled-X*). Ésta se aplica sobre un par de qubits: uno de ellos, a, es el elemento de control, mientras que el otro, b, es el objetivo (o *target*). La puerta cambia el valor lógico de b sólamente cuando el qubit de control se encuentra en el estado básico $|1\rangle$. Por supuesto, esta puerta admite una representación matricial como cualquier otra:

$$U_{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad , \qquad \begin{array}{|c|c|c|} \hline \text{a} & \text{b} & \text{cNOT}|ab\rangle = |a\ b \oplus a\rangle \\ \hline 0 & 0 & |00\rangle \\ 0 & 1 & |01\rangle \\ 1 & 0 & |11\rangle \\ 1 & 1 & |10\rangle \\ \hline \end{array} \tag{3.14}$$

Esta puerta no es más que la versión cuántica de la puerta XOR, que aplica la suma exclusiva (o suma modulo 2) de los bits que recibe. Sin embargo, no deja de ser un caso particular de toda una familia de puertas, las ***controlled-U***, que dejan invariante el qubit

de control y aplican la puerta unitaria U sobre el *target* cuando el primero se halla en el estado $|1\rangle$:

$$U = \begin{pmatrix} x_1 & x_2 \\ x_3 & x_4 \end{pmatrix} \qquad \Rightarrow \qquad cU \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x_1 & x_2 \\ 0 & 0 & x_3 & x_4 \end{pmatrix} \tag{3.15}$$

En esta misma línea, existen incluso puertas con varios qubits de control. Estas puertas modifican el estado del target en caso de que todos los controles se encuentren en $|1\rangle$. El ejemplo más destacable es la puerta de Toffoli cuántica, que no es más que una ccZ (*controlled-controlled-Z*).

¿Y qué ocurre cuando haya que aplicar una **puerta sobre un sólo qubit**? Previamente, presentamos una serie de puertas cuánticas que estaban representadas por matrices de $\mathbb{C}^{2\times 2}$, adecuadas para alterar un solo qubit. Sin embargo, ahora los estados están representados por vectores de dimensión 2^n, de manera que estas matrices dejan de ser válidas. Será necesario recurrir a matrices, todavía unitarias, de $\mathbb{C}^{2^n\times 2^n}$.

Cada vez que queramos aplicar alguna puerta sobre algún qubit, hemos de aplicar el producto tensorial de n puertas, cada una sobre su correpondiente qubit. En caso de que queramos aplicar U sobre uno y solamente uno de los n qubits, digamos el j-ésimo, la operación consistirá en el producto tensorial de $n-1$ matrices identidad por la matriz U (esta última en la posición j). Para operar con más qubits, hacemos lo mismo, colocando matrices identidad en las posiciones correspondientes a los qubits que queramos dejar invariantes. Veamos un ejemplo:

Dado un sistema de $n = 3$ qubits y un estado $|\psi_3\rangle$ como el antes visto, queremos aplicar, a la vez, una puerta X sobre el primer qubit y una Hadamard sobre el tercer qubit. Llamaremos U al operador con el que modificamos el sistema completo (que será una matriz cuadrada de dimensión $2^3 = 8$), y escribiremos cada puerta acompañada de un subíndice denotando el qubit sobre el que actúan. U viene dada, entonces, por

$$U = X_1 \otimes I_2 \otimes H_3 = \left[\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right] \otimes H_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\Longrightarrow \quad U|\psi_3\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \alpha_4 + \alpha_5 \\ \alpha_4 - \alpha_5 \\ \alpha_7 + \alpha_8 \\ \alpha_7 - \alpha_8 \\ \alpha_1 + \alpha_2 \\ \alpha_1 - \alpha_2 \\ \alpha_3 + \alpha_4 \\ \alpha_3 - \alpha_4 \end{pmatrix}$$

Vemos que lo que hacen estas puertas, aunque no actúen sobre todos los qubits, es modificar las amplitudes (y por tanto las probabilidades) con las que se presentan cada uno de los vectores de la base.

3.3 Circuitos cuánticos

En las páginas que preceden, hemos descubierto los elementos que se emplean en Computación Cuántica con fines diversos (cálculo, simulaciones, etc.). Para utilizarlos, podemos integrarlos en un "circuito cuántico", como en el ejemplo que se representa en la siguiente figura:

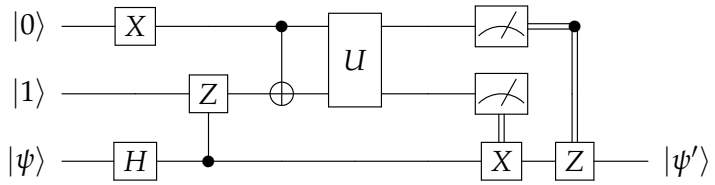


Figura 2: Ejemplo de circuito cuántico

Desgranemos su contenido:

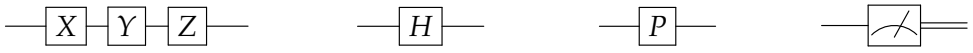
A cada qubit del sistema le hacemos corresponder una línea horizontal, sobre la que se situarán las distintas acciones que pretendemos realizar sobre él. Los circuitos cuánticos se interpretan de izquierda a derecha y, leyendo según este orden, podemos distinguir tres fases diferenciadas:

1. La preparación de estados previa a la ejecución del circuito. A la izquierda de cada una de las líneas mencionadas, se indica el estado inicial del qubit correspondiente. Es habitual partir desde el estado básico $|00...00\rangle$, pero conviene indicarlo siempre.

2. El cuerpo del circuito, compuesto por todas las puertas lógicas que actúan sobre los qubits y las diferentes relaciones que se establecen entre ellos. Es en esta parte cuando el estado de los n qubits se ve alterado.

3. La salida que resulta del circuito. Aunque se parte conociendo el estado de n qubits, puede interesarnos un número menor de estados como output. Esto es lo que ocurre en la Figura 2, ya que solo obtenemos el estado final del tercer qubit, habiendo "perdido" los dos restantes.

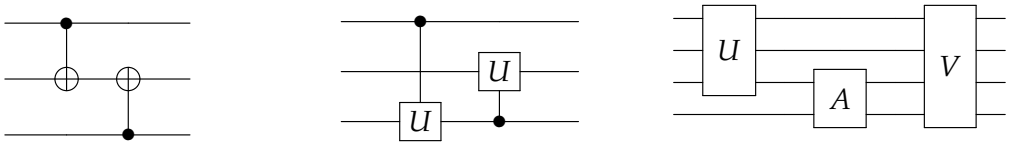
Sigamos con esta exploración sobre los circuitos cuánticos pasando por las diferentes puertas cuánticas que podemos representar:



(a) Matrices de Pauli (b) Hadamard (c) Fase (d) Medida

Figura 3: Puertas cuánticas sobre un sólo qubit.

La puerta de medida merece especial atención, ya que aparece un elemento nuevo: un "cable clásico". Avanzamos en 3.2.2 que, ante la medida del qubit, éste colapsa sobre uno de los dos estados básicos, $|0\rangle$ ó $|1\rangle$, y esta operación es la única irreversible entre todas las que podemos encontrar. Estos hechos fuerzan la necesidad de una forma alternativa de representar el canal por el que comunicamos el estado de dicho qubit. Así, la puerta de medida siempre irá seguida de una doble línea horizontal delatando este nuevo cable.



(a) Puertas cNOT (b) Puertas *controlled-U* (c) Puertas sobre varios qubits

Figura 4: Puertas cuánticas sobre varios qubits

En las puertas controladas se observa que hay una línea vertical uniendo las líneas correspondientes a los qubits que asocia. El punto negro se coloca sobre el cable correspondiente al qubit de control, mientras que el otro extremo es el *target*. La operación realizada por la puerta cNOT se representa por $\oplus$, aunque podríamos poner una puerta X en su lugar.

3.4 El formalismo estabilizador

La comunicación de información, codificada en qubits, está sujeta a un ruido que puede alterar el estado de alguno de ellos, provocando que el mensaje de llegada difiera de aquel que envió el emisor. Para tratar de corregir estos posibles errores, existe multitud de códigos de detección y corrección de errores.

Dentro de estos códigos, podemos encontrar un tipo de gran relevancia, que va a suponer el centro en torno al cual gira este trabajo: los códigos estabilizadores. Detrás de ellos, se esconde todo un formalismo que encuentra en la teoría de grupos su principal apoyo.

3.4.1 El grupo estabilizador

Volviendo a las definiciones que dimos en la base teórica sobre grupos, hay una que no dimos entonces porque era oportuno reservarla para este apartado:

Definición 15. *Dado un grupo G y un conjunto X, podemos considerar para cada $x \in X$*

$$\mathrm{Stab}(x) = \{g \in G \mid ac(g, x) = x\},$$

*al que llamaremos **estabilizador** de x en G (o grupo de isotropía de G).*

Esta definición es directamente aplicable a la situación que necesitamos abordar: dado un estado puro $|\psi\rangle$, decimos que la matriz unitaria U **estabiliza** a $|\psi\rangle$ si $|\psi\rangle$ es un vector propio de U con valor propio 1, esto es, $U|\psi\rangle = |\psi\rangle$. Esto será coherente con la definición 15 si consideramos la acción por traslación de $U_{\mathbb{C}}(2^n) \equiv \mathcal{U}_n$ sobre $\left(\mathbb{C}^2\right)^{\otimes n}$.

El estabilizador de un cierto estado puro $|\psi\rangle$ en $\mathcal{U}_n$ constituye un subgrupo dentro del grupo unitario $\mathcal{U}_n$ al que llamaremos **grupo estabilizador** y notaremos $\mathrm{Stab}\left(|\psi\rangle\right)$. Comprobemos que cumple los requisitos para ser un subgrupo:

Es claro que la matriz identidad, I, estabiliza cualquier estado puro. Obsérvese también que si dos matrices U y V estabilizan un estado $|\psi\rangle$, también lo harán su producto

y sus matrices inversas:

$$\left.\begin{array}{l} U|\psi\rangle = |\psi\rangle \\ V|\psi\rangle = |\psi\rangle \end{array}\right\} \quad \Rightarrow \quad UV|\psi\rangle = U(V|\psi\rangle) = U|\psi\rangle = |\psi\rangle$$

$$U|\psi\rangle = |\psi\rangle \quad \Rightarrow \quad |\psi\rangle = U^{-1}U|\psi\rangle = U^{-1}(U|\psi\rangle) = U^{-1}|\psi\rangle$$

Esta idea puede extenderse: un grupo puede estabilizar más de un estado. Dado un grupo estabilizador G, denominaremos **código estabilizador** asociado a G al subespacio $\mathcal{V} := \{|\psi\rangle : \sigma|\psi\rangle = |\psi\rangle \ \ \forall \sigma \in G\}$ formado por los estados que deja invariante.

La idea fundamental del formalismo estabilizador es abandonar la representación de un estado cuántico $|\psi\rangle$ a partir de un vector de amplitudes (como hemos estado haciendo hasta ahora), para pasar a representarlo a partir de su grupo estabilizador.[9] Esto podría parecer poco práctico, puesto que se pasa de necesitar 2^n coeficientes, en el primer caso, a 2^{2n} para representar cada una de las matrices en el segundo. Sin embargo, vamos a ver que somos capaces de encontrar una representación mucho más compacta.

Es el momento de recordar las matrices de Pauli que conocimos en 3.9. Es sencillo comprobar que se cumplen las siguientes igualdades:

$$\begin{array}{lll} X^2 = +I & Y^2 = +I & Z^2 = +I \\ XY = +iZ & YZ = +iX & ZX = +iY \\ YX = -iZ & ZY = -iX & XZ = -iY \end{array} \tag{3.16}$$

En particular, se tiene que las matrices de Pauli conmutan o anticonmutan dos a dos. A la vista de estas igualdades, resulta evidente que la identidad, junto con las matrices de Pauli, multiplicadas por $\{\pm 1, \pm i\}$, forman un subgrupo de $U_{\mathbb{C}}(2)$ (con el producto de matrices).

