Un poco de computación cuántica: Algoritmos más comunes

Guillermo Morales-Luna
Centro de Investigación y Estudios Avanzados del IPN
(CINVESTAV-IPN)
gmorales@cs.cinvestav.mx

11 de diciembre de 2003

Resumen

El presente escrito constituye una introducción a la computación cuántica, es más bien un cursillo "tutorial" y el único reclamo de originalidad es la de su presentación. De hecho, ha sido grande la tentación de sutituir el adjetivo "cuántica" en "computación" por la frase adjetival "paralela basada en álgebra exterior". Nuestra presentación deja de lado la discusión sobre la plausibilidad de la computación cuántica y las nociones inherentes de la física cuántica involucradas. Aquí sólo nos concentraremos en presentar este paradigma como uno abstracto de cómputo y en él presentamos sus funciones primitivas, los esquemas de composición y la codificación de entradas y salidas. Nos basamos en planteamientos ortodoxos de álgebra exterior y utilizamos una notación más acorde con esta última.

Antes que nada presentamos las nociones básicas necesarias de álgebra exterior. Al iniciar el concepto de computación cuántica propiamente presentamos a los espacios de Hilbert donde se realiza, la noción de *qubit* e ilustramos el mecanismo de cómputo con el algoritmo ya clásico de Deutsch-Josza. Luego presentamos un algoritmo para calcular transformaciones discretas de Fourier en tiempo lineal. Finalmente, presentamos el célebre algoritmo de Shor para la factorización de enteros en tiempo polinomial.

Contents

1 Productos tensoriales

1.1 Productos de vectores y espacios

Sea U un espacio vectorial sobre el campo $\mathbb{C}$ de los números complejos. Denotemos por $\mathcal{L}(U,V)$ al espacio de transformaciones lineales de U en V. El *dual* del espacio U es $U^* = \mathcal{L}(U,\mathbb{C})$. Si $u^* \in U^*$ escribimos, para cada $u \in U$, $\langle u^*|u\rangle := u^*(u)$. $\langle\cdot|\cdot\rangle : U^* \times U \to \mathbb{C}$ es una transformación bilineal.

Sea V otro espacio vectorial también sobre $\mathbb{C}$. El *producto tensorial* de U con V es $U \otimes V = \mathcal{L}(V^*, U)$. Se tiene:

Propiedad 1.1 *$U \times V$ se identifica con un subconjunto de $U \otimes V$.*

En efecto, consideremos $\Phi : U \times V \to U \otimes V$ tal que $\forall (u,v) \in U \times V$, $\Phi(u,v) \in \mathcal{L}(V^*, U)$ es la transformación $\Phi(u,v) : w^* \mapsto \langle w^*|v\rangle u$. Claramente Φ es bilineal. Se tiene que $\Phi(u_1,v_1) = \Phi(u_2,v_2)$ si y sólo si existe $k \in \mathbb{C}$ tal que $(u_1,v_1) = (ku_2, k^{-1}v_2)$. Esta última condición define una relación de equivalencia $\equiv_0$ en $U \times V$. Así pues, el espacio cociente $(U \times V)/\equiv_0$ se identifica con un subespacio de $U \otimes V$. De hecho, la aplicación $\Phi(u,v) \in \mathcal{L}(V^*, U)$ se denota como $u \otimes v = \Phi(u,v)$ y se dice ser el *producto tensorial* del vector u con el vector v. Debido a la linealidad de los operadores involucrados se tiene que valen las relaciones siguientes:

$$\begin{aligned}
(zu) \otimes v &= z(u \otimes v) \\
u \otimes (zv) &= z(u \otimes v) \\
(u_1 + u_2) \otimes v &= (u_1 \otimes v) + (u_2 \otimes v) \\
u \otimes (v_1 + v_2) &= (u \otimes v_1) + (u \otimes v_2)
\end{aligned}$$

Resulta claro que el producto tensorial de vectores no es conmutativo, ni siquiera cuando $U = V$.

Propiedad 1.2 *Si $\dim U = m$ y $\dim V = n$ entonces $\dim(U \otimes V) = mn$.*

En efecto, si V es de dimensión finita, entonces su dual V^* es de la misma dimensión, n, y obviamente $\dim(\mathcal{L}(V^*, U)) = nm$.

Así pues, si $U = \mathbb{C}^m$ y $V = \mathbb{C}^n$ entonces, prácticamente, $U \otimes V = \mathbb{C}^{mn}$.

Propiedad 1.3 *Si $B_U = \{u_0, u_1, \ldots, u_{m-1}\}$ es una base de U y $B_V = \{v_0, v_1, \ldots, v_{n-1}\}$ es una base de V entonces $(u_i \otimes v_j)_{i<m, j<n}$ es una base de $U \otimes V$, donde, para cada i, j, $u_i \otimes v_j$ es la aplicación $w^* = \sum_{k=0}^{n-1} w_j v_j^* \mapsto w_j u_i$. Esta base se dice ser la* base producto.

En efecto, $B_{V^*} = \{v_0^*, v_1^*, \ldots, v_{n-1}^*\}$ es una base del dual V^*, donde $\langle v_{j_1}^*|v_{j_2}\rangle = \delta_{j_1 j_2}$. Ahora, respecto a las bases B_U y B_V, la aplicación $u_i \otimes v_j$ se representa mediante la matriz $D_{ij} = (\delta_{i_1 j_1 ij})_{i_1<m, j_1<n}$, donde $\delta_{i_1 j_1 ij} = 1$ si y sólo si $(i_1, j_1) = (i, j)$ y $\delta_{i_1 j_1 ij} = 0$ si y sólo si $(i_1, j_1) \neq (i, j)$. De manera un poco más general, escribamos cada $u \in U$ como $u = \sum_{i=0}^{m-1} a_i u_i$ y $v = \sum_{j=0}^{n-1} b_j v_j$, también cada $w^* \in V^*$ como $w^* = \sum_{j=0}^{n-1} c_j v_j^*$. En consecuencia, $\langle w^*|v\rangle = \sum_{j=0}^{n-1} b_j c_j$, y $(u \otimes v)(w^*) = \sum_{i=0}^{m-1} a_i \left(\sum_{j=0}^{n-1} b_j c_j\right) u_i$.

1.2 Productos de transformaciones lineales

Supongamos que U_1, U_2, V_1, V_2 son espacios vectoriales, de dimensiones respectivas m_1, m_2 y n_1, n_2, y que $K : U_1 \to U_2$ y $L : V_1 \to V_2$ son sendas transformaciones lineales. Las transformaciones lineales *duales* $K^* : U_2^* \to U_1^*$ y $L^* : V_2^* \to V_1^*$ están definidas mediante las relaciones

$$\begin{aligned}
\langle K^*(u_2^*)|u_1\rangle &= \langle u_2|K(u_1)\rangle \\
\langle L^*(v_2^*)|v_1\rangle &= \langle v_2|L(v_1)\rangle
\end{aligned}$$

Propiedad 1.4 *Si K se representa, respecto a bases B_{U_1} y B_{U_2}, mediante una matriz $M_K \in \mathbb{C}^{m_2 \times m_1}$ entonces K^* se representa, respecto a las bases duales, mediante la matriz* traspuesta *$M_K^T \in \mathbb{C}^{m_1 \times m_2}$.*

Ahora definamos una transformación $K \otimes L : U_1 \otimes V_1 \to U_2 \otimes V_2$ haciendo $(K \otimes L)(u_1 \otimes v_1) = K(u_1) \otimes L(v_1)$.

Propiedad 1.5 *Si K se representa, respecto a bases B_{U_1} y B_{U_2}, mediante la matriz $M_K \in \mathbb{C}^{m_2 \times m_1}$ y L se representa, respecto a bases B_{V_1} y B_{V_2}, mediante la matriz $M_L \in \mathbb{C}^{n_2 \times n_1}$ entonces $(K \otimes L)$ se representa, respecto a las bases productos, mediante la matriz* producto tensorial $M_K \otimes M_L \in \mathbb{C}^{m_2 n_2 \times m_1 n_1}$ *siguiente:*

$$M_K \otimes M_L = \begin{bmatrix} m_{00}^{(K)} M_L & m_{01}^{(K)} M_L & \cdots & m_{0,m_1-1}^{(K)} M_L \\ m_{10}^{(K)} M_L & m_{11}^{(K)} M_L & \cdots & m_{1,m_1-1}^{(K)} M_L \\ \vdots & \vdots & \ddots & \vdots \\ m_{m_2-1,0}^{(K)} M_L & m_{m_2-1,1}^{(K)} M_L & \cdots & m_{m_2-1,m_1-1}^{(K)} M_L \end{bmatrix}$$

