Inter*net* Electronic Journal
Nanociencia et Moletrónica

Octubre 2003, Vol. 1; N°1, págs. 35-44

Procesos Elementales en una Computadora Cuántica

C. Bautista
Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla
14 Sur y Av. San Claudio. Edif. 135 Puebla
Pue. 72570 México
e-mail: bautista@cs.buap.mx

recibido: Julio 14, 2003 revisado: Septirmbre 16, 2003 publicado: Octubre 16, 2003

Citation of the article:
C. Bautista, "Procesos Elementales en una Computadora Cuántica", Internet Electrón. J. Nanocs. Moletrón. 2003**,** 1(1), 35-44: http://www.revista-nanociencia.ece.buap.mx

Procesos Elementales en una Computadora Cuántica

C. Bautista
Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla
14 Sur y Av. San Claudio. Edif. 135 Puebla
Pue. 72570 México
e-mail: bautista@cs.buap.mx

recibido: Julio 14, 2003 revisado: Septiembre 16, 2003 publicado: Octubre 16, 2003

Internet Electron. J. Nanocs. Moletrón. 2003, 1(1), pags. 35-44

Resumen:
Se describen los algoritmos cuánticos más elementales usando formalismos puramente matemáticos

Palabras clave: computadoras cuánticas, algoritmos cuánticos, codificación superdensa, algoritmo de Deutsch-Jozsa, algoritmia.

1. Introducción

Una de las aportaciones de la computación cuántica [1] a la teoría de la computación es que el ``como" hacer cálculos depende de la teoría matemática usada. Debido a que las computadoras cuánticas hacen uso de partículas elementales, la física que las gobierna es la *mecánica cuántica*. Que las leyes de la mecánica cuántica son muy diferentes a las leyes de la mecánica clásica es un lugar común. Una de las diferencias más notables es el carácter probabilístico de la mecánica cuántica.

En consecuencia la computación clásica, que se diseña siguiendo las leyes de la mecánica clásica, es muy diferente a la computación cuántica. Se cree, aunque hasta la fecha aún no ha sido probado, que la computación cuántica es intrínsecamente más eficiente que su contraparte clásica. En parte la intención del presente trabajo es explicar la evidencia que apoya ésta afirmación.

2. Formalismo matemático

La computación clásica hace uso extensivo de funciones sobre conjuntos. Los datos de entrada se transforman en los de salida mediante funciones. Por ejemplo, de la función que copia fielmente datos: $x \to (x,x)$ llamada *morfismo diagonal* ó *``fan-out''*; así como también de la función dual, la que borra datos $(x,y) \to x$ llamada *morfismo proyección* ó *``fan-in''* Sin embargo, ninguno de éstos procedimientos son posibles en una computadora cuántica, debido a que sus procesos son de un carácter más restringido que los de una computadora clásica, pero a la vez son más ricos en estructura. Un uso inteligente de tal estructura ha permitido diseñar algunos algoritmos exponencialmente más eficientes que su contraparte clásica, como veremos en lo que sigue.

La computación cuántica usa matrices como forma de transformar la información de entrada, que es un vector, en otro vector. El conocimiento de álgebra lineal es esencial para entender los procesos cuánticos.

2.1 Estados. El bloque de información fundamental en la computación cuántica es el *qubit*. En la práctica se usan los qubits denotados $|0\rangle$ y $|1\rangle$. Más que preocuparnos por explicar lo que son, trataremos de explicar sus propiedades. Sólo diremos que $|0\rangle$, $|1\rangle$ forman una base ortonormal de un espacio vectorial complejo denotado C^2. Luego si $\langle -|-\rangle$ denota el producto interno de te C^2 tenemos que, abusando de la notación

$$\langle \alpha | \beta \rangle = \begin{cases} 1, & si \ \alpha = \beta \\ 0, & si \ \alpha \neq \beta \end{cases}$$

donde $\langle \alpha | \beta \rangle$ denota el producto interno de $|\alpha\rangle$ con $|\beta\rangle$. Debemos decir que también se considera como qubit a cualquier vector en con norma 1.

Así como en la computación clásica se pueden formar cadenas de bits (que pueden formar bytes, por ejemplo), en la computación cuántica se pueden formar cadenas de qubits para describir el estado de la computadora, pero entre ellos hay que poner el símbolo $\otimes$ de producto tensorial. Por ejemplo $|0\rangle \otimes |1\rangle \otimes |1\rangle \otimes |0\rangle$ es un estado. Para abreviar, denotamos con $|\alpha_1 \ldots \alpha_n\rangle$ al estado $|\alpha_1\rangle \otimes \cdots \otimes |\alpha_n\rangle$. La propiedad que distingue al tensor $\otimes$ es la de ser bilineal:

$$(\zeta_1|0\rangle + \zeta_2|1\rangle) \otimes (\chi_1|0\rangle + \chi_2|1\rangle) = \zeta_1\chi_1|00\rangle + \zeta_1\chi_2|01\rangle + \zeta_2\chi_1|10\rangle + \zeta_2\chi_2|11\rangle$$

para cualesquiera $\zeta_1, \zeta_2, \chi_1, \chi_2$ números complejos.