En caso de que tratemos con un sistema de n qubits, los operadores que se pueden formar a partir de ellas vienen dados por los diferentes productos tensoriales de n matrices de Pauli (o identidad) posibles. A estos operadores los llamaremos **operadores de Pauli sobre n qubits**. Por las propiedades del producto tensorial, se garantiza que cualquier producto de estos operadores da lugar a otro operador de Pauli. En definitiva, la identidad y los operadores de Pauli sobre n qubits, multiplicados por un factor de $\{\pm 1, \pm i\}$ también forman un grupo:

$$\mathcal{P}_n = \left\{ u \cdot \bigotimes_{i=1}^{n} P_i \ / \ u \in \{\pm 1, \pm i\} \ , \ P_i \in \{I, X, Y, Z\} \ \forall i \in \{1, ..., n\} \right\} \tag{3.17}$$

A este grupo lo llamaremos **Grupo de Pauli** sobre n qubits, que consta, claramente, de $|\mathcal{P}_n| = 4^{n+1}$ elementos. La ley de composición interna con la que se define este grupo no es otra que la composición de operadores, esto es, el producto de matrices unitarias de dimensión 2^n.

Después de estas definiciones, estamos en condiciones de presentar el siguiente teorema [2], que nos aporta el grupo de operadores con el que vamos a trabajar.

[9] En lugar de hacer evolucionar el vector que describe al sistema, pasamos a interesarnos por la evolución de los operadores que actúan sobre él. Así, abandonamos una representación de Schrödinger para adoptar una representación de Heisenberg.

Teorema 6. *Dado un estado $|\psi\rangle$ de n qubits, las siguientes afirmaciones son equivalentes:*

(i) *$|\psi\rangle$ se puede obtener a partir del estado $|0\rangle^{\otimes n}$ por medio de únicamente la acción de puertas cNOT, Hadamard y de fase.*

(ii) *$|\psi\rangle$ se puede obtener a partir del estado $|0\rangle^{\otimes n}$ por medio de únicamente la acción de puertas cNOT, Hadamard, de fase y de medida.*

(iii) *$|\psi\rangle$ es estabilizado por exactamente 2^n operadores de Pauli.*

(iv) *$|\psi\rangle$ queda completamente determinado por $S(|\psi\rangle) = Stab(|\psi\rangle) \bigcap \mathcal{P}_n$ (grupo de los operadores de Pauli que estabilizan a $|\psi\rangle$) .*

Las afirmaciones (iii) y (iv) resumen, pues, la representación que tomaremos a partir de ahora: para hablar de un estado de n qubits, usaremos el grupo de 2^n operadores de Pauli que lo estabilizan. Como un grupo G cualquiera tiene un sistema generador de tamaño $\log_2 |G|$ (proposición 1), nos bastará con tener n operadores para describir $S(|\psi\rangle)$ por completo. Veámoslo con un ejemplo para el caso $n = 3$:

Tomando el estado $|011\rangle$, por ejemplo, es sencillo darse cuenta que el grupo que lo estabiliza es

$$S(|011\rangle) = \{I,\ Z_1,\ -Z_2,\ -Z_3,\ -Z_1Z_2,\ -Z_1Z_3,\ Z_2Z_3,\ Z_1Z_2Z_3\},$$

pero es sufiente con conocer únicamente tres de sus elementos, ya que el resto se pueden generar a partir de ellos: $S(|011\rangle) = \langle Z_1,\ -Z_2,\ -Z_3\rangle$.

Prestemos atención a la notación que vamos a emplear a partir de ahora: se omitirán los símbolos de producto tensorial, $\otimes$ y se indicará a qué qubit está afectando la puerta que se escriba. Además, en caso de que sea la identidad la que está actuando sobre un qubit, también será omitida. Veamos algunos ejemplos para comparar y comprender mejor esta novedad:

$$Z_1 \equiv Z_1 \otimes I_2 \otimes I_3 \quad , \quad -Z_1Z_3 \equiv Z_1 \otimes I_2 \otimes (-Z_3) \quad , \quad I \equiv I_1 \otimes I_2 \otimes I_3$$

Antes de acabar, hay que destacar que el único código estabilizado por la matriz $-I$ es el subespacio trivial, por lo que nunca la incluiremos en el grupo estabilizador de los estados que representaremos. También es interesante darse cuenta de que $\mathcal{S} \equiv S(|\psi\rangle)$ **es un grupo abeliano**, sin importar el estado $|\psi\rangle$ que consideremos [12]. Para ver esto, pensemos en un estado $|\psi\rangle \neq 0$ que es invariante bajo la acción por traslación de todo elemento de $\mathcal{S}$ y consideremos $\sigma, \tau \in \mathcal{S}$ dos operadores de Pauli arbitrarios del estabilizador. Por las igualdades 3.16, sabemos que $\sigma\tau = u\tau\sigma$, donde $u \in \{\pm 1, \pm i\}$. Aplicando dos veces la definición de estabilizador se sigue $|\psi\rangle = \sigma\tau\,|\psi\rangle = u\tau\sigma\,|\psi\rangle = u\,|\psi\rangle$, de manera que $|\psi\rangle = u\,|\psi\rangle$ y se tiene forzosamente que la fase $u = +1$. Se tiene por tanto que $\sigma\tau = \tau\sigma$.

Puertas de Clifford

En el Teorema 6, en (i) y (ii) concretamente, se entiende que serán las puertas cNOT, Hadamard y de fase aquellas que vamos a emplear de aquí en adelante. ¿Por qué éstas y no otras? Resulta que, al aplicar estas puertas, conseguimos que los operadores de Pauli que representan nuestro estado cuántico permanezcan dentro del grupo de Pauli

(ahondaremos sobre esto en la sección 4), de ahí su capital importancia para los códigos estabilizadores.

Debido a su relevancia, les pondremos nombre: las puertas cNOT, Hadamard y de fase son las conocidas como **puertas de Clifford**, y los circuitos cuánticos que cuentan únicamente con ellas para operar serán los circuitos de Clifford. Si, además, incluimos también puertas de medida, se obtienen los denominados **circuitos estabilizadores**, llamando **estado estabilizador** a cualquier estado obtenido desde $|0\rangle^{\otimes n}$ por medio de un circuito estabilizador. Veremos algunos ejemplos en 3.4.2.

Un apunte necesario sobre las puertas de Clifford tiene que ver con la **complejidad** de los circuitos. El siguiente resultado supone una pieza fundamental en la simulación de circuitos cuánticos:

Teorema 7 (Gottesman-Knill)**.** *Sea un circuito cuántico constituido únicamente por puertas de Clifford y medidas dentro del grupo de Pauli, siendo la preparación uno de los estados básicos del sistema. Dicho circuito puede simularse de forma eficiente en un ordenador clásico.*

Por tanto, este teorema nos garantiza que toda simulación que hagamos de circuitos estabilizadores implicará tiempos de orden polinomial, siendo realista plantearse la realización clásica de dichas simulaciones.

Además de los circuitos estabilizadores, hay otro ámbito en el que las puertas de Clifford adquieren relevancia: la **universalidad**. En computación clásica, decimos que una puerta lógica es universal si toda operación puede ser implementada usándola sólamente a ella. Ejemplos de puertas lógicas universales son la puerta NAND, que es irreversible, o la puerta de Toffoli para computación reversible. En Computación Cuántica, esta definición se verá modificada:

Definición 16. *Un conjunto de puertas se dice* ***universal para la computación cuántica*** *si toda operación unitaria puede ser aproximada con precisión arbitraria por un circuito cuántico compuesto únicamente por puertas de dicho conjunto.*

Si bien es cierto que las puertas de Clifford no constituyen un conjunto de puertas universal, sí que forman parte de los conjuntos más sencillos que se pueden formar: las puertas de Clifford, junto con la puerta $\pi/8$ (que, recordemos, es la rotación $R_{\frac{\pi}{4}}$), forman un conjunto capaz de aproximar tanto como se quiera cualquier operación unitaria.

Otro ejemplo de conjunto de puertas universal lo constituyen la puerta de Toffoli (algo más compleja que la rotación $\pi/8$) acompañada, también ahora, por las puertas de Clifford. En definitiva, podemos imaginar que, a pesar de no poder representar cualquier circuito, el formalismo estabilizador nos permite estudiar un amplio abanico de ejemplos interesantes.

3.4.2 Ejemplos de circuitos estabilizadores

El objetivo de este apartado es el de transmitir al lector la importancia de las puertas de Clifford y el formalismo estabilizador. Podría parecer que, restringiendo tanto las operaciones a usar, estaríamos sacrificando algunas de las ventajas fundamentales de la Computación Cuántica, pero vamos a ver que estas herramientas son, de hecho, el origen de muchas de estas ventajas.

1. Generación de estados entrelazados

Un sistema de varias partículas se dice que está en estado de **entrelazamiento** cuando es imposible describir cada partícula del sistema de forma independiente sin que ello suponga una pérdida de información. Matemáticamente, esto se traduce en términos de separabilidad: un estado en entrelazamiento no puede expresarse como producto tensorial de los estados de cada uno de los subsistemas.

El primer ejemplo de circuito estabilizador a mostrar consigue generar estados (de n qubits) en entrelazamiento empleando únicamente una puerta Hadamard y $n-1$ puertas cNOT:

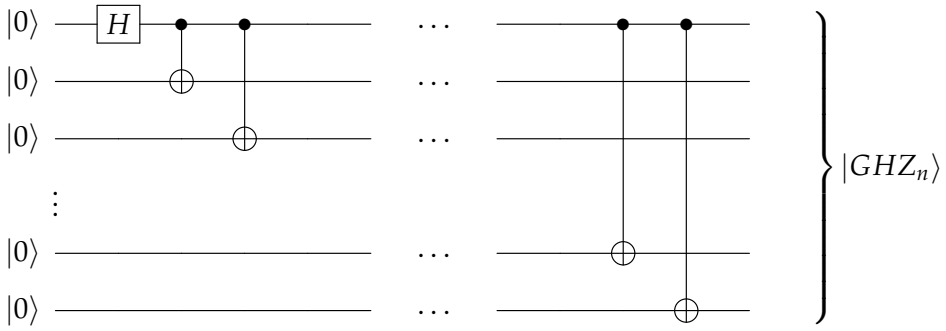


Figura 5: Circuito generador de estados GHZ

Este circuito toma como input el estado básico $|0\rangle^{\otimes n}$ y lo transforma en el estado GHZ (Greenberger-Horne-Zeilinger) de n qubits:

$$|GHZ_n\rangle = \frac{|0\rangle^{\otimes n} + |1\rangle^{\otimes n}}{\sqrt{2}}, \qquad n \geq 2 \tag{3.18}$$

Estos estados son la forma más simple de observar entrelazamiento en sistemas de n qubits. Además de la definición explícita dada en (3.18), podemos dar una descripción muy elegante de ellos gracias a los generadores de su grupo estabilizador:

$$S_1^{(GHZ_n)} = \prod_{k=1}^{n} X_k \qquad , \qquad S_k^{(GHZ_n)} = Z_{k-1}Z_k \ \text{ para } k \in \{2, ..., n\}. \tag{3.19}$$

En el caso de un sistema de $n = 2$ qubits, el circuito presentado nos permite conseguir un conjunto importante de estados, conocido como la base de estados de Bell (o pares EPR[10]). El circuito transforma cada uno de los cuatro estados básicos en un par EPR:

$$\begin{aligned} |00\rangle \mapsto |\beta_{00}\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} & |01\rangle \mapsto |\beta_{01}\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\ |10\rangle \mapsto |\beta_{10}\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}} & |11\rangle \mapsto |\beta_{11}\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}} \end{aligned} \tag{3.20}$$

El entrelazamiento es un recurso muy importante de la Computación Cuántica, ya que son precisamente las correlaciones entre estados las que conducen a su potencia. Además, elementos como los estados de Bell tienen aplicaciones incluso en transmisión de la información, intuyéndose futuras mejoras en criptografía y comunicación.

[10] En honor a Eintein, Podolsky y Rosen, por la "paradoja" con el mismo nombre, que ponía de manifiesto las peculiaridades del entrelazamiento cuántico.

2. Teleportación Cuántica

Este sorprendente fenómeno, se introdujo en 1993 por medio de una historia [14]. Imaginemos que un par de amigos, Alice y Bob, crearon un par EPR $|\beta_{00}\rangle$ antes de que Bob se mudara a un lugar muy lejano. Como recuerdo del otro, cada amigo se quedó con un qubit de los que componen el par EPR: Alice con el primero y Bob con el segundo.