Ahora bien, si U es un espacio vectorial de dimensión m y $K : U \to U$ es lineal, definimos recursivamente: $K^{\otimes 1} = K$, $K^{\otimes n} = K^{\otimes(n-1)} \otimes K$. Naturalmente, $K^{\otimes n}$ es la *n-ésima potencia tensorial de K*. Si $M_K = (m_{ij})_{i,j<m}$ es la matriz cuadrada de orden m que representa a K respecto a una cierta base B_U, se tiene que $K^{\otimes n}$ quedará representada por la matriz $M_{K^{\otimes n}} = \left(m_{ij}^{(n)}\right)_{i,j<m^n}$ determinada como sigue:

Cada índice entero $i < m^n$ se escribe en base m como una palabra de n dígitos $0, \ldots, m-1$, $i = \sum_{\iota=0}^{n-1} \xi_\iota m^\iota = (\xi_{n-1} \cdots \xi_1 \xi_0)_m = \boldsymbol{\xi}(i)$. Para una tal palabra $\boldsymbol{\xi} = \xi_{n-1} \cdots \xi_1 \xi_0$, definamos $\mathrm{car}(\boldsymbol{\xi}) = \xi_0$ y $\mathrm{cdr}(\boldsymbol{\xi}) = \xi_{n-1} \cdots \xi_1$. Evidentemente, vistas las palabras como representaciones de números en base m: $(\boldsymbol{\xi})_m = m(\mathrm{cdr}(\boldsymbol{\xi}))_m + \mathrm{car}(\boldsymbol{\xi})$, $\mathrm{car}(\boldsymbol{\xi}) = \boldsymbol{\xi} \bmod m$ y $\mathrm{cdr}(\boldsymbol{\xi}) = (\boldsymbol{\xi} - \mathrm{car}(\boldsymbol{\xi}))/m$.

Debido a la propiedad 1.5, se tiene la recurrencia

$$m_{\boldsymbol{\xi}(i),\boldsymbol{\xi}(j)}^{(n)} = m_{\mathrm{cdr}(\boldsymbol{\xi}(i)),\mathrm{cdr}(\boldsymbol{\xi}(j))}^{(n-1)} \cdot m_{\mathrm{car}(\boldsymbol{\xi}(i)),\mathrm{car}(\boldsymbol{\xi}(j))} \tag{1}$$

con la cual es posible calcular las entradas de la matriz $M_{K^{\otimes n}}$ siguiendo las representaciones en base m de los índices correspondientes.

2 Nociones básicas de Computación Cuántica

2.1 Postulado de medición

En [2] se presentó la base de la computación cuántica.

Sea $\mathbb{C}$ el campo de los números complejos, y para cada m, n sea $\mathbb{C}^{m \times n}$ el espacio de matrices de orden $m \times n$, es decir, de matrices con m renglones y n columnas, con entradas números complejos. Para cada matriz $M = (m_{ij})_{i,j} \in \mathbb{C}^{m \times n}$ su *transpuesta hermitiana* es $M^H = \left(m_{ji}^H\right)_{ji} \in \mathbb{C}^{n \times m}$ donde para cada pareja de índices $(i, j) \in [\![0, m-1]\!] \times [\![0, n-1]\!]$, $m_{ji}^H = \overline{m_{ij}}$ (si $z = a + ib \in \mathbb{C}$ es un número complejo, naturalmente $\overline{z} = a - ib \in \mathbb{C}$ es su *conjugado*). Una matriz $M = (m_{ij})_{i,j} \in \mathbb{C}^{m \times n}$ se dice ser *unitaria* si $M^H M = \mathbf{1}_{nn}$, donde $\mathbf{1}_{nn}$ denota a la matriz *identidad* de orden $n \times n$.

Al subconjunto consistente de los vectores columnas unitarias en $\mathbb{C}^{m \times 1}$ (es decir, el espacio de vectores columnas de dimensión m) se le llama *conjunto de estados* de un sistema físico cerrado, y la dimensión m se conoce como el *grado de libertad* del sistema. En $\mathbb{C}^{m \times 1}$ se tiene que cada estado es un vector en la esfera euclidiana unitaria de $\mathbb{C}^m$. Sea pues $E_m = \{\mathbf{v} \in \mathbb{C}^m | 1 = \mathbf{v}^H \mathbf{v} =: \langle \mathbf{v} | \mathbf{v} \rangle\}$ el conjunto de estados.

Sea $\mathbf{e}_j = (\delta_{ij})_{i<m}$ el j-ésimo vector de la base canónica de $\mathbb{C}^m$. Se tiene que todo vector de la base canónica es un estado. Se dice que un estado $\mathbf{v} = (v_{i1})_{i<m}$ *produce* la *salida* i con una probabilidad $|v_{i1}|^2 = \mathrm{Re}(v_{i1})^2 + \mathrm{Im}(v_{i1})^2$. Se tiene el siguiente

Postulado de Medición: Si el estado actual es $\mathbf{v} = (v_{i1})_{i<m}$ entonces, para cada $i < m$, con probabilidad $|v_{i1}|^2$ se realiza lo siguiente: Se emite la respuesta i y se transita al estado $\mathbf{e}_i$; es decir este último será el estado actual en el paso siguiente.

De hecho el proceso de medición se realiza al final de cualquier algoritmo cuántico, así que el último estado actual al que se refiere en su enunciado es el estado *final*.

Ahora, sea $U \in \mathbb{C}^{m \times m}$ una matriz unitaria cuadrada de orden $m \times m$. U determina una transformación ortogonal $\mathbb{C}^m \to \mathbb{C}^m$: $\mathbf{v} \mapsto U\mathbf{v}$. De hecho, al restringirla a E_m se tiene una transformación $E_m \to E_m$. U se dice ser una *compuerta cuántica*. Un *algoritmo cuántico* es la composición de un número finito de compuertas cuánticas.

2.2 *qubits* y palabras de longitud finita

En particular, para $m = 2$, la base canónica del espacio $\mathbb{H}_1 = \mathbb{C}^2$ consta de los vectores $\mathbf{e}_0 = [1\ \ 0]^T$ y $\mathbf{e}_1 = [0\ \ 1]^T$. Si $z_0, z_1 \in \mathbb{C}$ son complejos tales que $|z_0|^2 + |z_1|^2 = 1$, entonces $z_0\mathbf{e}_0 + z_1\mathbf{e}_1$ es un estado, llamado *qubit*, apócope inglés de *bit cuántico* (*quantum bit*).

Identificamos al primer vector básico $\mathbf{e}_0$ con el valor de verdad *falso*, o *cero*, y al segundo $\mathbf{e}_1$ con el valor de verdad *verdadero*, o *uno*. Así pues, cada estado es una "superposición" de ambos valores cero y uno.

Para cada $n > 1$, definimos recursivamente $\mathbb{H}_n = \mathbb{H}_{n-1} \otimes \mathbb{H}_1$. De aquí resulta que $\dim(\mathbb{H}_n) = 2^n$ y una base de este espacio es $B_{\mathbb{H}_n} = \left(\mathbf{e}_{\varepsilon_{n-1}\cdots\varepsilon_1\varepsilon_0}\right)_{\varepsilon_{n-1},\ldots,\varepsilon_1,\varepsilon_0\in\{0,1\}}$, donde, puesto de manera recursiva, $\mathbf{e}_{\varepsilon_{n-1}\cdots\varepsilon_1\varepsilon_0} = \mathbf{e}_{\varepsilon_{n-1}\cdots\varepsilon_1} \otimes \mathbf{e}_{\varepsilon_0}$. Aquí queremos llamar la atención del lector para que tome en cuenta el cambio de nuestra notación respecto a la asumida de manera convencional en el mundo de la Física y de la Computación Cuántica. En general se suele escribir

$$|\varepsilon_{n-1}\cdots\varepsilon_1\varepsilon_0\rangle := \mathbf{e}_{\varepsilon_{n-1}\cdots\varepsilon_1\varepsilon_0} = \mathbf{e}_{\varepsilon_{n-1}} \otimes \cdots \otimes \mathbf{e}_{\varepsilon_1} \otimes \mathbf{e}_{\varepsilon_0} =: |\varepsilon_{n-1}\rangle \cdots |\varepsilon_1\rangle\,|\varepsilon_0\rangle \tag{2}$$

Evidentemente, cada índice $i \in [\![0, 2^n - 1]\!]$ puede escribirse en binario como una cadena de bits de longitud n: $i = (\varepsilon_{n-1}\cdots\varepsilon_1\varepsilon_0)_2$. Así pues identificaremos a cada índice con la cadena que lo representa: $i \leftrightarrow \boldsymbol{\varepsilon} = \varepsilon_{n-1}\cdots\varepsilon_1\varepsilon_0$. Mediante esta identificación: $[\![0, 2^n - 1]\!] \approx \{0,1\}^n$.

Si $\mathbf{z} \in E_{2^n}$ es un vector en la esfera unitaria euclidiana en $\mathbb{H}_n$ entonces $\sum_{\boldsymbol{\varepsilon}\in\{0,1\}^n} z_{\boldsymbol{\varepsilon}}\mathbf{e}_{\boldsymbol{\varepsilon}}$ es un estado correspondiente a una *palabra de información de longitud n*, y es también la concatenación de n qubits.