Otra propiedad de los productos tensoriales es que preservan el concepto de base de un espacio vectorial, esto es, productos tensoriales de bases es de nuevo una base. En particular, los productos

$$|\alpha_1 \cdots \alpha_n\rangle, \quad \alpha_i \in \{0,1\}, \quad i = 1, \ldots, n$$

forman una base ortonormal de C^2, que es un espacio vectorial complejo de dimensión 2^n, con respecto al producto interno

$$\langle \alpha_1 \ldots \alpha_n | \beta_1 \ldots \beta_n \rangle = \langle \alpha_1 | \beta_1 \rangle \cdots \langle \alpha_n | \beta_n \rangle$$

Tal base se llama *base del cálculo*.

3. Evolución

La clase de instrucciones que permiten cambiar el estado de una computadora cuántica son la formada por la matrices unitarias complejas así como las matrices de proyección ó más generalmente, matrices que satisfacen la condición de completitud. Específicamente, una matriz U de orden $n\times n$ se llama unitaria si satisface

$$U\, U^* = Id_n$$

donde U^* es la matriz transpuesta conjugada de U y Id_n es la matriz identidad de orden $n\times n$. Mientras que una colección finita $M_1,...,M_k$ de matrices de orden $n\times n$ se dice que satisfacen la condición de completitud si se cumple

$$M_1 M_1^* + \ldots + M_k M_k^* = Id_n$$

(ver [1]). Tales matrices se llaman de *medición*. Nótese que la condición de completitud contiene como caso particular, cuando $k=1$, a la condición unitaria.

Si tenemos la computadora cuántica en estado $|\varphi\rangle$, y U matriz unitaria, entonces podemos cambiar (evolucionar) la máquina al estado $|\psi\rangle$ que se obtiene al aplicar la matriz U a $|\varphi\rangle$: $|\psi\rangle = U|\varphi\rangle$. La forma en que actúan las matrices de medición $M_1, \ldots, M_k$ sobre un estado es más sutil. Si $|\varphi\rangle$ es el estado de la máquina antes de la medición, después es el estado llamado *colapsado*

$$|\psi\rangle = \frac{M_i|\psi\rangle}{\sqrt{\langle M_i|\psi\rangle | M_I|\psi\rangle\rangle}}$$

para algún $1 \le i \le k$. No podemos de antemano con toda certeza saber que M_i va a ser aplicado. Lo que se puede decir es una probabilidad *p* de obtener el estado colapsado:

$$p = \langle M|\varphi\rangle | M|\varphi\rangle\rangle .$$

La teoría de la algoritmia [2] nos enseña que la eficiencia de un algoritmo se mide en términos de las llamadas *operaciones elementales*. Tales operaciones son, en computación clásica, las comparaciones, las sumas, los productos, etc. Pero todas ellas se consideran elementales cuando actúan sobre un número finito y fijo de bits. Es en base a estas operaciones elementales que se construyen y se estudian algoritmos más complicados.

De manera análoga, en computación cuántica, la construcción de algoritmos se hace a partir de ciertas matrices unitarias que actúan sobre un número pequeño de qubits: uno, dos o tres [3]. La forma de construir matrices (compuertas) más complicadas es usando productos
tensoriales: si $\{A_1,\ldots,A_k\}$, $\{B_1,\ldots,B_s\}$ son matrices de medición entonces también lo son $A_i \otimes B_j$, *i=1,...,k, j=1,...,s*. Tales actúan sobre vectores de la siguiente forma

$$(A \otimes B)(|\varphi_1\rangle \otimes |\varphi_2\rangle) = A|\varphi\rangle \otimes B|\varphi_2\rangle .$$

El proceso (cerrado) de cálculo en una computadora cuántica es

Preparación $\mapsto$ Evolución $\mapsto$ Medición

Preparación se refiere a colocar el sistema en un estado fijo, usualmente en $|\varphi\rangle = |0\ldots0\rangle$. Evolución significa aplicar secuencialmente diferentes matrices unitarias al estado de preparación. Mientras que medir es considerar los posibles resultados con convenientes matrices de medición y aplicarlas al estado final de la evolución.

3. Ejemplos

3.1 Generación de números aleatorios**.** La instrucción más usada en los algoritmos cuánticos es
la matriz *H* llamada *compuerta de Hadamard*. Tal actúa sobre C^2 de la forma siguiente

$$H|\alpha\rangle = \frac{1}{\sqrt{2}}|0\rangle + (-1)^{\alpha}|1\rangle$$

Sean $P_1, P_2 : C^2 \to C^2$ transformaciones lineales tales que con respecto a la base $|0\rangle, |1\rangle$ tienen matrices

$$P_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

y considérese el siguiente algoritmo cuántico

qRAN(n)
1 Preparación: $|\varphi\rangle = |0\rangle$
2 Evolución: $|\psi\rangle = H|\varphi\rangle$
3 Medición: $|\alpha\rangle = \{P_1, P_2\}|\psi\rangle$

donde $\alpha = 0$ ó $\alpha = 1$. Nótese que la probabilidad de obtener $|0\rangle$ es

$$p = \langle P_1|0\rangle | P_1 0\rangle .$$

De forma similar, la probabilidad de obtener $|1\rangle$ es también 1/2. Por lo tanto, el algoritmo cuántico *qRAND* genera los números 0,1 al azar. Generar números realmente aleatorios en una computadora clásica es una tarea imposible.