Después de un tiempo, Alice necesita enviar un estado cuántico cualquiera a Bob (por el motivo que sea, no nos metemos en sus asuntos) de la forma $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$. Alice no conoce el estado exacto que ha de enviar (no sabe las amplitudes a y b), y sólo puede enviarle a Bob información clásica, como cadenas de ceros y unos. ¿Qué puede hacer Alice?

Cuando se repartieron el estado entrelazado, Alice y Bob quedaron conectados para siempre: es el momento de aprovecharlo. Para triunfar en la comunicación, Alice diseña un circuito con el cual Bob podrá decodificar el mensaje a partir de los bits que reciba:

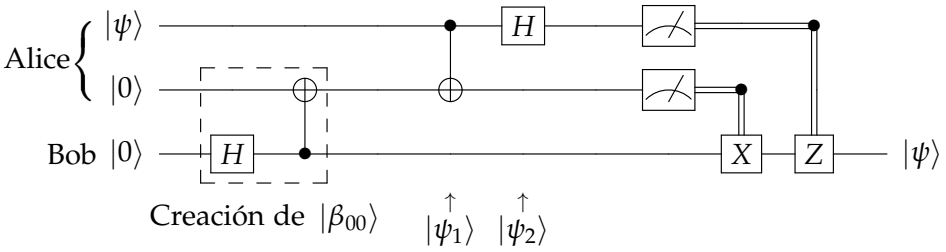


Figura 6: Circuito para teleportar el estado $|\psi\rangle$.

Veamos cómo este circuito puede ayudarles a teleportar el estado $|\psi\rangle$ como muestra la figura 6. El estado inicial del sistema completo es

$$|\psi_0\rangle = |\psi\rangle\,|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\left[a\,|0\rangle\,(|00\rangle + |11\rangle) + b\,|1\rangle\,(|00\rangle + |11\rangle)\right].$$

Alice hace pasar sus qubits por una puerta cNOT y una Hadamard, quedando el estado del sistema como sigue:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\left[a\,|0\rangle\,(|10\rangle + |01\rangle) + b\,|1\rangle\,(|00\rangle + |11\rangle)\right]$$

$$\begin{aligned}|\psi_2\rangle &= \frac{1}{2}\left[a(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + b(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)\right] \\ &= \frac{1}{2}\left[|00\rangle\,(a\,|0\rangle + b\,|1\rangle) + |01\rangle\,(a\,|1\rangle + b\,|0\rangle) + |10\rangle\,(a\,|0\rangle - b\,|1\rangle) + |11\rangle\,(a\,|1\rangle - b\,|0\rangle)\right].\end{aligned}$$

En esta última expresión es donde se ve la *magia* del entrelazamiento: al medir los qubits de Alice (obteniéndose A_1, A_2), sabemos en qué estado se encuentra el de Bob y qué operación deberíamos usar para transformarlo en $|\psi\rangle$. En la tabla 1, se estudian los distintos casos que podemos encontrar.

Es inmediato darse cuenta de que las acciones que se explicitan en la tabla se resumen con un un par de puertas controladas que tienen $|\psi_B\rangle$ como target: una cX controlada por el segundo qubit de Alice y una cZ controlada por el primero. Queda así comprobado que el circuito creado por Alice cumple su cometido y puede enviar el estado cuántico que Bob tanto necesitaba.

(A_1, A_2)	$\|\psi_B\rangle$	Operación a realizar sobre $\|\psi_B\rangle$
00	$a\|0\rangle + b\|1\rangle$	Dejarlo igual.
01	$a\|1\rangle + b\|0\rangle$	Aplicar una puerta X.
10	$a\|0\rangle - b\|1\rangle$	Aplicar una puerta Z.
11	$a\|1\rangle - b\|0\rangle$	Aplicar ambas puertas X y Z.

Tabla 1: Transformaciones sobre el qubit de Bob según las medidas de Alice.

Un lector exigente podría ahora quejarse de que en el circuito de la figura 6 aparece una puerta cZ, que no forma parte del conjunto de puertas de Clifford. Sin embargo, de las igualdades de (3.10), (3.11) y (3.12) que aplicar una puerta cZ es equivalente a aplicar la secuencia $H_3 \to \text{cNOT} \to H_3$, por lo que la promesa de que este circuito fuera estabilizador no se ha roto.

3. Códigos de corrección de errores: código de inversión del qubit

El circuito que nos disponemos a emplear a continuación no es estrictamente un código estabilizador, pero sí da gran protagonismo a las puertas cNOT, además de ser un ejemplo muy ilustrativo para entender cómo el formalismo estabilizador puede aplicarse en códigos de corrección de errores.

Contextualicemos primero con un planteamiento más general, que puede encontrarse en [16]. Supongamos que queremos transmitir un mensaje a través de un canal con ruido. Este canal altera el estado del sistema que queremos enviar de diversas formas, aplicando una puerta Z sobre algún qubit o cambiando un 0 por un 1 (aplicándose una puerta X sobre ese qubit), entre otras. En definitiva, el conjunto de errores que pueden aparecer forma parte del grupo de Pauli, $\mathcal{E} = \{E_a\} \subset \mathcal{P}_n$.

Recordemos que un código estabilizador asociado a un grupo $\mathcal{S}$ venía dado por $\mathcal{V} = \{|\psi\rangle : \sigma|\psi\rangle = |\psi\rangle, \forall \sigma \in \mathcal{S}\}$, esto es, se trata del subespacio propio asociado al valor propio $+1$ de todos los elementos de $\mathcal{S}$. Considerando $\mathcal{S}$ generado por un sistema $\{\sigma_i\}$, la detección de errores se realiza fijándonos en las relaciones de conmutación de los generadores con cada error:

- Si E_a anticonmuta con un cierto generador $\sigma_i \in \mathcal{S}$, se tiene que

$$\sigma_i E_a |\psi\rangle = -E_a \sigma_i |\psi\rangle = -E_a |\psi\rangle ,$$

  por lo que $E_a|\psi\rangle$ es un vector propio de σ_i con valor propio -1. Se entiende, por tanto, que $E_a|\psi\rangle$ es ortogonal al código $\mathcal{V}$. Así, midiendo cada generador σ_i, se puede definir un coeficiente $s_{i,a} \in \{0,1\}$, según si σ_i y E_a conmutan o anticonmutan:

$$\sigma_i E_a = (-1)^{s_{i,a}} E_a \sigma_i$$

  Se obtiene así el vector $\overline{s_a} = (s_{1,a}, ..., s_{n-k,a})$, que representa el síndrome del error E_a (k es el número de qubits codificados y n el número de qubits empleados en la codificación).

- Si $E_a \in \mathcal{S}$, el error no corrompe el sistema y deja el estado dentro del código $\mathcal{V}$.

- Si $E_a \notin \mathcal{S}$, pero conmuta con todos sus elementos, es decir, $E_a \in Z(\mathcal{S}) - \mathcal{S}$. Este caso es el más delicado, pues E modifica los elementos del código $\mathcal{V}$ pero no los saca de él. Así, E será un error indetectable para este código.

Con todo ello, podemos concluir que un código $\mathcal{V}$ puede corregir un conjunto de errores $\mathcal{E}$ si y solo si $E_a^\dagger E_b \in \mathcal{S} \bigcup (\mathcal{P}_n - Z(\mathcal{S}))$, para todo $E_a, E_b \in \mathcal{E}$.

Aprovechando las nociones recién explicadas, proponemos un circuito [17] capaz de corregir un único error del tipo aplicar una puerta X_i sobre un qubit i:

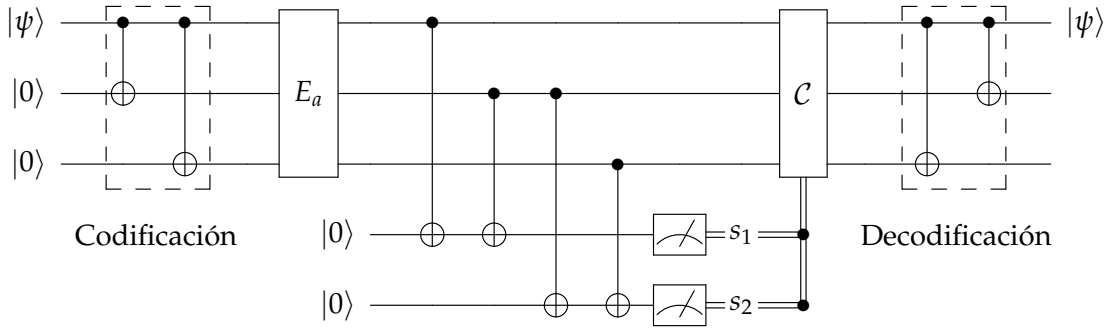


Figura 7: Circuito de corrección del error "invertir un qubit".

Queremos enviar un estado $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ a través de un canal con ruido, con la garantía de que dicho estado llegue inalterado al destino. Lo primero que hacemos para conseguirlo es codificar un qubit "lógico" con la ayuda de 3 qubits físicos. La codificación que se lleva a cabo es la siguiente:

$$|0\rangle_L = |000\rangle \ , \ |1\rangle_L = |111\rangle \qquad \Longrightarrow \qquad \alpha\,|0\rangle + \beta\,|1\rangle \longrightarrow \alpha\,|000\rangle + \beta\,|111\rangle = |\psi\rangle_L \, .$$

$\{|0\rangle_L, |1\rangle_L\}$ es la base del código estabilizador $\mathcal{V}$ que vamos a usar. Es inmediato ver que el código está asociado al grupo $\mathcal{S} = \langle Z_1Z_2, Z_2Z_3\rangle$ (y tenemos la suerte de que el único error que conmuta con los generadores del grupo es la identidad).

Al atravesar el error (que consideramos localizado entre la codificación y la corrección), puede ocurrir que el estado se quede dentro del código $\mathcal{V}$ o que sea transformado en un vector $E_a\,|\psi\rangle$ perteneciente a un subespacio $\mathcal{V}^*$(de dimensión 2) ortogonal a él. Midiendo los generadores de $\mathcal{S}$, Z_1Z_2 y Z_2Z_3, obtendremos dos valores s_1 y s_2 que constituyen el síndrome que nos revelará cuál ha sido el error. En la tabla 2 se valoran las distintas posibilidades.

Error E_a	Subespacio $\mathcal{V}^*$	s_1	s_2
$I_1I_2I_3$	$\text{Lin}(\|000\rangle, \|111\rangle) = \mathcal{V}$	0	0
$X_1I_2I_3$	$\text{Lin}(\|100\rangle, \|011\rangle)$	1	0
$I_1X_2I_3$	$\text{Lin}(\|010\rangle, \|101\rangle)$	1	1
$I_1I_2X_3$	$\text{Lin}(\|001\rangle, \|110\rangle)$	0	1

Tabla 2: Efecto de los posibles errores sobre el código.

Conocido el síndrome del error E_a, solo falta aplicar una puerta unitaria que vuelva a aplicar el error E_a según proceda, atendiendo al síndrome (s_1, s_2) que recibe por medio de los controles. Ahora ya tendríamos el estado $|\psi\rangle_L$ como antes de sufrir la transformación E_a, lo que nos permite recuperar el estado $|\psi\rangle$ que queríamos enviar gracias a una simple decodificación (inversión de las operaciones que aplicamos durante la codificación).

De forma muy similar, podemos construir un circuito capaz de corregir un error del tipo "cambiar de fase un estado", que se manifiesta como el paso de un qubit a través de una puerta Z. En este caso, tendríamos que cambiar la base a medir a $\{X_1X_2, X_2X_3\}$, cosa que en el circuito se traduciría en poner una puerta Hadamard para cada qubit antes y después del error E_a, ya que esto reduce el nuevo problema al recién resuelto.

3.4.3 La matriz estabilizadora

Terminamos la introducción al formalismo estabilizador con una nueva forma de representar el estado en que se halle el sistema. Para ello, hemos de pasar por una reformulación del grupo de Pauli, basada en un recurso al que llamaremos *etiquetas* [12].