2.3 Compuertas cuánticas

Para $n = 1$, consideraremos las siguientes *compuertas básicas*, llamadas también *operadores cuánticos*:

Identidad. $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. $I : \mathbb{H}_1 \to \mathbb{H}_1$ es el operador identidad.

Negación. $N = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Se tiene $N : \begin{bmatrix} z_0 \\ z_1 \end{bmatrix} \mapsto \begin{bmatrix} z_1 \\ z_0 \end{bmatrix}$. N es unitaria y tiene como función *permutar señales*, es de hecho "una reflexión a lo largo de la diagonal principal".

Hadamard. $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Se tiene $H : \begin{bmatrix} z_0 \\ z_1 \end{bmatrix} \mapsto \frac{1}{\sqrt{2}}\begin{bmatrix} z_0 + z_1 \\ z_0 - z_1 \end{bmatrix}$. H es unitaria y tiene como función "reflejar el plano respecto al eje x y rotar luego un ángulo de $\frac{\pi}{4}$ radianes, en sentido opuesto a las manecillas del reloj".

Naturalmente, $N^{\otimes n}$ y $H^{\otimes n}$ son sendas compuertas en $\mathbb{H}_n$. Las matrices que las representan, respecto a la base producto $B_{\mathbb{H}_n}$, pueden ser calculadas mediante la relación (1).

Observamos aquí, primeramente, que $N^{\otimes n}$ actúa como el "complemento a 2^n-1", es decir, en los vectores básicos se tiene

$$N^{\otimes n}\left(\mathbf{e}_{\varepsilon_{n-1}\cdots\varepsilon_1\varepsilon_0}\right) = \mathbf{e}_{\delta_{n-1}\cdots\delta_1\delta_0} \tag{3}$$

donde $(\varepsilon_{n-1}\cdots\varepsilon_1\varepsilon_0)_2 + (\delta_{n-1}\cdots\delta_1\delta_0)_2 = 2^n - 1$.

Observamos también que

$$\begin{aligned} H^{\otimes 1}(\mathbf{e}_0) &= \frac{1}{\sqrt{2}}(\mathbf{e}_0 + \mathbf{e}_1) \\ H^{\otimes 2}(\mathbf{e}_{00}) &= \frac{1}{(\sqrt{2})^2}(\mathbf{e}_{00} + \mathbf{e}_{01} + \mathbf{e}_{10} + \mathbf{e}_{11}) \end{aligned}$$

y de manera general

$$H^{\otimes n}(\mathbf{e}_{0\cdots0}) = \frac{1}{(\sqrt{2})^n}\left(\sum_{\boldsymbol{\varepsilon}\in\{0,1\}^n} \mathbf{e}_{\boldsymbol{\varepsilon}}\right) \tag{4}$$

es decir, el operador $H^{\otimes n}$ aplicado al primer vector básico $\mathbf{e}_{0\cdots0}$ produce el estado que ""promedia" a todos los demás con pesos uniformes".

Negación controlada. Sea $C : \mathbb{H}_2 \to \mathbb{H}_2$ la transformación lineal que sobre los vectores básicos actúa $\mathbf{e}_x \otimes \mathbf{e}_y \mapsto \mathbf{e}_x \otimes \mathbf{e}_{x\oplus y}$ (recuerdo una vez más que $\oplus$ es la disyunción excluyente, o más bien la adición módulo 2). Esta transformación se llama *negación controlada* debido a que en su salida, el segundo qubit es la negación del primero sólo si en la entrada el segundo qubit "estaba prendido". Esto puede verse como que el segundo qubit de entrada sirve de "control" para aplicar el operador de negación al primero, el cual hace las veces de "argumento".

C no es el producto tensorial de dos transformaciones unitarias en $\mathbb{H}_1$. Se tiene que C queda representado, respecto a la base canónica de $\mathbb{H}_2$ por la matriz

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Negación controlada cambiada. Sea $D : \mathbb{H}_2 \to \mathbb{H}_2$ la transformación lineal tal que $(\mathbf{x}, \mathbf{y}) \mapsto D(\mathbf{x}, \mathbf{y}) = C(\mathbf{y}, \mathbf{x})$ que tan sólo cambia los roles de variable de control y variable de argumento. Se tiene

$$D = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

En el espacio de transformaciones unitarias, C y D generan un subgrupo con la operación de composición. La tabla de multiplicación del subgrupo es de la forma:

$\circ$	I	C	D	CD	DC	CDC
I	I	C	D	CD	DC	CDC
C	C	I	CD	D	CDC	DC
D	D	DC	I	CDC	C	CD
CD	CD	CDC	C	DC	I	D
DC	DC	D	CDC	I	CD	C
CDC	CDC	CD	DC	C	D	I

Alternativamente, podemos decir que este grupo queda *presentado* por su unidad I, dos generadores C, D y la relación $CDC = DCD$. De hecho este grupo es isomorfo al grupo de permutaciones de 3 elementos, S_3. En efecto, si $\rho = (1, 2)$ es la *reflexión* y $\phi = (1, 2, 3)$ es el ciclo de orden 3, entonces se puede identificar $C \leftrightarrow \rho$, $D \leftrightarrow \rho \circ \phi$.

Reversos. Entre los elementos que aparecen en el ejemplo anterior, $R_2 = CDC : \mathbb{H}_2 \to \mathbb{H}_2$ queda representado, respecto a la base canónica, mediante la matriz

$$R_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

es decir, es tal que $R_2(\mathbf{e}_i \otimes \mathbf{e}_j) = \mathbf{e}_j \otimes \mathbf{e}_i$. De hecho, para cada $n \geq 2$, actuando sobre la base canónica, se tiene:

$$R_n = R_2^{\otimes n}\left(\mathbf{e}_{\varepsilon_{n-1}\cdots\varepsilon_1\varepsilon_0}\right) = \mathbf{e}_{\varepsilon_0\varepsilon_1\cdots\varepsilon_{n-1}} \tag{5}$$

es decir, el efecto de este operador es *revertir* el orden de la "palabra de entrada", por lo cual, R_n se dice ser el *operador reverso*.

Un estado en $\mathbb{H}_n$, digamos $\boldsymbol{\sigma}(\mathbf{z}) = \sum_{\boldsymbol{\varepsilon}\in\{0,1\}^n} z_{\boldsymbol{\varepsilon}}\mathbf{e}_{\boldsymbol{\varepsilon}}$ está determinado por las 2^n coordenadas del vector $\mathbf{z} \in E_{2^n}$. Si $U : \mathbb{H}_n \to \mathbb{H}_n$ es una compuerta cuántica, el estado $\boldsymbol{\sigma}(U\mathbf{z})$ al que arriba al aplicársele el operador U consta también de 2^n coordenadas. De esta manera, un cálculo que involucra un número exponencial de términos se hace en "un paso" de cómputo cuántico y es posible así acelerar el proceso de corrida.

Toda combinación lineal de elementos en la base cuyos coeficientes formen un punto en la esfera euclidiana unitaria de $\mathbb{C}^{2^n}$ es un estado, $\boldsymbol{\sigma}(\mathbf{z})$. Se dice que el estado es *descomponible*, o *factorizable*, si es de la forma $\mathbf{z}_1 \otimes \cdots \otimes \mathbf{z}_n = \boldsymbol{\sigma}(\mathbf{z})$, con $\mathbf{z}_i \in \mathbb{H}_1$. Un estado que no es descomponible se dice ser *revuelto* (*entangled state*).

(x,z)	$(x,(z+f(x)) \bmod 2)$
(0,0)	$(0,f(0))$
(0,1)	$(0,\overline{f(0)})$
(1,0)	$(1,f(1))$
(1,1)	$(1,\overline{f(1)})$

Recuadro 1: Acción de la matriz unitaria U_f en el algoritmo de Deutsch-Jozsa.