Recordemos que un sistema de n qubits estaba asociado al espacio de Hilbert $\left(\mathbb{C}^2\right)^{\otimes n}$. Este espacio está, a su vez, asociado al grupo $G = \mathbb{Z}_2^n$ (del que obtenemos los estados básicos), de dimensión $\mathfrak{g} = 2^n$, por medio de la relación:

$$|g\rangle = |g(1)\rangle \otimes ... \otimes |g(n)\rangle \qquad (g \in G), \tag{3.21}$$

donde $g = (g(1), ..., g(n))$ es un elemento del grupo. Teniendo esto en mente, podemos escribir las matrices de Pauli X y Z como los siguientes operadores que actúan sobre elementos de $\left(\mathbb{C}^2\right)^{\otimes n}$:

$$X(g) := \sum_{h\in G} |h+g\rangle\langle h| \qquad , \qquad Z(g) := \sum_{h\in G} \chi_g(h)\,|h\rangle\langle h| \tag{3.22}$$

donde $g \in G$ y $\chi_g : G \longrightarrow \{-1, +1\}$ es un homomorfismo definido por

$$\chi_g(h) = \exp\left(2\pi i \sum_{i=1}^{n} \frac{g(i)h(i)}{2}\right). \tag{3.23}$$

Con esta notación, las componentes $g(i) = 1$ se traducirán en que la puerta X o Z actúa sobre el qubit i, mientras que $g(j) = 0$ quiere decir que no se aplican, esto es, la matriz identidad se encuentra en su lugar. Tomemos, por ejemplo, el elemento $g = (0, 1) \in \mathbb{Z}_2^2$ para ver esta nomeclatura en acción:

$$\begin{aligned} X(0,1) &= |00+01\rangle\langle 00| + |01+01\rangle\langle 01| + |10+01\rangle\langle 10| + |11+01\rangle\langle 11| \\ &= |01\rangle\langle 00| + |00\rangle\langle 01| + |11\rangle\langle 10| + |10\rangle\langle 11| \\ &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = I_1 \otimes X_2 \end{aligned}$$

$$\begin{aligned} Z(0,1) &= e^{i\pi\cdot 0}\,|00\rangle\langle 00| + e^{i\pi\cdot 1}\,|01\rangle\langle 01| + e^{i\pi\cdot 0}\,|10\rangle\langle 10| + e^{i\pi\cdot 1}\,|11\rangle\langle 11| \\ &= +\,|00\rangle\langle 00| - |01\rangle\langle 01| + |10\rangle\langle 10| - |11\rangle\langle 11| \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = I_1 \otimes Z_2 \end{aligned}$$

De esto se deduce que cualquier operador de Pauli podrá reescribirse de la siguiente forma

$$T_a = i^{-x_a \cdot z_a} X(x_a) Z(z_a) \ , \qquad x_a, z_a \in \mathbb{Z}_2^n \tag{3.24}$$

o lo que es lo mismo:

$$T_a = i^{-x_a \cdot z_a} \left(X_1^{x_{a1}} \otimes ... \otimes X_n^{x_{an}} \right) \left(Z_1^{z_{a1}} \otimes ... \otimes Z_n^{z_{an}} \right) \ , \qquad x_a, z_a \in \mathbb{Z}_2^n \tag{3.25}$$

Notaremos $(x_a, z_a) \in \mathbb{Z}_2^{2n}$ al vector de etiquetas que determina de forma única al correspondiente operador de Pauli, T_a.

Una vez hemos asimilado esta escritura de los operadores de Pauli por medio de etiquetas, podemos presentar el ente matemático con el que vamos a representar el sistema de n qubits y que haremos evolucionar a conveniencia. Ante la falta de convenio en la bibliografía para apodarlo (*check matrix* en [1] o *tableau* en [2]), el nombre **matriz estabilizadora** será suficientemente representativo.

La idea es utilizar esta matriz para representar los distintos generadores del grupo $S(|\psi\rangle)$. Para el caso en que tengamos n qubits, ya sabemos que es suficiente con n generadores para que el grupo quede completamente representado. La matriz será, pues, de dimensión $n \times 2n$: una fila por cada generador, que viene representado por su correspondiente etiqueta. Por ejemplo, en el caso de un sistema de $n = 3$ qubits, la matriz estabilizadora que reúne los generadores T_1, T_2 y T_3 vendrá dada por

$$\begin{array}{ccc} T_1 & \rightarrow & \left(\begin{array}{ccc|ccc} x_{11} & x_{12} & x_{13} & z_{11} & z_{12} & z_{13} \\ x_{21} & x_{22} & x_{23} & z_{21} & z_{22} & z_{23} \\ x_{31} & x_{32} & x_{33} & z_{31} & z_{32} & z_{33} \end{array} \right) \\ T_2 & \rightarrow & \\ T_3 & \rightarrow & \end{array} \tag{3.26}$$

4 Algoritmo de Gottesman-Knill

En esta sección, se propone un método para poder ir averiguando el estado en que se encuentra un sistema a lo largo de su evolución, pasando por puertas cuánticas y sufriendo medidas. A pesar de funcionar correctamente, en la siguiente sección veremos que este algoritmo puede ser mejorado, atendiendo a relaciones de conmutatividad entre operadores.

4.1 Puertas unitarias en el formalismo estabilizador

En el apartado anterior se desveló una nueva forma de representar los sistemas por medio de su grupo estabilizador. El problema que se plantea con ello es la necesidad de adaptar todo lo que habíamos aprendido hasta ahora, ya que los procesos de medida y aplicación de puertas cuánticas se habían diseñado para trabajar según las amplitudes que presentara cada uno de los vectores de la base.

Supongamos que se aplica una puerta unitaria U sobre el estado $|\psi\rangle$ que, recordemos, estará descrito por el grupo de matrices $S(|\psi\rangle) \equiv \mathcal{S}$. Entonces, para cualquier elemento $T \in \mathcal{S}$, se tiene

$$U|\psi\rangle = UT|\psi\rangle = UTU^\dagger U|\psi\rangle \ , \tag{4.1}$$

de manera que el vector $U|\psi\rangle$ es estabilizado por la matriz $UTU^\dagger$, de donde se deduce que el grupo $U\mathcal{S}U^\dagger = \{UTU^\dagger / T \in \mathcal{S}\}$ estabiliza el estado $U|\psi\rangle$. Es más, si $\mathcal{S} = \langle T_1, T_2, ..., T_l \rangle$, se tiene que $UT_1U^\dagger, ..., UT_lU^\dagger$ generan $U\mathcal{S}U^\dagger$. Así, para conocer cómo afectan las puertas sobre el estabilizador, será suficiente averiguar cómo modifican a los generadores del estabilizador.

Necesitamos que la acción por conjugación de una puerta, sobre los generadores del grupo de Pauli, los mantenga dentro del mismo, esto es, $\mathrm{ac}(U, P) = UPU^\dagger \in \mathcal{P}_n$, $\forall P \in \mathcal{P}_n$, $\forall U \in \mathcal{U}_n$. Por ello, U está en el normalizador $N_{\mathcal{U}_n}(\mathcal{P}_n)$, al cual llamaremos **grupo de Clifford**. Se puede demostrar [18], en el caso $n = 1$, que dicho grupo está generado por puertas de Hadamard, fase y cNOT, extendiéndose con productos tensoriales para $\mathcal{P}_n$. Podemos ahora comprender que, en el Teorema 6, las puertas mencionadas en (i) y (ii) no fueron escritas de forma arbitraria.

Veamos algunos ejemplos de las puertas unitarias que actúan sobre un sólo qubit. Por ejemplo, la puerta de Hadamard afecta a cada una de las matrices de Pauli como se muestra a continuación:

$$HXH^\dagger = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix} = Z \tag{4.2}$$

$$HYH^\dagger = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 0 & 2i \\ -2i & 0 \end{pmatrix} = -Y \tag{4.3}$$

$$HZH^\dagger = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix} = X \tag{4.4}$$

Para ver el efecto de las puertas de fase, el procedimiento sería muy similar; y para las puertas de Pauli, podemos utilizar las igualdades 3.16 simplificando los cálculos enormemente. Resumimos todos los resultados en la tabla 3.

Estos resultados son directamente aplicables al caso en que nuestro sistema esté formado por n qubits. Por ejemplo, el estado $|0\rangle^{\otimes n}$ tiene al grupo $\langle Z_1, ..., Z_n\rangle$ por estabilizador y si hacemos pasar cada uno de los qubits por una puerta Hadamard, el grupo estabilizador del sistema pasaría a ser $\langle X_1, ..., X_n\rangle$.

Avanzando hacia al caso en que la puerta unitaria U relaciona más de un qubit (digamos k), hemos de entender que la operación a realizar es $UAU^\dagger = B$, donde tanto A como B son matrices de dimensión 2^k resultado del producto tensorial de k matrices de Pauli. El ejemplo más obvio es el de la puerta *controlled*-NOT (que representaremos simplemente por U):

$$\begin{aligned} UX_1U^\dagger &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = X_1 \otimes X_2 \end{aligned} \tag{4.5}$$

Podríamos proceder de igual manera con cualquier operador de Pauli que quisiéramos utilizar. Esto es aplicable a cualquier matriz unitaria U que queramos aplicar sobre varios qubits, pero el Teorema 6 asegura que sólo vamos a necesitar las puertas cNOT de este tipo. De nuevo, el resto de resultados a considerar se encuentran recogidos en las tablas que presentamos a continuación:

Puerta	Entrada	Salida
X	X	X
	Y	$-Y$
	Z	$-Z$
Y	X	$-X$
	Y	Y
	Z	$-Z$
Z	X	$-X$
	Y	$-Y$
	Z	Z
H	X	Z
	Y	$-Y$
	Z	Z

Puerta	Entrada	Salida
P	X	Y
	Y	$-X$
	Z	Z
cNOT	X_1	X_1X_2
	X_2	X_2
	X_1X_2	X_1
	Y_1	Y_1X_2
	Y_2	Z_1Y_2
	Y_1Y_2	$-X_1Z_2$
	Z_1	Z_1
	Z_2	Z_1Z_2
	Z_1Z_2	Z_2

Tabla 3: Acción de puertas unitarias sobre las matrices de Pauli

Obsérvese que en la tabla hay información redundante: debido a las relaciones entre las matrices de Pauli (3.16), conociendo lo que ocurre con dos de ellas se puede calcular lo que ocurre con la tercera. En nuestro algoritmo, se empleará sólamente la información referente a las puertas X y Z sin que esto suponga pérdida alguna de información.

4.2 La medida en el formalismo estabilizador

Asociado a todo operador de Pauli $T \in \mathcal{P}_n$, puede realizarse la medida del mismo sobre su base de vectores propios. Consideramos para ello la descomposición espectral $T = \sum \lambda P_\lambda$. Dado un estado cuántico $|\psi\rangle$, la medida que obtiene el valor propio λ ocurre con probabilidad $\|P_\lambda |\psi\rangle\|^2$, transformando $|\psi\rangle$ en el estado $P_\lambda |\psi\rangle$.

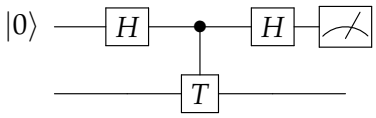


Figura 8: Medida de T

En la figura 8, se muestra el circuito empleado para medir (sobre un qubit auxiliar) un qubit en la base de autovectores de un operador T cualquiera.

Pretendemos, por tanto, medir un operador $T \in \mathcal{P}_n$[11], cuyos valores propios sólo podrán ser ± 1. El sistema está en un estado $|\psi\rangle$ estabilizado por $\mathcal{S} = S(|\psi\rangle) = \langle T_1, ...T_l\rangle$, pueden entonces ocurrir dos cosas:

a. T conmuta con todos los generadores del estabilizador:

Como $T_j T |\psi\rangle = T T_j |\psi\rangle = T |\psi\rangle \ \forall j = 1, ..., n$, $T |\psi\rangle$ está también estabilizado por $S(|\psi\rangle)$ y es, por tanto, un múltiplo de $|\psi\rangle$. Como $T^2 = I$, forzosamente $T |\psi\rangle = \pm |\psi\rangle$, de modo que T o $-T$ es un elemento del estabilizador.

Si $+T \in S(|\psi\rangle)$, se tiene que $T |\psi\rangle = |\psi\rangle$ y de ahí que al medir T se vaya a obtener el valor $+1$ con probabilidad 1. Además, como $|\psi\rangle$ ya se encuentra en el subespacio que proyecta T, la medida no alterará el estado del sistema, dejando el grupo estabilizador invariante. Lo mismo ocurriría si fuera $-T$ el que está en el estabilizador, obteniéndose en este caso el valor -1 al medir.

En definitiva, si $\mathcal{C}_\mathcal{S}(T) = \mathcal{S}$, su medida será un procedimiento determinista que no altera al grupo estabilizador: $\mathcal{S}' = \mathcal{S}$.

b. T anticonmuta con uno o más generadores del estabilizador y conmuta con el resto:

Antes de nada, veamos que todo se reduce al caso en que T anticonmuta con sólo uno de los generadores del estabilizador: si T anticonmuta con dos generadores T_1 y T_2, por ejemplo, resulta que T conmuta con su producto $T_1 T_2$. Así, podemos sustituir uno de los dos generadores por el producto (que también es un elemento del grupo estabilizador), dejando sólo un generador que anticonmute con T. Podríamos repetir este paso las veces que hiciera falta según cuántos generadores anticonmutasen con T.

T tiene dos valores posibles que pueden resultar de su medida, ± 1, cuyos proyectores son $\frac{I \pm T}{2} |\psi\rangle\langle\psi|$, respectivamente. Por tanto, las probabilidades (según el Postulado 4') de medir cada uno de estos valores propios viene dada por:

$$p_{+1} = \mathrm{Tr}\left(\frac{I+T}{2} |\psi\rangle\langle\psi|\right) \qquad p_{-1} = \mathrm{Tr}\left(\frac{I-T}{2} |\psi\rangle\langle\psi|\right) \tag{4.6}$$

Y usando que $T_1 |\psi\rangle = |\psi\rangle$ y $T T_1 = -T_1 T$ llegamos a que

$$p_{+1} = \mathrm{Tr}\left(\frac{I+T}{2} T_1 |\psi\rangle\langle\psi|\right) = \mathrm{Tr}\left(T_1 \frac{I-T}{2} |\psi\rangle\langle\psi|\right) \tag{4.7}$$

[11] Suponemos, sin pérdida de generalidad, que no va acompañado de factor multiplicativo $u \in \{-1, \pm i\}$.

Por último, empleamos que $\mathrm{Tr}(ABC) = \mathrm{Tr}(BCA)$ para llevar T_1 a la derecha de la expresión y absorberlo con $\langle\psi|$ (puesto que $T_1 = T_1^\dagger$). Con ello, deducimos que

$$p_{+1} = \mathrm{Tr}\left(\frac{I-T}{2}|\psi\rangle\langle\psi|\right) = p_{-1} \qquad \Rightarrow \qquad p_{+1} = p_{-1} = \frac{1}{2} \tag{4.8}$$

Concluimos, por tanto, que en este caso el resultado de la medida será aleatorio y que, una vez medido T obteniéndose ± 1, el estado colapsará convirtiéndose en $|\psi^{\pm}\rangle = \frac{I\pm T}{\sqrt{2}}|\psi\rangle$. Su grupo estabilizador pasará a ser $\mathcal{S}' = \langle \pm T, T_2, ..., T_n\rangle = \langle \pm T, \mathcal{C}_{\mathcal{S}}(T)\rangle$.

4.3 Implementación y eficiencia del algoritmo

Antes de nada, es imprescindible introducir algunas nociones sobre complejidad y eficiencia que se emplearán a continuación.

Para determinar si un algoritmo es o no eficiente, necesitamos cuantificar los recursos que consume este, es decir, el número de operaciones que se han de ejecutar para poder llevarlo a cabo. Esto se hace comparando dicho número con el comportamiento asintótico de una cierta función, que suelen ser polinomios, logaritmos y exponenciales.

Definición 17. *Sean $f(n)$ y $g(n)$ dos funciones sobre $\mathbb{N}$. Decimos que "$f(n)$ se encuentra en la clase de funciones $O(g(n))$", o simplemente "$f(n)$ es $O(g(n))$", si existen $c \in \mathbb{R}^+$ y $n_0 \in \mathbb{N}$ tales que $f(n) \leq cg(n)$, $\forall n > n_0$.*

Como es de esperar, la complejidad de un algoritmo en términos de operaciones se traduce en la cantidad de tiempo requerido para ejecutarlo. Diremos que un algoritmo se ejecuta en tiempo $poly(n)$ si, en la definición anterior, $g(n)$ es un polinomio de cualquier grado en n, mientras que tiempo $polylog(n)$ hará a referencia a polinomios en $(\log n)$.[12]

Definición 18. *Un algoritmo clásico de simulación es **eficiente** si se puede llevar a cabo en tiempo $poly(n)$, donde n es el número del qubits del circuito cuántico que se está representando.*

Los mejores algoritmos son, entonces, aquellos que presentan un número de operaciones lineal con el número de qubits e incluso inferior (no será el caso con los nuestros). Con respecto a los algoritmos ineficientes, destacaremos que los más frecuentes pertenecen a la clase $O(e^n)$. Tras esta introducción, ya estamos en condiciones de ver lo que ocurre con este algoritmo.

Comenzando con las puertas cuánticas, poco queda por decir acerca de su manipulación. En 4.1 ya veíamos que sólo interesa la aplicación de puertas de Clifford, y que al aplicar una puerta U necesitamos **actualizar todos los generadores** del grupo estabilizador según lo reflejado en la tabla 3.

Por tanto, a cada puerta que actúe sobre el sistema, el algoritmo contempla modificar las n etiquetas que representan a los distintos generadores. El siguiente resultado, demostrado en [19], aclara el tiempo que se consume durante este proceso:

Teorema 8. *Sea U una puerta normalizadora y σ un operador de Pauli expresado en términos de su etiqueta. Se tiene entonces que la etiqueta correspondiente a $U\sigma U^\dagger$ puede ser calculada en tiempo $polylog(\mathfrak{g})$.*

[12] Esta notación trata con logaritmos en cualquier base, ya que la diferencia no es más que una constante multiplicativa.

El teorema se formula para puertas normalizadoras[13], que son una generalización de las puertas de Clifford, de manera que se puede aplicar en este contexto. Recordemos que $\mathfrak{g} = 2^n$ era la dimensión del grupo $\mathbb{Z}_2^n$ y el espacio de Hilbert asociado $(\mathbb{C})^{\otimes n}$. Por tanto, se deduce que la acción de una puerta sobre un generador se consigue llevar a cabo en un tiempo lineal con el número de qubits. Para completar el proceso con todos los generadores, se requiere un tiempo $O(n^2)$.

Pasemos al proceso de medida. Dado que los vectores propios de la matriz Z son los estados básicos $|0\rangle$ y $|1\rangle$, asociados a los valores propios $+1$ y -1 respectivamente, la medida de un qubit a equivale a la medida del operador Z_a (puerta Z actuando sobre dicho qubit). La etiqueta correspondiente a dicho operador es el vector $e_{a+n} \in \mathbb{Z}_2^{2n}$, que tiene todas sus coordenadas nulas salvo la $(a+n)$-ésima, que está ocupada por un 1.

De forma análoga a lo expuesto en la figura 8, se tiene que el circuito empleado para medir la puerta Z es el siguiente:

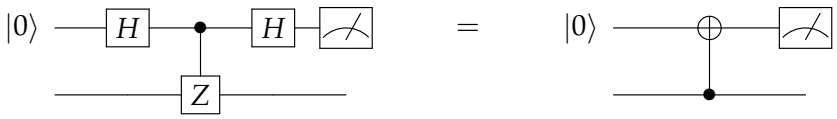


Figura 9: Medida de Z

En el apartado 4.2, vimos que el problema de la medida se reduce a averiguar si el operador Z_a pertenece o no al grupo estabilizador del estado, $\langle T_1, ..., T_n \rangle$. La forma que Gottesman y Knill plantean es encontrando la solución $(c_1, ..., c_n) \in \mathbb{Z}_2^n$ de la ecuación

$$\pm Z_a = T_1^{c_1} \cdot ... \cdot T_n^{c_n} \tag{4.9}$$

La ecuación (4.9) se puede traducir en un sistema de ecuaciones cuya matriz de coeficientes es la matriz estabilizadora del estado:

$$(0 \dots 0 \overset{(a+n)}{1} 0 \dots 0) = \begin{pmatrix} c_1 & \dots & c_n \end{pmatrix} \left(\begin{array}{ccc|ccc} x_{11} & \dots & x_{1n} & z_{11} & \dots & z_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nn} & z_{n1} & \dots & z_{nn} \end{array}\right) \tag{4.10}$$

Si (4.10) tiene solución, nos encontramos en el caso en que $Z_a \in \mathcal{S}$ y no hay que actualizar el sistema. Sin embargo, para averiguar cuál es el signo que acompaña a Z_a (signo del autovalor medido), se ha de resolver el sistema, y esto implica emplear una reducción gaussiana que es un proceso en $O(n^3)$.

En caso contrario, la medida del operador es un proceso aleatorio, de manera que se decide si el autovalor es $+1$ o -1 "lanzando una moneda". Además, tenemos que actualizar el grupo estabilizador: reemplazamos un generador T_j que anticonmute con Z_a por él, y multiplicamos por T_j el resto de generadores que anticonmuten con Z_a. Este es un proceso que emplea $O(n^2)$ operaciones [2].

Si sumamos todos los procesos implicados, concluimos que el algoritmo está en la clase $O(n^3)$ y es, por ende, eficiente. Sin embargo, aún podemos mejorarlo.

[13]Las puertas normalizadoras se definen a partir de ciertos recursos que escapan el contenido de este trabajo. Por ello, sólo diremos que suponen una generalización de las puertas de Clifford.

5 Algoritmo de Aaronson-Gottesman

En esta sección, se explica el algoritmo desarrollado por Scott Aaronson y Daniel Gottesman [2], que introduce mejoras de eficiencia respecto al que tratábamos anteriormente. La clave para conseguirlo se encuentra en explotar las relaciones de conmutatividad (y anticonmutatividad) que presentan las matrices de Pauli (3.16).

En este caso, necesitamos cambiar la representación del sistema, ampliando la matriz estabilizadora. Consideraremos también otros n generadores del "destabilizador", esto es, operadores que no forman parte del estabilizador pero que, junto con él, generan el grupo de Pauli completo. Además, ahora la matriz también incluye una columna extra para tener en cuenta la fase de cada operador. Para no confundir la notación con todo lo visto anteriormente, notaremos $R_i = \pm P_1 ... P_n$[14] $(i = 1, ..., 2n)$ a cada fila (generador) que aparece en esta nueva versión de la matriz.

El algoritmo representa un estado por la matriz de variables binarias x_{ij}, z_{ij}, r_i para cada $i, j \in \{1, ..., 2n\}$:

$$\begin{array}{ccc|ccc|c} R_1 & \to & x_{11} & \cdots & x_{1n} & z_{11} & \cdots & z_{1n} & r_1 \\ \vdots & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ R_n & \to & x_{n1} & \cdots & x_{nn} & z_{n1} & \cdots & z_{nn} & r_n \\ \hline R_{n+1} & \to & x_{(n+1)1} & \cdots & x_{(n+1)n} & z_{(n+1)1} & \cdots & z_{(n+1)n} & r_{n+1} \\ \vdots & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ R_{2n} & \to & x_{(2n)1} & \cdots & x_{(2n)n} & z_{(2n)1} & \cdots & z_{(2n)n} & r_{2n} \end{array} \tag{5.1}$$

Los elementos destabilizadores están representados por las filas $\{1, ..., n\}$ mientras que los generadores del grupo estabilizador se corresponderán con las filas $\{n+1, ..., 2n\}$. Las $2n$ primeras entradas en cada fila serán la etiqueta del operador de Pauli correspondiente, mientras que r_i denota la fase que lo acompaña: $r_i = 0$ para fases positivas y $r_i = 1$ para fases negativas.

Ahora que ya conocemos la nueva matriz estabilizadora, solo nos queda conocer qué hemos de hacer para actualizarla ante de la acción de las posibles puertas de Clifford y de medida que podamos aplicar sobre el sistema.