2.4 Evaluación de funciones booleanas

Sea $V=\{0,1\}$ el conjunto de valores de verdad clásicos. Obviamente hay 2^{2^n} funciones booleanas $V^n \to V$ y hay 2^{n2^n} funciones $V^n \to V^n$. Cada una de las 2^n asignaciones $\boldsymbol{\varepsilon}=(\varepsilon_{n-1},\ldots,\varepsilon_1,\varepsilon_0)\in V^n$ se puede poner en correspondencia con el vector $\mathbf{e}_{\boldsymbol{\varepsilon}}\in\mathbb{H}_n$ de la base "canónica" de $\mathbb{H}_n$. Sea U una matriz permutación de orden $2^{n+1}\times 2^{n+1}$ tal que $U(\mathbf{e}_{\boldsymbol{\varepsilon}}\otimes\mathbf{e}_{\mathbf{0}})=(\mathbf{e}_{\boldsymbol{\varepsilon}}\otimes\mathbf{e}_{f(\boldsymbol{\varepsilon})})$. U es pues unitaria. Sea $A\subset V^n$ un conjunto no-vacío de asignaciones y sea $a=\mathrm{card}(A)$ su cardinalidad. Al considerar el estado $\mathbf{u}_A=\frac{1}{\sqrt{a}}\sum_{\boldsymbol{\varepsilon}\in A}\mathbf{e}_{\boldsymbol{\varepsilon}}\otimes\mathbf{e}_{\mathbf{0}}$ se tiene $U(\mathbf{u}_A)=\frac{1}{\sqrt{a}}\sum_{\boldsymbol{\varepsilon}\in A}\mathbf{e}_{\boldsymbol{\varepsilon}}\otimes\mathbf{e}_{f(\boldsymbol{\varepsilon})}$ y así en un solo paso de cómputo se calcula a un promedio ponderado de la imagen de las asignaciones con índice en A. El proceso final de medición consiste en la selección de una pareja $\mathbf{e}_{\boldsymbol{\varepsilon}}\otimes\mathbf{e}_{f(\boldsymbol{\varepsilon})}$, $\boldsymbol{\varepsilon}\in A$, cada una con probabilidad $\frac{1}{a}$.

2.5 Algoritmo de Deutsch-Jozsa

Sea $V=\{0,1\}$ el conjunto de valores de verdad clásicos. De las $2^2=4$ funciones booleanas $f:V\to V$ dos son constantes y las otras dos son equilibradas. Al nombrarlas

$$f_0:\begin{array}{ccc}0&\mapsto&0\\1&\mapsto&0\end{array},\quad f_1:\begin{array}{ccc}0&\mapsto&0\\1&\mapsto&1\end{array},\quad f_2:\begin{array}{ccc}0&\mapsto&1\\1&\mapsto&0\end{array},\quad f_3:\begin{array}{ccc}0&\mapsto&1\\1&\mapsto&1\end{array}$$

se tiene que las funciones constantes son f_0 y f_3, y las equilibradas son f_1 y f_2.

El propósito del algoritmo de Deutsch-Jozsa es decidir, para una f dada, si acaso es constante o equilibrada "utilizando un solo paso de cómputo".

Sea U_f una matriz permutación de orden $2^2\times 2^2$ tal que $U_f(\mathbf{e}_x\otimes\mathbf{e}_z)=(\mathbf{e}_x\otimes\mathbf{e}_{(z+f(x)) \bmod 2})$. U_f es pues unitaria. De hecho es muy similar al funcionamiento de la compuerta "negación controlada", salvo que en aquella, la función f es propiamente la identidad. En la tabla 1 ilustramos la acción de U_f refiriéndonos solamente a los índices de vectores básicos.

Considerando el operador de Hadamard H, hagamos $H_2=H\otimes H$. Primero se tiene, $H(\mathbf{e}_0)=\mathbf{x}_0=\frac{1}{\sqrt{2}}(\mathbf{e}_0+\mathbf{e}_1)$ y $H(\mathbf{e}_0)=\mathbf{x}_1=\frac{1}{\sqrt{2}}(\mathbf{e}_0-\mathbf{e}_1)\in\mathbb{H}_1$ y luego $H_2(\mathbf{e}_0\otimes\mathbf{e}_1)=H(\mathbf{e}_0)\otimes H(\mathbf{e}_1)=\mathbf{x}_0\otimes\mathbf{x}_1$. Claramente, $\mathbf{x}_0\otimes\mathbf{x}_1=\frac{1}{2}(\mathbf{e}_{00}-\mathbf{e}_{01}+\mathbf{e}_{10}-\mathbf{e}_{11})\in\mathbb{H}_2$. Por tanto,

$$\begin{aligned}
U_f(\mathbf{x}_0\otimes\mathbf{x}_1) &= \frac{1}{2}(\mathbf{e}_{0,f(0)}-\mathbf{e}_{0,\overline{f(0)}}+\mathbf{e}_{1,f(1)}-\mathbf{e}_{1,\overline{f(1)}})\\
&= \frac{1}{\sqrt{2}}\mathbf{e}_0\otimes\left[\frac{1}{\sqrt{2}}(\mathbf{e}_{f(0)}-\mathbf{e}_{\overline{f(0)}})\right]+\frac{1}{\sqrt{2}}\mathbf{e}_1\otimes\left[\frac{1}{\sqrt{2}}(\mathbf{e}_{f(1)}-\mathbf{e}_{\overline{f(1)}})\right]\\
&= \begin{cases}\frac{1}{\sqrt{2}}\mathbf{e}_0\otimes\mathbf{x}_1+\frac{1}{\sqrt{2}}\mathbf{e}_1\otimes\mathbf{x}_1 & \text{si } f=f_0\\ \frac{1}{\sqrt{2}}\mathbf{e}_0\otimes\mathbf{x}_1-\frac{1}{\sqrt{2}}\mathbf{e}_1\otimes\mathbf{x}_1 & \text{si } f=f_1\\ -\frac{1}{\sqrt{2}}\mathbf{e}_0\otimes\mathbf{x}_1+\frac{1}{\sqrt{2}}\mathbf{e}_1\otimes\mathbf{x}_1 & \text{si } f=f_2\\ -\frac{1}{\sqrt{2}}\mathbf{e}_0\otimes\mathbf{x}_1-\frac{1}{\sqrt{2}}\mathbf{e}_1\otimes\mathbf{x}_1 & \text{si } f=f_3\end{cases}\\
&= \begin{cases}\mathbf{x}_0\otimes\mathbf{x}_1 & \text{si } f=f_0\\ \mathbf{x}_1\otimes\mathbf{x}_1 & \text{si } f=f_1\\ -\mathbf{x}_1\otimes\mathbf{x}_1 & \text{si } f=f_2\\ -\mathbf{x}_0\otimes\mathbf{x}_1 & \text{si } f=f_3\end{cases}
\end{aligned}$$

En consecuencia,

$$H_2U_fH_2(\mathbf{e}_0\otimes\mathbf{e}_1)=H_2U_f(\mathbf{x}_0\otimes\mathbf{x}_1) \quad = \quad \begin{cases} H\mathbf{x}_0\otimes H\mathbf{x}_1 & \text{si } f=f_0\\ H\mathbf{x}_1\otimes H\mathbf{x}_1 & \text{si } f=f_1\\ -H\mathbf{x}_1\otimes H\mathbf{x}_1 & \text{si } f=f_2\\ -H\mathbf{x}_0\otimes H\mathbf{x}_1 & \text{si } f=f_3 \end{cases}$$

$$= \quad \begin{cases} \mathbf{e}_0\otimes\mathbf{e}_1 & \text{si } f=f_0\\ \mathbf{e}_1\otimes\mathbf{e}_1 & \text{si } f=f_1\\ -\mathbf{e}_1\otimes\mathbf{e}_1 & \text{si } f=f_2\\ -\mathbf{e}_0\otimes\mathbf{e}_1 & \text{si } f=f_3 \end{cases}$$

vale decir, al aplicar el algoritmo cuántico $H_2U_fH_2$ (nótese que utilizamos notación algebraica: los operadores se aplican de derecha a izquierda), partiendo del vector básico $\mathbf{e}_0\otimes\mathbf{e}_1$ se obtiene un vector de la forma $\varepsilon\mathbf{e}_S\otimes\mathbf{e}_1$ donde $\varepsilon\in\{-1,1\}$ es un signo y S es una señal que indica si acaso f es o no equilibrada. En otras palabras, la respuesta S coincide con $f(0)\oplus f(1)$, donde $\oplus$ es la *disyunción excluyente*, XOR. La auscultación del valor S se realiza siguiendo el postulado de medición, y su valor está apareciendo leyendo sólo el primer qubit. Al efectuar la medición se elige al vector básico $\mathbf{e}_S\otimes\mathbf{e}_1$ con probabilidad $\varepsilon^2=1$.

3 Algoritmo para el cálculo de la Transformada Discreta de Fourier

Sea $f:[\![0,n-1]\!]\to\mathbb{C}$ una función. La *transformada discreta de Fourier* de f es la función $\hat{f}:[\![0,n-1]\!]\to\mathbb{C}$ tal que para cada $j\in[\![0,n-1]\!]$: $\hat{f}(j)=\frac{1}{\sqrt{n}}\sum_{k=0}^{n-1}\exp\left(\frac{2\pi ijk}{n}\right)f(k)$. Aquí i es la raíz cuadrada de -1.