Primero, necesitamos ver cómo implementaremos la composición de operadores (o producto de matrices de dimensión 2^n) que, recordemos, otorgaba a los operadores de Pauli el atributo de grupo. La denotaremos por el símbolo $+$, pues la composición de dos operadores tiene una etiqueta que recuerda a la suma de etiquetas de los sumandos. Se define, pues, una subrutina encargada de computar la recién mencionada operación:

rowsum(h,i) Esta subrutina toma las filas h e i de la matriz estabilizadora (generadores R_h y R_i) y reescribe sobre la fila h, transformándola en el generador $R_i + R_h$.

Lo primero que hace es ver qué ocurre con la fase del nuevo generador. Para ello, definimos la función $g : \mathbb{Z}_2^4 \longrightarrow \{-1, 0, +1\}$, que acepta como parámetros las etiquetas (x_1, z_1, x_2, z_2) correspondiente a dos matrices de Pauli y devuelve un coeficiente r tal que i^r es la fase resultante cuando dichas matrices han sido multiplicadas.

[14] Sólo consideramos ± 1 para la fase del operador R_i y nos olvidamos de $\pm i$. De lo contrario, se daría el caso en que $R_i^2 = -I$, cosa que sabemos imposible para códigos no triviales.

Es sencillo comprobar que la forma explícita de esta función será:

$$g(x_1, z_1, x_2, z_2) = \begin{cases} 0 & \text{si } (x_1, z_1) = (0,0) \\ x_2(1-2z_2) & \text{si } (x_1, z_1) = (0,1) \\ z_2(2x_2-1) & \text{si } (x_1, z_1) = (1,0) \\ z_2 - x_2 & \text{si } (x_1, z_1) = (1,1) \end{cases} \tag{5.2}$$

Gracias a ella, seremos ahora capaces de establecer cuál es el nuevo valor para r_h. Nos fijamos en el valor de la siguiente expresión, en la que se reúnen las aportaciones de las fases de cada uno de los generadores y las que ofrecen los n productos de matrices de Pauli que se van a llevar a cabo:

$$d = 2r_h + 2r_i + \sum_{j=1}^{n} g(x_{ij}, z_{ij}, x_{hj}, z_{hj})$$

Entonces, asignaremos $r_h := 0$ si $d \equiv 0 \pmod 4$ y $r_h := 1$ cuando $d \equiv 2 \pmod 4$ (nunca se dará el caso en que d sea congruente a 1 o 3).

Ahora que ya tenemos la fase, podemos modificar las etiquetas que denotan al generador R_h. Esta etapa es más sencilla, simplemente tenemos que ejecutar $x_{hj} := x_{ij} \oplus x_{hj}$ $z_{hj} := z_{ij} \oplus z_{hj}$ para cada columna $j \in \{1, ..., n\}$, donde la operación $\oplus$ no es más que la suma módulo 2.

5.1 Representación de estados puros

Llegamos al tan esperado algoritmo mejorado para simular circuitos cuánticos. Comenzamos ilustrando el caso en que lo que hacemos evolucionar es un estado puro, para generalizarlo posteriormente al caso de estados mezcla.

El estado de partida será $|0\rangle^{\otimes n}$, de manera que $S\left(|0\rangle^{\otimes n}\right) = \langle Z_1, ..., Z_n \rangle$. Por tanto, añadiendo una fila auxiliar R_{2n+1} a la matriz (su utilidad se verá más tarde), su configuración inicial vendrá dada por $r_i = 0$ para todo $i \in \{1, .., 2n+1\}$ y $x_{ij} = \delta_{ij}$, $z_{ij} = \delta_{(i-n)j}$, $\forall i \in \{1, ..., 2n+1\}$ $\forall j \in \{1, ..., n\}$. Por ejemplo, si $n = 2$ la matriz estabilizadora en el instante inicial adoptará la forma

$$\left(\begin{array}{cc|cc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \end{array}\right) \tag{5.3}$$

Durante la simulación, modificaremos el estado del sistema por medio de puertas de Clifford, que actúan según lo recogido en la Tabla 3. ¿Cómo interpretará el algoritmo la acción de cada una de ellas?

- **cNOT sobre b controlado por a:** Para cada fila $i \in \{1, ..., 2n\}$, se tienen $r_i := r_i \oplus x_{ia} z_{ib} (x_{ib} \oplus z_{ia} \oplus 1)$ y $x_{ib} := x_{ib} \oplus x_{ia}$, $z_{ia} := z_{ia} \oplus z_{ib}$.
- **Hadamard sobre el qubit a:** Para todo $i \in \{1, ..., 2n\}$, asignamos $r_i := r_i \oplus x_{ia} z_{ia}$ e intercambiamos x_{ia} y z_{ia}.

- **Puerta de fase sobre el qubit** a**:** Cambiaremos $r_i := r_i \oplus x_{ia}z_{ia}$ y $z_{ia} := z_{ia} \oplus x_{ia}$ para todas las filas $i \in \{1, ..., 2n\}$.

Lo que tiene más enjundia de este algoritmo es el proceso de medida, cuya esencia son las consideraciones que expusimos en 4.2. Comentaremos primero lo que hace el algoritmo y, seguido de ello, daremos la justificación teórica:

Medida del qubit a **en la base canónica:** Comprobamos si hay algún índice $p \in \{n+1, ..., 2n\}$ tal que $x_{pa} = 1$. Se abre la puerta a dos posibilidades distintas:

Caso 1 : tal p **existe.** En caso de haber más de uno, tomaremos el mínimo de ellos.

La medida será aleatoria, así que el estado requiere ser actualizado. Para tal fin, llamamos a rowsum(i, p) para todas las filas $i \in \{1, ..., 2n\}$ tales que $i \neq p$ y $x_{ia} = 1$. Tras ello, guardamos la fila R_p en la R_{p-n} y llenamos R_p de ceros, excepto r_p (que tomará los valores 0 o 1 con igual probabilidad) y $z_{pa} = 1$. Por último, devolvemos el valor r_p como resultado de la medida.

Caso 2 : no existe tal p**.**

Ahora la salida será determinista, así que la medida no alterará el estado. Sólo tenemos que ver si se observa 0 o 1. Para ello, rellenamos de ceros la fila R_{n+1} completa y sobre ella, sumamos con rowsum$(2n+1, i+n)$ todas las filas R_{i+n} con $i \in \{1, ..., n\}$ tales que $x_{ia} = 1$. Finalizamos devolviendo r_{2n+1} como salida de la medida.

Todo lo anterior parece muy abstracto, pero vamos a ver cómo todo cobra sentido después de varias apreciaciones:

1. Empezamos por definir el producto simpléctico pertinente para trabajar en estas situaciones. Recordando la definición 11, se entiende que, considerando dos filas de la matriz como $R_i \equiv (x_i, z_i) \in \mathbb{Z}_2^{2n}$, el producto simpléctico de generadores se define como

$$(R_h|R_i) = z_h \cdot x_i - z_i \cdot x_h \pmod 2 = \bigoplus_{j=1}^{n} (x_{ij}z_{hj} \oplus x_{hj}z_{ij}) \tag{5.4}$$

De esta definición se deducen un par de resultados importantes:

Lema 1. *Sean A, B dos matrices de Pauli representadas, respectivamente, por las etiquetas $a \equiv (x_a, z_a)$, $b \equiv (x_b, z_b) \in \mathbb{Z}_2^2$. Se verifican entonces las siguientes afirmaciones:*

(i) A y B conmutan si, y solo si, $(a|b) = 0$.

(ii) A y B anticonmutan si, y solo si, $(a|b) = 1$.

Demostración. Nos hallamos en el caso de la definición 5.4 en que $n = 1$. Por tanto, el producto simpléctico que nos interesa vendrá dado por $(a|b) = x_a z_b \oplus x_b z_a$. Es inmediato, con una prueba por casos (considerando las 16 posibles combinaciones de ceros y unos para la 4-upla (x_a, z_a, x_b, z_b)) demostrar las afirmaciones del lema. □

Lema 2. *Sean R_h y R_i dos generadores de la matriz estabilizadora. Entonces R_h conmuta con R_i si $(R_h|R_i) = 0$, y anticonmuta con él si $(R_h|R_i) = 1$.*

Demostración. Sean dos operadores de Pauli $P = i^k P_1 ... P_n$ y $Q = i^l Q_1 ... Q_n$ (donde P_i y Q_i son matrices de Pauli). Es claro que $[P, Q] = 0$ si, y solo si, el número de índices tales que $\{P_j, Q_j\} = 0$ es par. De forma análoga, $\{P, Q\} = 0$ si, y solo si, dicho número de índices es impar. A partir de ahora, identificamos P con R_h y Q con R_i.

Supongamos que $(R_h|R_i) = 0$, esto es, $\bigoplus_{j=1}^{n}(x_{hj}z_{ij} \oplus x_{ij}z_{hj}) = 0$. Tenemos entonces que el número de sumandos $(x_{hj}z_{ij} \oplus x_{ij}z_{hj})$ que no se anulan es par (ya que reduciendo módulo 2 se anula). Por el lema 1, esto equivale a que el número de parejas (P_j, Q_j) que anticonmutan es par. Por lo mencionado al comienzo de la demostración, esto equivale, a su vez, a la primera afirmación del lema.

Supongamos ahora que $(R_h|R_i) = 1$ y, de forma análoga a lo anterior, deducimos que el número de sumandos $(x_{hj}z_{ij} \oplus x_{ij}z_{hj})$ que resultan 1 ha de ser impar (para que no se anule reduciendo módulo 2). Por tanto, el número de parejas (P_j, Q_j) que anticonmutan es impar, lo que equivale a que R_h y R_i anticonmuten. □

2. Recogemos las relaciones entre los distintos generadores de la matriz en la siguiente

Proposición 6. *Las siguientes afirmaciones sobre la matriz estabilizadora se cumplen siempre:*

(i) $R_{n+1}, ..., R_{2n}$ *generan* $S(|\psi\rangle)$, *y* $R_1, ..., R_{2n}$ *generan* $\mathcal{P}_n$.

(ii) *Los generadores* $R_1, ..., R_n$ *conmutan entre ellos.*

(iii) $\forall h \in \{1, ..., n\}$, R_h *anticonmuta con* R_{h+n}.

(iv) *Para todos* $i, h \in \{1, ..., n\}$ *tales que* $i \neq h$, R_i *conmuta con* R_{h+n}.

Es fácil comprobar que estas afirmaciones se cumplen para el estado inicial. Además, el algoritmo fuerza que todas estas afirmaciones sigan siendo verdad tras cada actualización del grupo estabilizador que hagamos.

3. Supongamos que la medida del qubit a da lugar a una salida determinista. Esto ocurría en caso en que Z_a pertenecía al estabilizador. Con la nueva notación (usando $+$ para el producto de operadores), la ecuación (4.9) se convierte en

$$\sum_{h=1}^{n} c_h R_{h+n} = \pm Z_a. \tag{5.5}$$

Nos proponemos averiguar el valor de los coeficientes, ya que sumando los R_{h+n} adecuados, podremos saber si la fase del output es positiva o negativa. Obsérvese que, para todo $i \in \{1, ..., n\}$,

$$c_i \overset{\substack{L2\\P6}}{=} \sum_{h=1}^{n} c_h (R_i|R_{h+n}) = \left(R_i \middle| \sum_{h=1}^{n} c_h R_{h+n} \right) = (R_i|Z_a) \tag{5.6}$$

Así, comprobando si R_i anticonmuta con Z_a (cosa que ocurrirá si y solo si $x_{ia} = 1$, puesto que esto denotará que R_i tiene una matriz X o Y en la posición a), podemos saber si $c_i = 1$ y, por tanto, si rowsum$(2n + 1, i + n)$ ha de ser llamada.

4. Si nos ponemos en el caso en que Z_a no está en el estabilizador, multiplicamos algunos operadores por uno que anticonmuta con él. El siguiente lema aclara cómo altera esta operación las relaciones de conmutatividad de los distintos operadores:

Lema 3. *Sean A, B y C tres operadores de Pauli sobre un mismo espacio de Hilbert. Entonces:*
(a) Si A y C conmutan, $[A,B]=0$ si y solo si $[A,BC]=0$.
(b) Si A y C anticonmutan, $[A,B]=0$ si y solo si $\{A,BC\}=0$.