En $\mathbb{C}^n$ consideremos la base canónica formada por los vectores $\mathbf{e}_j=(\delta_{ij})_{i=0}^{n-1}$, $j=0,\ldots,n-1$. Para cada vector $\mathbf{f}=\sum_{j=0}^{n-1}f(j)\mathbf{e}_j\in\mathbb{C}^n$, su *transformada discreta de Fourier* es $\mathrm{TDF}(\mathbf{f})=\hat{\mathbf{f}}=\sum_{j=0}^{n-1}\hat{f}(j)\mathbf{e}_j\in\mathbb{C}^n$. Es claro que TDF es una transformación lineal y, respecto a la base canónica de $\mathbb{C}^n$, se representa por la matriz $\mathrm{TDF}=\frac{1}{\sqrt{n}}\left(\exp\left(\frac{2\pi ijk}{n}\right)\right)_{jk}$, la cual es en efecto unitaria, de hecho la matriz hermitiana de TDF, TDF^H, tiene la misma estructura que TDF salvo que los exponentes en cada entrada tienen el signo "$-$".

En particular, se tiene

$$\forall j\in[\![0,n-1]\!]:\ \mathrm{TDF}(\mathbf{e}_j)=\frac{1}{\sqrt{n}}\sum_{k=0}^{n-1}\exp\left(\frac{2\pi ijk}{n}\right)\mathbf{e}_k. \tag{6}$$

y, por supuesto,

$$\mathrm{TDF}(\mathbf{f})=\sum_{j=0}^{n-1}f(j)\,\mathrm{TDF}(\mathbf{e}_j). \tag{7}$$

Ahora, supongamos que $n=2^\nu$ es una potencia de 2. En este caso, la TDF puede calcularse mediante el procedimiento de la llamada *transformada rápida de Fourier* TRF (o si se quiere, FFT por sus siglas en inglés: *Fast Fourier Transform*). Este procedimiento es de complejidad en tiempo $O(\nu 2^\nu)=O(n\log n)$. Mas utilizando el paralelismo inherente a la computación cuántica, se le calculará aquí en un tiempo $O(\nu)$.

Observamos, por un lado, que $\mathbb{H}_\nu=\mathbb{C}^n$, y por otro lado, de la ecuación (6), que para los primeros valores de ν se tiene:

$\nu=1$

$$\begin{aligned}\mathrm{TDF}(\mathbf{e}_0) &= \frac{1}{\sqrt{2}}(\mathbf{e}_0+\mathbf{e}_1)\\ \mathrm{TDF}(\mathbf{e}_1) &= \frac{1}{\sqrt{2}}(\mathbf{e}_0-\mathbf{e}_1)\end{aligned}$$

$\nu=2$

$$\mathrm{TDF}(\mathbf{e}_{00}) \quad = \quad \frac{1}{\sqrt{2}}(\mathbf{e}_0+\mathbf{e}_1)\otimes\frac{1}{\sqrt{2}}(\mathbf{e}_0+\mathbf{e}_1)$$

$$\begin{aligned}
\mathrm{TDF}(\mathbf{e}_{01}) &= \frac{1}{\sqrt{2}}(\mathbf{e}_0 - \mathbf{e}_1) \otimes \frac{1}{\sqrt{2}}(\mathbf{e}_0 + i\mathbf{e}_1) \\
\mathrm{TDF}(\mathbf{e}_{10}) &= \frac{1}{\sqrt{2}}(\mathbf{e}_0 + \mathbf{e}_1) \otimes \frac{1}{\sqrt{2}}(\mathbf{e}_0 - \mathbf{e}_1) \\
\mathrm{TDF}(\mathbf{e}_{11}) &= \frac{1}{\sqrt{2}}(\mathbf{e}_0 - \mathbf{e}_1) \otimes \frac{1}{\sqrt{2}}(\mathbf{e}_0 - i\mathbf{e}_1)
\end{aligned}$$

De manera general, a cada índice $j \in [\![0, 2^\nu - 1]\!]$ lo identificaremos con la palabra $\boldsymbol{\varepsilon}_j = \varepsilon_{j,\nu-1} \cdots \varepsilon_{j,1}\varepsilon_{j,0}$ que lo representa en base 2. Así, el vector básico $\mathbf{e}_{\boldsymbol{\varepsilon}_j} \in \mathbb{H}_\nu$ es el producto tensorial de los vectores básicos $\mathbf{e}_{\varepsilon_{j,k}} \in \mathbb{H}_1$. Tendremos entonces, en $\mathbb{H}_\nu$, que para cada $j = 0, \ldots, 2^\nu - 1$:

$$\begin{aligned}
\mathrm{TDF}(\mathbf{e}_{\boldsymbol{\varepsilon}_j}) &= \bigotimes_{k=0}^{\nu-1} \frac{1}{\sqrt{2}} \left(\mathbf{e}_0 + \exp\left(\frac{\pi i j}{2^k}\right) \mathbf{e}_1 \right) \\
&= \tfrac{1}{\sqrt{2}}\left(\mathbf{e}_0 + \exp\left(\tfrac{\pi i j}{2^0}\right)\mathbf{e}_1\right) \otimes \tfrac{1}{\sqrt{2}}\left(\mathbf{e}_0 + \exp\left(\tfrac{\pi i j}{2^1}\right)\mathbf{e}_1\right) \otimes \cdots \otimes \tfrac{1}{\sqrt{2}}\left(\mathbf{e}_0 + \exp\left(\tfrac{\pi i j}{2^{\nu-1}}\right)\mathbf{e}_1\right) \qquad (8)
\end{aligned}$$

La forma de los factores en este producto tensorial sugiere considerar los operadores $Q_k : \mathbb{H}_1 \to \mathbb{H}_1$ con representación, respecto a la base canónica, dada mediante la matriz unitaria $Q_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\frac{\pi i}{2^k}\right) \end{bmatrix}$. De hecho, como en (8) la potencia en la exponenciación va cambiando, podemos considerar más bien un correspondiente operador "controlado": $Q^c_{kj} = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\pi i \frac{j}{2^k}\right) \end{bmatrix}$. Así, por ejemplo, si $j = 1$ entonces $Q^c_{k1} = Q_k$ en tanto que si $j = 0$ entonces $Q^c_{k0} = I$ coincide con la identidad.

Observamos además que para $\mathbf{x}_0 = \frac{1}{\sqrt{2}}(\mathbf{e}_0 + \mathbf{e}_1) = H(\mathbf{e}_0)$ se tiene $Q^c_{kj}(\mathbf{x}_0) = \frac{1}{\sqrt{2}}\left(\mathbf{e}_0 + \exp\left(\pi i \frac{j}{2^k}\right)\mathbf{e}_1\right)$.

Ahora, a cada $j \in [\![0, 2^\nu - 1]\!]$ representémoslo en base-2 mediante la palabra $\boldsymbol{\varepsilon}_j$. Se tiene que para cada $\ell \in [\![0, \nu - 1]\!]$, $\frac{\varepsilon_{j,\ell} 2^\ell}{2^k} = \frac{\varepsilon_{j,\ell}}{2^{k-\ell}}$. Por tanto, $\exp\left(\pi i \frac{j}{2^k}\right) = \exp\left(\pi i \frac{\sum_{\ell=0}^{\nu-1} \varepsilon_{j,\ell} 2^\ell}{2^k}\right) = \prod_{\ell=0}^{\nu-1} \exp\left(\pi i \frac{\varepsilon_{j,\ell}}{2^{k-\ell}}\right)$ y en consecuencia, $Q^c_{kj} = Q^c_{k-\nu+1,\varepsilon_{j,\nu-1}} \circ \cdots \circ Q^c_{k-1,\varepsilon_{j,1}} \circ Q^c_{k,\varepsilon_{j,0}}$. Como k ha de variar entre 0 y $\nu - 1$ vemos que se ha de disponer de $2(2\nu - 1)$ compuertas de la forma $Q^c_{\kappa\varepsilon}$, $\kappa \in [\![-(\nu-1), \nu-1]\!]$, $\varepsilon \in \{0, 1\}$.

Observamos también que si $j < 2^{\nu_1}$, con $\nu_1 \leq \nu$ entonces todos los dígitos, en su representación binaria, con índices entre $\nu_1 - 1$ y $\nu - 1$ son 0, y por tanto las correspondientes compuertas controladas actuarán como la identidad. Definamos pues para cada $(j, k) \in [\![0, 2^\nu - 1]\!] \times [\![0, \nu - 1]\!]$,

$$P_{jk} = Q^c_{k-\nu_1+1,\varepsilon_{j,\nu_1-1}} \circ \cdots \circ Q^c_{k-1,\varepsilon_{j,1}} \circ Q^c_{k,\varepsilon_{j,0}}, \qquad (9)$$

donde $\nu_1 = \lceil \log_2 j \rceil + 1$. Tenemos pues: $P_{jk}(\mathbf{x}_0) = \frac{1}{\sqrt{2}}\left(\mathbf{e}_0 + \exp\left(\pi i \frac{j}{2^k}\right)\mathbf{e}_1\right)$.

Fijo $j \in [\![0, 2^\nu - 1]\!]$ para $k = 0, \ldots, \nu - 1$ los términos $P_{jk}(\mathbf{x}_0)$ van dando los de la derecha de la ec. (8) y éstos van apareciendo de izquierda a derecha según se les muestra ahí. Sin embargo, para cada $k \in [\![0, \nu - 1]\!]$ observamos que en la definición (9) se está utilizando una notación algebraica, es decir, los operadores $Q^c_{k-\ell,\varepsilon_{j,\ell}}$ se aplican en orden inverso: de derecha a izquierda. De hecho, haciendo un poco más explícita la definición (9) se tiene:

$$\begin{aligned}
Q^c_{0,\varepsilon_{j,0}}(\mathbf{x}_0) &= P_{j0}(\mathbf{x}_0) \\
Q^c_{1,\varepsilon_{j,0}} \circ Q^c_{0,\varepsilon_{j,1}}(\mathbf{x}_0) &= P_{j1}(\mathbf{x}_0) \\
Q^c_{2,\varepsilon_{j,0}} \circ Q^c_{1,\varepsilon_{j,1}} \circ Q^c_{0,\varepsilon_{j,2}}(\mathbf{x}_0) &= P_{j2}(\mathbf{x}_0) \\
&\vdots \quad \vdots \\
Q^c_{\nu-1,\varepsilon_{j,0}} \circ \cdots \circ Q^c_{2,\varepsilon_{j,\nu-3}} \circ Q^c_{1,\varepsilon_{j,\nu-2}} \circ Q^c_{0,\varepsilon_{j,\nu-1}}(\mathbf{x}_0) &= P_{j,\nu-1}(\mathbf{x}_0)
\end{aligned}$$

es decir, para cada $k \in [\![0, \nu-1]\!]$ se han de aplicar consecutivamente las compuertas $Q^c_{\ell,\varepsilon_{j,k-\ell}}$, con $\ell = 0, \ldots, k$, la cual selecciona a los dígitos en la representación binaria de j yendo del más significativo hacia el menos significativo.

Así pues, será necesario utilizar los operadores reverso para intercambiar el orden de los bits de cada índice $j \in [\![0, 2^\nu - 1]\!]$.

Ahora bien, cada bit ε se representa por el vector básico $\mathbf{e}_\varepsilon$. Así que cada operador controlado $Q^c_{k,\varepsilon}$, cuyo dominio es $\mathbb{H}_1$ puede identificarse con el operador $\mathbf{x} \mapsto Q^{c2}(\mathbf{x}, \mathbf{e}_\varepsilon)$ donde

$$Q^{c2} = (I \otimes Q_k) \circ C \circ (I \otimes Q_k^H) \circ C \circ (Q_k \otimes I). \tag{10}$$

El algoritmo para calcular la transformada de Fourier es entonces el siguiente:

Entrada. $n = 2^\nu$, $\mathbf{f} \in \mathbb{C}^n = \mathbb{H}_\nu$.

Salida. $\hat{\mathbf{f}} = \text{TDF}(\mathbf{f}) \in \mathbb{H}_\nu$.

Procedimiento $\texttt{TDF}(n, \mathbf{f})$

1. Sea $\mathbf{x}_0 := H(\mathbf{e}_0)$.
2. Para cada $j \in [\![0, 2^\nu - 1]\!]$, o equivalentemente, para cada $(\varepsilon_{j,\nu-1} \cdots \varepsilon_{j,1}\varepsilon_{j,0}) \in \{0,1\}^\nu$, hágase en paralelo:
   (a) Para cada $k \in [\![0, \nu - 1]\!]$ hágase en paralelo:
      i. Sea $\boldsymbol{\delta} := R_k\left(\boldsymbol{\varepsilon}_j|_k\right)$ el reverso de la cadena formada por los $(k+1)$ bits menos significativos (véase la ec. (5)).
      ii. Sea $\mathbf{y}_{jk} := \mathbf{x}_0$.
      iii. Para $\ell = 0$ hasta k hágase $\{\ \mathbf{y}_{jk} := Q^{c2}(\mathbf{y}_{jk}, \mathbf{e}_{\delta_{j,\ell}})$ (véase la ec. (10)) $\}$
   (b) Sea $\mathbf{y}_j := \mathbf{y}_{j0} \otimes \cdots \otimes \mathbf{y}_{j,\nu-1}$ (véase la ec. (8)).
3. Dése como resultado $\hat{\mathbf{f}} = \sum_{j=0}^{2^\nu-1} f_j \mathbf{y}_j$.

El cálculo de la transformada inversa discreta de Fourier, $\texttt{TIDF}(n, \mathbf{f})$ se hace de manera similar cambiando la matriz Q_k por su hermitiana Q_k^H que sólo involucra el cambio de signo en la potencia de su elemento $(2, 2)$.

4 Algoritmo de Shor

Este algoritmo es de tipo cuántico y tiene el propósito de factorizar a un número entero dado n como el producto de dos enteros menores, si esto es posible, o bien indicar que n es primo, en otro caso.

4.1 Pequeño recordatorio de Teoría de Números

Dados dos enteros n, m, su *máximo común divisor* es $d = \text{mcd}(n, m)$ tal que d divide a ambos n y m, es decir es un divisor común de n y m, y cualquier otro divisor común divide a d también. Se puede ver que d es el menor entero positivo que se puede escribir como una combinación lineal de n y m con coeficientes enteros. El *Algoritmo de Euclides* calcula, para dos enteros n y m dados, $d = \text{mcd}(n, m)$ y lo expresa de la forma $d = an + bm$, con $a, b \in \mathbb{Z}$.

Los enteros n y m son *primos relativos* si $\text{mcd}(n, m) = 1$, es decir, si no poseen un divisor común que no sea trivial. Sea $\Phi(n) = \{m \in [\![1, n]\!] | \, \text{mcd}(n, m) = 1\}$ el conjunto de enteros positivos primos relativos con n, menores que n. Se tiene que el número de elementos en $\Phi(n)$ es el valor de la *función de Euler* ϕ en n. Con la multiplicación módulo n, $\Phi(n)$ es un grupo de orden $\phi(n)$. Así pues, si $m \in \Phi(n)$ entonces $m^{\phi(n)} = 1 \bmod n$. Por tanto, para cada entero $m \in \Phi(n)$ existe un mínimo elemento r, divisor de $\phi(n)$, tal que $m^r = 1 \bmod n$. Tal r se dice ser el *orden* de m en el grupo multiplicativo de residuos módulo n.

Sea n un entero para el cual se ha de buscar un factor entero no trivial. Elijamos un entero m tal que $1 < m < n$. Si $\text{mcd}(n, m) = d > 1$, entonces d es un factor no-trivial de n. En otro caso, $\text{mcd}(n, m) = 1$, y se tiene que m quedará en el grupo multiplicativo de residuos de n, i.e. $m \in \Phi(n)$. Si acaso m tuviera ahí un orden par r, entonces $k = m^{\frac{r}{2}}$ es tal que $k^2 = 1 \bmod n$. En consecuencia, $(k-1)(k+1) = 0 \bmod n$, es decir n divide al producto de dos números menores que él. Por tanto, los factores primos de n han de aparecer como factores de esos números. Así pues al calcular $\text{mcd}(n, k-1)$ y $\text{mcd}(n, k+1)$ obtendremos factores no-triviales de n.

Un primer problema en este procedimiento de decisión consiste entonces en encontrar un elemento de orden par en el grupo multiplicativo de residuos módulo n. Si se elige m al azar, la probabilidad de que m

sea de orden par es $1-\frac{1}{2^\ell}$ donde ℓ es el número de factores primos en n. Así pues, la probabilidad de que al cabo de k intentos no se haya localizado un tal m es $2^{-k\ell}$ y obviamente esto tiende a cero muy rápidamente al incrementar k. Así pues, bien vale la pena repetir pruebas arbitrarias de selección de un elemento (impar) menor que n para localizar uno de orden par en el grupo multiplicativo de residuos módulo n.

Sin embargo, desde el punto de vista computacional, el mayor problema que presenta el algoritmo descrito radica en el cálculo del orden del elemento actual m en $\Phi(n)$: el número de potencias de m a calcular es del orden de $\phi(n)$ que a su vez es de orden n.

Sea $\nu = \lceil \log_2 n \rceil$ el número de bits necesarios para escribir a n, es decir, sea ν el *tamaño* de n. Resulta claro que $O(n) = O(2^\nu)$ lo cual indica que el procedimiento anterior es de complejidad exponencial respecto al tamaño de la entrada. El algoritmo de Shor se fundamenta en un procedimiento polinomial en ν para realizar la tarea de calcular el orden de un elemento.

4.2 Algoritmo cuántico para el cálculo de órdenes

Supongamos dado $n \in \mathbb{N}$. Sea $\nu = \lceil \log_2 n \rceil$ su tamaño. Sea κ tal que $n^2 \leq 2^\kappa < 2n^2$, es decir, $\kappa = \lceil 2 \log_2 n \rceil$. Se considerará $\kappa + \nu$ qubits y se trabajará en el espacio $\mathbb{H}_{\kappa+\nu} = \mathbb{H}_\kappa \otimes \mathbb{H}_\nu$, de dimensión $2^{\kappa+\nu} = 2^\kappa \cdot 2^\nu$.