Demostración. Comenzamos con el caso (a), suponiendo que $AC=CA$:

$$AB=BA \quad \Rightarrow \quad A(BC)=BAC=(BC)A \quad \Rightarrow \quad [A,BC]=0$$
$$A(BC)=(BC)A \Rightarrow A(BC)C=(BC)AC \Rightarrow AB=BCCA=BA \Rightarrow [A,B]=0$$

Vamos con (b), suponiendo que $AC=-CA$:

$$AB=BA \quad \Rightarrow \quad A(BC)=BAC=-(BC)A \quad \Rightarrow \quad \{A,BC\}=0$$
$$A(BC)=-(BC)A \Rightarrow A(BC)C=-(BC)AC \Rightarrow AB=-BC(-CA)=BA \Rightarrow [A,B]=0$$

Tratándose de matrices de Pauli, sabemos que si no conmutan, entonces anticonmutan. Así, pueden deducirse nuevas implicaciones sin más que negar alguna de las tesis. □

Ya disponemos de todos los ingredientes necesarios para comprender lo que hacemos al medir el qubit a, es decir, al medir el operador Z_a:

Caso 1. Al tomar un $p \in \{n+1,...,2n\}$ tal que $x_{pa}=1$, estamos seleccionando un generador del estabilizador, R_p, que anticonmuta con Z_a. Al "sumar" este generador con el resto de generadores R_i $(i \in \{1,...,2n\})$ que también anticonmutan con Z_a (los que cumplen $x_{ia}=1$), conseguimos que los correspondientes productos conmuten con Z_a. Llevamos, finalmente, el generador R_p, que anticonmuta con Z_a, al destabilizador (posición $p-n$) y registramos Z_a como el generador R_p del estabilizador.

Se tiene entonces que $R_p=Z_a$ conmuta con todas los generadores de la matriz salvo con R_{p-n}, cumpliéndose entonces (iii) y (iv) en la proposición 6. Por su parte, el lema 3 garantiza que se verifique (ii).

Caso 2. Al no tener ningún $x_{pa}=1$, Z_a conmuta con todos los generadores del estabilizador y, de ahí, $+Z_a$ o $-Z_a$ pertenece a él (aunque no tiene por qué ser necesariamente unos de los generadores). Sumamos aquellos generadores del estabilizador correspondientes a los generadores del destabilizador que anticonmutan con Z_a. Con ello, estamos computando la suma (5.5) de aquellos términos con $c_h=1$. De este modo, sabiendo el signo de la fase en el primer miembro de la igualdad, seremos conocedores del signo que aparece en el segundo miembro: un signo $+$ significará que la salida es el estado $|0\rangle$ y $-$ dará a entender que el output es $|1\rangle$.

En las explicaciones para justificar la medida en el algoritmo, descubrimos estaba diseñada para verificar las condiciones de la proposición 6. ¿Qué ocurre con ellas cuando aplicamos una puerta de Clifford U? Veamos que conserva conmutatividad:

$$AB=BA \Leftrightarrow UABU^\dagger=UBAU^\dagger \Leftrightarrow (UAU^\dagger)(UBU^\dagger)=(UBU^\dagger)(UAU^\dagger)$$

Cuando aplicamos una puerta sobre el sistema, esta se traduce en la acción por conjugación sobre todos los operadores. Por tanto, estos seguirán conservando las relaciones de conmutatividad previas a su aplicación.

5.2 Representación de estados mezcla

Si recordamos lo visto en 3.1.1, la matriz densidad supone una generalización de los estados puros. En este apartado, buscamos adaptar el algoritmo para un cierto tipo de estados mezcla: los **estados mezcla estabilizadores**, es decir, estados concebidos como una distribución uniforme sobre todos los estados que se encuentran en el código $\mathcal{V}$, un subespacio estabilizado por un grupo descrito a partir de $r < n$ generadores. Sabemos, por [12], que el código asociado tendrá dimensión $|\mathcal{V}| = \frac{\mathfrak{g}}{|\mathcal{S}|} = 2^{n-r}$.

Será necesario conocer la forma en que se escribe la matriz densidad de un estado mezcla en términos de su estabilizador. Si M es un operador de Pauli, $\frac{I+M}{2}$ resulta ser el proyector sobre el espacio propio de M asociado al autovalor $+1$. Por tanto, la matriz densidad que representa a un estado mezcla estabilizador (estando su estabilizador generado por lo operadores $M_1, ..., M_r$) vendrá dada por [15]

$$\rho = \frac{1}{2^r} \prod_{i=1}^{r} (I + M_i) \tag{5.7}$$

En definitiva, la matriz densidad del sistema resulta ser el proyector sobre su código estabilizador. Un ejemplo nos ayudará a comprenderlo mejor. Tomemos el código estabilizador que usamos en el ejemplo 3. Códigos de corrección de errores: código de inversión del qubit. En ese caso, $\mathcal{V}$ era un subespacio vectorial generado por la base $\{|000\rangle, |111\rangle\}$ y estaba estabilizado por el grupo $\langle Z_1Z_2, Z_2Z_3\rangle$.

Calculando la matriz densidad como la distribución uniforme sobre los estados generadores, $\frac{1}{2}(|000\rangle\langle 000| + |111\rangle\langle 111|)$, obtenemos (en la base de estados básicos para $n = 3$) una matriz llena de ceros salvo en la primera y la última posición de la diagonal, que tienen un 1. Si hacemos los cálculos según la expresión (5.7), el resultado coincide con lo anterior:

$$\begin{aligned}\rho &= \frac{1}{2^2}(I + Z_1 \otimes Z_2 \otimes I_3)(I + I_1 \otimes Z_2 \otimes Z_3) \\ &= \text{diag}(1,1,0,0,0,0,1,1) \cdot \text{diag}(1,0,0,1,1,0,0,1) = \text{diag}(1,0,0,0,0,0,0,1)\end{aligned}$$

Demos otro ejemplo que sirve para generar infinidad de matrices densidad. Para estados mezcla, podemos decir que un estado (o sistema) es separable cuando su matriz densidad se puede expresar como producto tensorial de las matrices densidad de cada subsistema (cada qubit). Cada subsistema admite una representación muy sencilla: los estados básicos $|0\rangle$ y $|1\rangle$ tendrán a $|0\rangle\langle 0|$ y $|1\rangle\langle 1|$ por matrices densidad, mientras que el estado de máxima mezcla está asociado a la matriz $\frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) = \frac{I}{2}$.

Tomamos, como ejemplo, un sistema de $n = 7$ qubits que se encuentra inicialmente en estado separable. Los primeros 3 qubits estáran en estado $|0\rangle$, el cuarto y el quinto en estado de máxima mezcla y los dos últimos en estado $|1\rangle$. Entonces, su matriz densidad vendrá dada por

$$\begin{aligned}\rho &= |0\rangle\langle 0| \otimes |0\rangle\langle 0| \otimes |0\rangle\langle 0| \otimes \frac{I_4}{2} \otimes \frac{I_5}{2} \otimes |1\rangle\langle 1| \otimes |1\rangle\langle 1| \\ &= |000\rangle\langle 000| \otimes \frac{I_4}{2} \otimes \frac{I_5}{2} \otimes |11\rangle\langle 11|\end{aligned}$$

[15] Como la traza de un producto es el producto de las trazas (tanto para el producto usual de matrices como para el tensorial), es sencillo ver que se cumple que $\text{Tr}(\rho) = 1$. Además, ρ es un operador autoadjunto por serlo las matrices de Pauli, y de ahí, cada $I + M_i$. Así, podemos afirmar que ρ es una matriz densidad.

Para llevar a cabo nuestra simulación, situamos los r generadores del estabilizador en las filas $n+1, ..., n+r$ y sus correspondientes generadores destabilizadores como las filas $1, ..., r$. Las $2(n-r)$ filas restantes se rellenan con una colección de operadores $\overline{X_i}$ y $\overline{Z_i}$, que conmutan tanto con el estabilizador como con el destabilizador. Elegimos estos operadores de forma que cumplan las siguientes relaciones de conmutatividad: $[\overline{X_i}, \overline{X_j}] = [\overline{Z_i}, \overline{Z_j}] = [\overline{X_i}, \overline{Z_j}] = 0$ si $i \neq j$, y $\{\overline{X_i}, \overline{Z_i}\} = 0$[16]. Colocaremos $\overline{X_i}$ en las filas $r+i$ y $\overline{Z_i}$ en las filas $n+r+i$, para todo $i \in \{1, ..., n-r\}$.

Se empezará la simulación desde el estado mezcla inicial $|0...0\rangle\langle 0...0| \otimes \frac{I_{n-r+1}}{2} \otimes ... \otimes \frac{I_n}{2}$ (0 en los $n-r$ primeros qubits y el estado completamente mezclado en los últimos r qubits). En tal caso, se eligen los operadores $\overline{X_i} = X_{i+r}$ y $\overline{Z_i} = Z_{i+r}$. Notaremos $\bar{i} = i - n$ si $i \geq n+1$, e $\bar{i} = i + n$ si $i \leq n$. En tal caso, la proposición 6 se traducirá en que las filas R_i y R_j conmutarán a no ser que $i = \bar{j}$, en cuyo caso anticonmutarán. La matriz estabilizadora correspondiente a este estado mezcla sería una "matriz identidad" como la que usamos para estados puros.

Pasemos a estudiar el algoritmo en esta situación algo más compleja. Con respecto a la acción de puertas unitarias, podemos decir que se mantiene todo lo dicho anteriormente para los estados puros. Lo que sí cambia es la *medida* de un qubit a, que resulta algo más difícil, apareciendo ahora tres casos distintos:

Caso I: $x_{pa} = 1$ para algún $p \in \{n+1, ..., n+r\}$.

Lo que ocurre es que Z_a anticonmuta con (al menos) un elemento del estabilizador, de forma que la medida es aleatoria. Actualizamos el estado de igual forma que en el Caso I de estados puros.

Caso II: $x_p a = 0$ en todas las filas $p > r$.

Ahora Z_a conmuta con todo el estabilizador y es, de hecho, un elemento suyo. El output de la medida tendrá un resultado determinista, que averiguamos como hicimos en el Caso II para estados puros, sumando todas las filas que anticonmutan con Z_a. La medida no altera el estado del sistema, dejándolo dentro del código $\mathcal{V}$.

Caso III: $x_{pa} = 0$ para todo $p \in \{n+1, ..., n+r\}$, pero $x_{ma} = 1$ para algún $m \in \{r+1, ..., n\} \bigcup \{n+r+1, ..., 2n\}$.

Ahora Z_a conmuta con todos los generadores del estabilizador, pero no forma parte de él. Por ello, el estado cambia pero se queda dentro del código estabilizador. La medida da lugar a un resultado aleatorio, pero la forma de actualizar el estado del sistema difiere respecto a lo que hacemos en el Caso I:

R_m anticonmuta con Z_a y por ello, tomará el papel de R_p en el Caso I: guardamos la fila R_m en la fila $R_{\overline{m}}$ y asignamos a la fila m las etiquetas correspondientes a Z_a, dejando el valor de la fase r_m en manos del azar. Una vez hemos hecho esto, intercambiamos las filas R_{n+r+1} y R_m, y también las filas R_{r+1} y $R_{\overline{m}}$. Por último, lo que antes era r pasa a ser $r+1$: el estabilizador ha ganado un nuevo generador, $R_{n+r+1} \equiv Z_a$, y el correspondiente destabilizador R_{r+1} (el antiguo R_m) anticonmuta con él.

Se sigue cumpliendo lo que vimos en 4: a cada paso, el nuevo grupo estabilizador es $\mathcal{S}' = \langle \pm Z_a, \mathcal{C}_{\mathcal{S}}(Z_a) \rangle$, solo que ahora $\mathcal{C}_{\mathcal{S}}(Z_a)$ coincide con el estabilizador previo, $\mathcal{S}$.

[16] Se entiende que la notación no ha sido arbitraria: los operadores $\overline{X_i}$ y $\overline{Z_i}$ mantienen las mismas relaciones de conmutatividad que las que presentan X_i y Z_i

Es evidente que el número de medidas que podemos hacer está limitado, ya que a partir de la $(n-r)$-ésima medida, el estabilizador ya tendría los n generadores que le permitimos tener como máximo.

Estudiando el algoritmo, nos damos cuenta de que está diseñado para que se verifiquen continuamente las condiciones que da la proposición 6, también para esta situación. Garantizamos así que el tanto el grupo estabilizador como el grupo de Pauli estén correctamente generados por las correspondiente filas de la matriz.