Para cada $m \in \Phi(n)$ definimos un operador lineal unitario $V_m : \mathbb{H}_{\kappa+\nu} \to \mathbb{H}_{\kappa+\nu}$ haciéndolo actuar en los vectores básicos como

$$V_m : \mathbf{e}_{\boldsymbol{\varepsilon}_j} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_i} \mapsto \mathbf{e}_{\boldsymbol{\varepsilon}_j} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{f(i,j,m)}} \tag{11}$$

donde $f(i,j,m) = (j + m^i) \bmod n$.

4.2.1 Elementos con orden potencia de 2

Supongamos dado $m \in \Phi(n)$ y que éste es tal que su orden r es una potencia de 2.

Sea $P_1 = H^{\otimes\kappa} \otimes I^{\otimes\nu}$ donde $H, I : \mathbb{H}_1 \to \mathbb{H}_1$ son los operadores de Hadamard e identidad respectivamente. Por la ec. (4) se tiene $P_1(\mathbf{e_0} \otimes \mathbf{e_0}) = \frac{1}{\sqrt{2^\kappa}} \sum_{\boldsymbol{\varepsilon} \in \{0,1\}^\kappa} \mathbf{e}_{\boldsymbol{\varepsilon}} \otimes \mathbf{e_0}$. Escribamos $\mathbf{s}_1 = P_1(\mathbf{e_0} \otimes \mathbf{e_0})$. Ahora, aplicando el operador V_m, resulta $V_m(\mathbf{s}_1) = \frac{1}{\sqrt{2^\kappa}} \sum_{i=0}^{2^\kappa - 1} \mathbf{e}_{\boldsymbol{\varepsilon}_i} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{f(i,0,m)}}$. Escribamos $\mathbf{s}_2 = V_m(\mathbf{s}_1)$.

Ya que $f(i,0,m) = m^i \bmod n$, f es una función periódica de período r respecto a i. Sea $J_j = \{i | 0 \leq i \leq 2^\kappa - 1 : i = j \bmod r\}$ la clase de índices que dejan residuo j al dividírseles entre r. Claramente $[\![0, 2^\kappa - 1]\!] = \bigcup_{j=0}^{r-1} J_j$, y la cardinalidad de cada conjunto J_j es $s = \frac{2^\kappa}{r}$, el cual número, para este caso, es entero. Por tanto, es posible reescribir

$$\mathbf{s}_2 = \frac{1}{\sqrt{2^\kappa}} \sum_{j=0}^{r-1} \left(\sum_{i \in J_j} \mathbf{e}_{\boldsymbol{\varepsilon}_i} \right) \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^j}} . \tag{12}$$

Si aquí se aplica el postulado de medición, entonces se elegirá a un vector de la forma $\mathbf{e}_{\boldsymbol{\varepsilon}_i} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}}$, $i \in J_{j_0}$, para una potencia fija $j_0 \leq r$, con probabilidad $\frac{r}{2^\kappa}$. El estado correspondiente a esta situación es de la forma

$$\mathbf{s}_3 = \sum_{i=0}^{2^\kappa - 1} g(i) \mathbf{e}_{\boldsymbol{\varepsilon}_i} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}} . \tag{13}$$

donde $g : i \mapsto \begin{cases} \sqrt{\frac{r}{2^\kappa}} & \text{si } i \in J_{j_0} \\ 0 & \text{si } i \notin J_{j_0} \end{cases}$

La función g también es periódica, de período r. Ahora, se tiene que la transformada de Fourier de g, $\hat{g}$ será también periódica pero de período proporcional a $\frac{1}{r}$.

Calculemos la transformada inversa discreta de Fourier de $\mathbf{s}_3$:

$$\check{\mathbf{s}_3} = \mathrm{TDF}^H(\mathbf{s}_3) = \sqrt{\frac{r}{2^\kappa}} \sum_{k=0}^{s-1} \left(\frac{1}{\sqrt{2^\kappa}} \sum_{\ell=0}^{2^\kappa - 1} \exp\left(-\frac{2\pi i \ell}{2^\kappa} (kr + j_0) \right) \mathbf{e}_\ell \right) \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}} ,$$

y al intercambiar el orden de las sumatorias se obtiene:

$$\mathbf{s}_4 = \check{\mathbf{s}_3} = \frac{1}{\sqrt{r}} \left(\sum_{\ell=0}^{2^\kappa - 1} \left(\frac{1}{s} \sum_{k=0}^{s-1} \exp\left(-\frac{2\pi i \ell k}{s} \right) \right) \exp\left(-\frac{2\pi i \ell j_0}{2^\kappa} \right) \mathbf{e}_\ell \right) \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}} . \tag{14}$$

Ya que $\exp\left(-\frac{2\pi i\ell}{s}\right)$ es una raíz s-ésima de la unidad, se tiene $\frac{1}{s}\sum_{k=0}^{s-1}\exp\left(-\frac{2\pi i\ell k}{s}\right)$ será 1 o 0 en función de que ℓ sea o no un múltiplo entero de s, es decir un número de la forma $\ell = ts$, con $t = 0, \ldots, r-1$. Aquí es esencial el hecho de que s es entero. Así pues, de (14),

$$\mathbf{s}_4 = \frac{1}{\sqrt{r}}\left(\sum_{t=0}^{r-1}\exp\left(-\frac{2\pi itj_0}{r}\right)\mathbf{e}_{\frac{2^\kappa t}{r}}\right)\otimes\mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}}. \tag{15}$$

Las relaciones (13) y (15), que expresan a $\mathbf{s}_3$ y $\mathbf{s}_4 = \check{\mathbf{s}}_3$, involucran ambas al orden r. Pero hay una diferencia esencial entre ellas: En (13) los índices i, en el primer qubit, correspondientes a coeficientes no-nulos dependen de la potencia "aleatoria" j_0, en tanto que en (15) no dependen de ésa, e involucran sin embargo, a r.

Si ahora se aplica el postulado de medición se obtendrá un valor de la forma $\frac{2^\kappa t_0}{r}$, con $t_0 \in [\![0, r-1]\!]$, cada uno con probabilidad r^{-1}. Si $t_0 = 0$, entonces no es posible obtener ninguna información acerca de r y se ha de repetir el procedimiento otra vez. En otro caso, al dividir entre 2^κ se tiene el valor racional $\frac{r_0}{r_1} = \frac{t_0}{r}$. Se conoce a r_0 y r_1 mas aún no se conoce t_0 ni r. Sin embargo, necesariamente r_1 divide a r. Así pues, se puede aplicar de nuevo el algoritmo cuántico a partir de $m_1 = m^{r_1}$. Procediendo recursivamente se obtiene una factorización $r = r_1 r_2 \cdots r_p$ conteniendo a lo más $\log_2 r$ factores.

En resumen, el algoritmo para localizar divisores de órdenes de elementos es el siguiente:

Entrada. $n \in \mathbb{N}$, $m \in \Phi(n)$ de orden potencia de 2.

Salida. r tal que $r|o(m)$.

Procedimiento `DivisorOrdenPotencia2`(n, m)

1. Sea $\nu := \lceil \log_2 n \rceil$, $\kappa := 2\nu$.
2. Defínase $V_m : \mathbb{H}_{\kappa+\nu} \to \mathbb{H}_{\kappa+\nu}$ como en la ec. (11).
3. Sea $\mathbf{s}_1 := (H^{\otimes\kappa} \otimes I^{\otimes\nu})(\mathbf{e_0} \otimes \mathbf{e_0})$.
4. Sea $\mathbf{s}_2 := V_m(\mathbf{s}_1)$.
5. Sea $\mathbf{s}_3 := \sum_{i=0}^{2^\kappa-1} g(i)\mathbf{e}_{\boldsymbol{\varepsilon}_i} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}}$ el estado equivalente a "tomar una medición" en $\mathbf{s}_2$. Entonces g queda determinada como en la ec. (13).
6. Sea $\mathbf{s}_4 := \texttt{TIDF}(2^\kappa, \mathbf{s}_3)$ la transformada inversa discreta de Fourier de $\mathbf{s}_3$.
7. Sea $\mathbf{e}_{\boldsymbol{\varepsilon}_k} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}}$ una medición de $\mathbf{s}_4$.
8. Si $k == 0$ repítase desde el paso 3. En otro caso sea $\frac{r_0}{r_1} = \frac{k}{2^\kappa}$ y dése como resultado r_1.

El algoritmo para calcular órdenes de elementos es el siguiente:

Entrada. $n \in \mathbb{N}$, $m \in \Phi(n)$ de orden potencia de 2.

Salida. r tal que $r = o(m)$.

Procedimiento `OrdenPotencia2`(n, m)

1. Sean inicialmente $r := 1$ y $m_1 := m$.
2. Repítase
   (a) sea $r_1 := \texttt{DivisorOrdenPotencia2}(n, m_1)$;
   (b) actualícese $r := r \cdot r_1$;
   (c) actualícese $m_1 := m_1^{r_1} \bmod n$.

   hasta tener $r_1 == 1$.
3. Dése como resultado r.

4.2.2 Elementos con orden arbitrario

Dejemos de suponer que el orden r de m sea una potencia de 2 en $\Phi(n)$. Siguiendo la misma línea que en el caso anterior, sea V_m definido como en la ec. (11). Sea $\mathbf{s}_1 = (H^{\otimes\kappa} \otimes I^{\otimes\nu})(\mathbf{e_0} \otimes \mathbf{e_0})$ y $\mathbf{s}_2 = V_m(\mathbf{s}_1)$. Reagrupando los términos según se hizo en la ec. (12) se puede escribir

$$\mathbf{s}_2 = \frac{1}{\sqrt{2^\kappa}} \sum_{j=0}^{r-1} \left(\sum_{i \in J_j} \mathbf{e}_{\boldsymbol{\varepsilon}_i} \right) \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^j}} . \tag{16}$$

donde los conjuntos J_j son clases de equivalencia, congruentes con j, módulo r, pero ahora no son de la misma cardinalidad. Si $u = 2^\kappa \bmod r$ y $s = (2^\kappa - u)/r$ entonces u clases tendrán $s+1$ elementos y las restantes tendrán s elementos. Definamos $s_j = s+1$ para $j = 1, \ldots, u$ y $s_j = s$ para $j = u+1, \ldots, r-1, 0$. Entonces el estado que representa el tomar una medición, como en la ec. (13), es, para algún $j_0 \in [\![0, r-1]\!]$:

$$\mathbf{s}_3 = \sum_{i=0}^{2^\kappa - 1} g(i) \mathbf{e}_{\boldsymbol{\varepsilon}_i} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}} . \tag{17}$$

donde $g : i \mapsto \begin{cases} \frac{1}{\sqrt{s_{j_0}}} & \text{si } i \in J_{j_0} \\ 0 & \text{si } i \notin J_{j_0} \end{cases}$

Calculando la transformada inversa discreta de Fourier y reagrupando términos, como en la ec. (14), se obtiene

$$\mathbf{s}_4 = \check{\mathbf{s}}_3 = \frac{1}{\sqrt{2^\kappa}} \left(\sum_{\ell=0}^{2^\kappa - 1} \left(\frac{1}{\sqrt{s_{j_0}}} \sum_{k=0}^{s_{j_0}-1} \exp\left(-\frac{2\pi i \ell k r}{2^\kappa} \right) \right) \exp\left(-\frac{2\pi i \ell j_0}{2^\kappa} \right) \mathbf{e}_\ell \right) \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}} . \tag{18}$$

pero en este caso el coeficiente que involucra a la suma interior nunca se anula (como r no necesariamente divide a 2^κ, aquí no se está sumando un "juego completo" de raices s_{j_0}-ésimas de la unidad). Al tomar una medición del primer qubit, la probabilidad de que se elija a $\mathbf{e}_\ell \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}}$ es entonces

$$P(\ell) = \frac{1}{\sqrt{2^\kappa s_{j_0}}} \left| \sum_{k=0}^{s_{j_0}-1} \exp\left(-\frac{2\pi i \ell k r}{2^\kappa} \right) \right|^2$$

y los máximos de esos valores corresponden a enteros $\ell = \texttt{EnteroMásPróximo}\left(\frac{k2^\kappa}{r}\right)$, $k = 0, \ldots, r-1$. Supongamos que tras una medición se haya elegido $\mathbf{e}_{\ell_k} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}}$, con $\ell_k = \texttt{EnteroMásPróximo}\left(\frac{k2^\kappa}{r}\right)$. Entonces, al dividir ese índice entre 2^κ se obtiene $\frac{\ell_k}{2^\kappa} \sim \frac{k}{r}$, y de aquí se quiere conocer r. Para esto hay que recordar la noción de *fracciones continuadas*.

Si $\frac{p}{q}$ es un número racional no-negativo, su *fracción continuada* es

$$\frac{p}{q} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ldots + \cfrac{1}{a_v}}} = [a_0, a_1, \ldots, a_v] \tag{19}$$

donde $a_0, a_1, \ldots, a_v$ son enteros no-negativos. Para cada $w \leq v$, la fracción continuada $[a_0, a_1, \ldots, a_w]$ se dice ser el *w-ésimo convergente* de $\frac{p}{q}$, y, en efecto, cada convergente es una aproximación racional a $\frac{p}{q}$. El algoritmo para calcular fracciones continuadas es directo:

Entrada. $\frac{p}{q} \in \mathbb{Q}$.

Salida. $[a_0, a_1, \ldots, a_v]$: fracción continuada que representa a $\frac{p}{q} \in \mathbb{Q}$.

Procedimiento `FracciónContinuada`$(\frac{p}{q})$

1. Sean inicialmente $lst := []$ (la lista vacía) y $xact := \frac{p}{q}$.
2. Mientras el denominador de $xact$ sea mayor que 1 hágase

(a) sea $i := \texttt{ParteEntera}(xact)$;
(b) escríbase $\frac{p_1}{q_1} = xact$;
(c) actualícese $xact := \frac{q_1}{p_1 - iq_1}$;
(d) actualícese $lst := lst * [i]$.

3. Actualícese $lst := lst * [xact]$.
4. Dése como resultado *lst*.

Así pues, luego de haber realizado una medición y haber obtenido el valor $\frac{\ell_k}{2^\kappa} < 1$, se calcula su fracción continuada $[a_0, a_1, \ldots, a_v]$ $(a_0 = 0)$ y los correspondientes convergentes $[c_0, c_1, \ldots, c_v]$ (también $c_0 = 0$), y entre éstos se selecciona a aquellos cuyos denominadores r_j sean menores que n, los cuales han de ser divisores del orden r de m.

En resumen, esta vez el algoritmo para localizar divisores de órdenes de elementos es el siguiente:

Entrada. $n \in \mathbb{N}$, $m \in \Phi(n)$.

Salida. r tal que $r|o(m)$.

Procedimiento $\texttt{DivisorOrden}(n, m)$

1. Sea $\nu := \lceil \log_2 n \rceil$, $\kappa = \lceil 2 \log_2 n \rceil$.
2. Defínase $V_m : \mathbb{H}_{\kappa+\nu} \to \mathbb{H}_{\kappa+\nu}$ como en la ec. (11).
3. Sea $\mathbf{s}_1 := (H^{\otimes\kappa} \otimes I^{\otimes\nu})(\mathbf{e_0} \otimes \mathbf{e_0})$.
4. Sea $\mathbf{s}_2 := V_m(\mathbf{s}_1)$.
5. Sea $\mathbf{s}_3 := \sum_{i=0}^{2^\kappa - 1} g(i)\mathbf{e}_{\boldsymbol{\varepsilon}_i} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}}$ el estado equivalente a "tomar una medición" en $\mathbf{s}_2$. Entonces g queda determinada como en la ec. (17).
6. Sea $\mathbf{s}_4 := \texttt{TIDF}(2^\kappa, \mathbf{s}_3)$ la transformada inversa discreta de Fourier de $\mathbf{s}_3$.
7. Sea $\mathbf{e}_{\boldsymbol{\varepsilon}_{\ell_k}} \otimes \mathbf{e}_{\boldsymbol{\varepsilon}_{m^{j_0}}}$ una medición de $\mathbf{s}_4$.
8. Si $\ell_k == 0$ repítase desde el paso 3. En otro caso
   (a) sea $[a_0, a_1, \ldots, a_v] := \texttt{FracciónContinuada}\left(\frac{\ell_k}{2^\kappa}\right)$;
   (b) sea $[c_0, c_1, \ldots, c_v]$ la lista de convergentes; y
   (c) dése como resultado la lista de denominadores, de los convergentes, que sean menores que n.

Habiendo obtenido divisores de órdenes, se puede proceder a obtener los órdenes de manera similar a como se bosquejó en el procedimiento `OrdenPotencia2`, mas en este caso hay que llevar un recuento de las varias posibilidades de divisores que arroja el procedimiento `DivisorOrden` descrito arriba.

Referencias

[1] G. Brassard, I. Chuang, S. Lloyds, C. Monroe, Quantum computing, *Proc. Natl. Acad. Sci. USA*, Vol. 95, pp. 1103211033, September 1998

[2] D. Deutsch, R. Jozsa, Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London A*, 439, 553-558, (1992).

[3] C. Lavor, L.R.U. Manssur, R. Portugal, Shor's Algorithm for Factoring Large Integers, `arXiv:quant-ph/0303175v1`, Mar. 2003.

[4] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.