5.3 Eficiencia en el algoritmo de Aaronson-Gottesman

Hemos repetido varias veces que este algoritmo suponía una mejora respecto al de Gottesman-Knill en términos de eficiencia. ¿A qué se debe?

El proceso más costoso en el "viejo" algoritmo era la medida de un qubit cuando ésta era determinista: la eliminación gaussiana que necesitábamos para resolver el sistema asociado era un proceso de $O(n^3)$. Ahora, se ha propuesto un algoritmo que se ahorra este paso: aprovechando el producto simpléctico de etiquetas para detectar las relaciones de conmutatividad bajamos el número de operaciones a $O(n^2)$ [2]. Este nuevo algoritmo iguala, pues, el tiempo para medir un qubit en cualquiera de los casos: no importa si la medida es determinista o aleatoria, ambas tomarán un tiempo $O(n^2)$.

En su *paper*, Aaronson y Gottesman completan este estudio sobre la eficiencia poniendo a prueba su algoritmo con un programa en C que implementa las cuatro operaciones fundamentales que están permitidas en los circuitos estabilizadores: cNOT, Hadamard, fase y medida. El experimento que realizan, simulando un sistema de n qubits, es el siguiente:

Fijado un parámetro $\beta > 0$, se seleccionan aleatoriamente $\lfloor \beta n \log_2 n \rfloor$ puertas del conjunto $\{cNOT, H, P\}$, aplicándose a un(dos) qubit(s) aleatorio(s) en $\{1, ..., n\}$. Una vez aplicadas todas las puertas, se miden todos los qubits desde el primero hasta el n-ésimo por orden.

En la figura 10 se muestra la comparativa de los tiempos de ejecución como función de n para distintos valores de β.

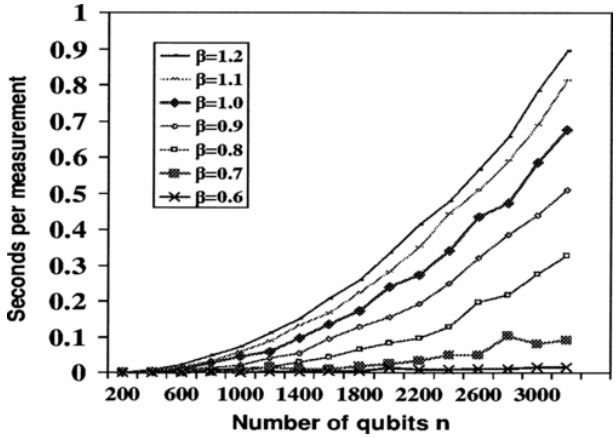


Figura 10: Tiempo medio para simular la medida de un qubit tras aplicar $\lfloor \beta n \log_2 n \rfloor$ puertas sobre n qubits, en un Pentium III de 650MHz con 256MB de RAM [2].

¿Cómo interpretar los resultados de dicha gráfica? Vemos que para β bajos, el tiempo de medida es lineal con el número de qubits, pero que las curvas se van transformando en parábolas a medida que crece β. Esto es debido a que este tiempo aumenta a medida que tenemos que llamar más veces a la subrutina *rowsum*: en el estado inicial no hay generadores de $\mathcal{S}$ tales que $x_{ia} = 1$; pero conforme vamos aplicando más y más puertas de Clifford, hay más operadores que cumplen dicha condición. Como un valor de β grande eleva el número de puertas aplicadas, la medida será también más trabajosa. Se termina, pues, concluyendo que el tiempo por medida se encontrará entre las dependencias lineal y cuadrática, sin poder dar información más detallada.

No obstante, esta mejora tiene un precio. Para poder desarrollar la medida como sugieren estos autores, necesitamos considerar n filas más en la matriz estabilizadora (más otra auxiliar), además de la columna que tiene en cuenta la fase de cada generador. Esto conlleva un aumento de la memoria ocupada: la matriz primitiva tenía dimensión $n \times 2n$, ocupando $2n^2$ bits; por su parte, la nueva versión es una matriz $(2n+1) \times (2n+1) \approx 4n^2$.

Este incremento de la memoria en un factor 2 es más que asumible, teniendo en cuenta que, de no querer asumirlo, tendríamos que lidiar con un tiempo que aumenta en un factor n, como impone el antiguo algoritmo. Pensemos en la simulación de un sistema de 3000 qubits: $4 \cdot 3000^2 = 36 \cdot 10^6$ bits ≈ 4.25 MB es una memoria insignificante respecto a un ordenador de mesa, mientras que un proceso 3000 veces más rápido sí supone una mejora enorme.

Un último apunte que [2] da sobre la eficiencia de su algoritmo tiene que ver con la preparación del estado inicial, esto es, con la matriz estabilizadora de partida:

La representación de estados puros no necesita ningún cálculo adicional: sólo necesitamos introducir las etiquetas, ya conocidas, de los generadores que conforman el grupo estabilizador. En cambio, para la representación de estados mezcla estabilizadores, se han de calcular los $2(n-r)$ operadores $\overline{X_i}$, $\overline{Z_i}$, que acompañan a los elementos estabilizadores y destabilizadores en la matriz. Para ello, se resuelve un sistema de ecuaciones que recoge las relaciones de conmutatividad que existen entre todas las filas de la matriz. Este es un proceso que tarda un tiempo $O(n^3)$.

En definitiva, la representación de estados puros es un proceso $O(n^2)$ en su conjunto, mientras que la de estados mezcla sube sus tiempos a $O(n^3)$. De todas formas, este segundo proceso no estaba siquiera contemplado por el algoritmo de Gottesman-Knill, por lo que el algoritmo de Aaronson-Gottesman supone ganancias, se mire por donde se mire.

6 Conclusiones

Este trabajo constituye una buena forma de ver cómo Física y Matemática son ciencias inseparables: para adentrarnos en la simulación clásica de circuitos cuánticos, nos vimos en la necesidad de recurrir a viejos conocidos como son la teoría de grupos o los espacios de Hilbert, pero también de aprender estructuras nuevas para un estudiante de Matemáticas como son los tensores o el producto simpléctico.

Aunque uno podría pensar en la teoría de grupos como un campo demasiado abstracto y puro para buscar aplicaciones, ésta ha demostrado ser una gran aliada en el estudio de las simulaciones: no sólo permite formular una representación completamente nueva de ciertos estados cuánticos, sino que también aparece a la hora de mejorar la eficiencia de los algoritmos estudiados.

Fue en esta mejora donde brilló el producto simpléctico. Éste nos otorgaba una manera muy eficiente de comprobar las relaciones de conmutatividad entre operadores, que sería el motivo principal para que el algoritmo de Aaronson-Gottesman [2] se impusiera sobre el algoritmo de Gottesman-Knill[1], pues se pasa de una complejidad $O(n^3)$ a $O(n^2)$ al implementar las novedades.

Estos pequeños avances nos permiten hacernos una idea del tipo de cosas en las que se trabaja en el ámbito de la Computación Cuántica: con un breve vistazo a algunos artículos de la bibliografía, se aprecia que estas herramientas suponen la clave de algunas investigaciones bastante actuales. En publicaciones más contemporáneas que las aquí revisadas, como [21] y [22], se emplean funciones de quasi-probabilidad y representaciones nuevas para conseguir simulaciones eficientes de estados aún más generales.

En definitiva, simulando clásicamente los sistemas cuánticos, podemos entender el futuro desempeño de los ordenadores cuánticos, y esto es un paso necesario antes de plantearse la construcción de los mismos.

Sin embargo, puede que llegue el día en que los ordenadores cuánticos sobrepasen con creces las capacidades de cualquier ordenador clásico. Es por ello que se debe plantear un debate acerca de qué se debe y qué no se debe hacer con estas, a priori, potentes máquinas. Las posibilidades de un poder computacional aún mayor que el actual son muy diversas: desde mejoras en la comunicación y en la cyberseguridad hasta la resolución de cálculos de complejidad inimaginable a día de hoy.

Como todo avance precedente, habrá aplicaciones que conlleven nuevos descubrimientos en Ciencia, fines militares e incluso repercusiones en nuestro día a día; de ahí que tengamos que evitar que se convierta en una amenaza y quedarnos con sus bondades, que prometen ser muchas.

Referencias

[1] N. Nielsen, I.L. Chuang,
Quantum Computation and Quantum Information, Cambridge U. P., Cambridge, 2010

[2] S. Aaronson, D. Gottesman,
Improved Simulation of Stabilizer Circuits, Phys. Rev. A 70, 052328 (2004),
arXiv: quant-ph/0406196

[3] D. Dummit, R. Foote,
Abstract algebra (3rd ed.), John Wiley & Sons, 2004

[4] F.J. Pérez González,
Análisis Funcional en Espacios de Banach, Departamento de Análisis Matemático,
Universidad de Granada (2019).
https://www.ugr.es/~fjperez/textos/Analisis_Funcional_en_Espacios_de_Banach.pdf

[5] R.B. Griffiths,
Consistent Quantum Theory. Cambridge University Press, 2002.

[6] M.A. de Gosson,
Symplectic methods in Harmonic Analysis and in Mathematical Physics.
26 Birkhauser, Basel 2011.

[7] J. Pérez Muñoz,
Álgebra Lineal y Geometría II, Departamento de Geometría y Geometría,
Universidad de Granada (2020)

[8] W. Hackbusch,
Tensor Spaces and Numerical Tensor Calculus,
Springer International Publishing, 2019.

[9] B.J. Broxson,
The Kronecker Product , UNF Graduate Theses and Dissertations 25, 2006.
https://digitalcommons.unf.edu/etd/25

[10] N. Zettili,
Quantum Mechanics: Concepts and Applications (2nd Revised ed.), Wiley, 2009

[11] J. Sakurai, J. Napolitano,
Modern quantum mechanics (2nd ed.). Addison Wesley, 2011

[12] J. Bermejo-Vega, M. Van den Nest,
Classical simulations of Abelian-group normalizer circuits with intermediate measurements, Quantum Information & Computation, Volume 14, Issue 3-4,
Marzo 2014, arXiv: 1210.3637 [quant-ph]

[13] G. Tóth, O. Gühne,
Entanglement detection in the stabilizer formalism, Phys. Rev. A 72, 022340 (2005),
arXiv: quant-ph/0501020

[14] C.H. Bennett, G. Brassard, C.Crépeau, R. Jozsa, A. Peres, W.K. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*, Phys. Rev. Lett. 70, 1895 (1993), DOI: https://doi.org/10.1103/PhysRevLett.70.1895

[15] S.J. Devitt, K. Nemoto, W.J. Munro, *Quantum Error Correction for Beginners*, Rep. Prog. Phys. 76 (2013) 076001, arXiv: 0905.2794 [quant-ph]

[16] T. Brun, I. Devetak, M. Hsieh, *Correcting Quantum Errors with Entanglement*, Science 314, 436-439 (2006), arXiv: quant-ph/0610092

[17] Artur Ekert, *Introduction to Quantum Information Science* (Lectures) https://www.youtube.com/c/ArturEkert/featured

[18] D. Gottesman, *Theory of fault-tolerant quantum computation*, Phys. Rev. A, 57 (1). pp. 127-137 DOI: https://doi.org/10.1103/PhysRevA.57.127

[19] J. Bermejo-Vega, M. Van den Nest, *Classical simulations of Abelian-group normalizer circuits with intermediate measurements*, Quantum Information & Computation, Volume 13, Issue 11-12, Septiembre 2018, arXiv:1201.4867 [quant-ph]

[20] B.Eastin, S.T. Flammia, *qcircuit 2.6.0 Tutorial* , Department of Physics and Astronomy, University of New Mexico.

[21] V. Vitch, C. Ferrie, D. Gross, J. Emerson, *Negative Quasi-Probability as a Resource for Quantum Computation*, New J. Phys. 14, 113011 (2012), arXiv: 1201.1256 [quant-ph]

[22] R. Raussendorf, J. Bermejo-Vega, E. Tyhurst, C. Okay, M. Zurel, *Phase space simulation method for quantum computation with magic states on qubits*, Phys. Rev. A 101, 012350 (2020), arXiv:1905.05374 [quant-ph]