3.1 **Codificación superdensa**. Como un ejemplo realmente notable es que la computación cuántica permite la transmisión de dos bits clásicos usando un sólo un estado de un qubit y con una sóla evolución. La transmisión de información requiere de cierto protocolo convenido entre las partes. Por ejemplo, el más conocido es el protocolo TCP/IP. Para el protocolo de codificación superdensa se suponen dos partes que se quieren comunicar entre sí: **A**licia y **B**eto. **A** le quiere enviar dos bits a **B**, pero sólo tiene acceso al envio por una sóla vez. Se supone además que ambos pueden ver el estado inicial $|\varphi\rangle = 1/\sqrt{2}|00\rangle + 1/\sqrt{2}\,|11\rangle$ y que **A** sólo tiene acceso al estado del primer qubit; **B** si tiene acceso a ambos.:

$$|\varphi\rangle = \frac{1}{\sqrt{2}}\left|\underset{\sim}{0}\underset{A}{0}\right\rangle + \frac{1}{\sqrt{2}}\left|\underset{\sim}{1}\underset{A}{1}\right\rangle .$$

Como **A** sólo tiene acceso al estado del primer qubit sólo puede modificar a $|\varphi\rangle$ usando instrucciones de la forma $U \otimes Id$, donde $U : C^2 \to C^2$ unitaria. Sean

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

representaciones matriciales, con respecto a la base $|0\rangle, |1\rangle$ de operadores sobre C^2 (tales se llaman*matrices de Pauli*). Luego el protocolo está en la siguiente tabla

Significado	***Envio***
00	$(Id \otimes Id)\|\psi\rangle = \frac{1}{\sqrt{2}}\|00\rangle + \frac{1}{\sqrt{2}}\|11\rangle$
01	$(Z \otimes Id)\|\psi\rangle = \frac{1}{\sqrt{2}}\|00\rangle - \frac{1}{\sqrt{2}}\|11\rangle$
10	$(X \otimes Id)\|\varphi\rangle = \frac{1}{\sqrt{2}}\|10\rangle + \frac{1}{\sqrt{2}}\|01\rangle$
11	$(iY \otimes Id) = \frac{1}{\sqrt{2}}\|01\rangle - \frac{1}{\sqrt{2}}\|01\rangle$

Lo que significa es que si **A** quiere enviar 00 le aplica al estado que comparte con **B** la instrucción $(Id \otimes Id)$ (i.e, no le hace nada), si 01 entonces aplica $(Z \otimes Id)$, etc. Una vez que **B** recibe el mensaje aplica la instrucción $(H \otimes Id)CNOT$ donde $CNOT : C^4 \to C^4$ está definida por

$$CNOT|\alpha\rangle \otimes |\beta\rangle = \begin{cases} |\alpha\rangle \otimes |\beta\rangle, & si\ \alpha = 0, \\ |\alpha\rangle \otimes X|\beta\rangle, & si\ \alpha = 1. \end{cases}$$

Después del procesamiento se obtiene la siguiente tabla

Significado	***Envio***
00	$\|00\rangle$
01	$\|01\rangle$
10	$\|10\rangle$
11	$\|11\rangle$

enseguida **B** usa las matrices de medición $P_a \otimes P_b$, $a,b = 1,2$. Obteniendo uno de los vectores de la base de cálculo con probabilidad 1. De donde, con toda certeza, **B** puede deducir el mensaje
de **A**licia.

4. Paralelismo cuántico

Como todo estudiante de lógica clásica elemental sabe, el cálculo de tablas de verdad es un asunto fácil pero engorroso. Si una fórmula booleana tiene *n* variables, su tabla de verdad tiene 2^n renglones, es decir, hay que valuar la fórmula booleana 2^n veces. Aún si se utiliza un computador clásico, tendríamos que esperar mucho tiempo (proporcional a 2^n) para obtener respuesta.

Resulta que una computadora cuántica puede hacer cálculos de forma más eficiente que una clásica para determinar cierta clase de fórmulas booleanas: las funciones boolenas constantes o las balanceadas.

Una fórmula booleana se llama *balanceada* si produce el mismo número de ceros que de unos.

Supongamos que de antemano sabemos que una fórmula booleana $f = f(x_1,\ldots x_n)$ de n variables es una de dos: balanceada o constante. Nuestro objetivo es construir algoritmos que puedan distinguir una opción de la otra. En un computador clásico, el resolver tal problema requiere al menos calcular $2^{n-1}+1$ renglones de la tabla de verdad de f, es decir, tenemos que hacer $2^{n-1}+1$ llamadas a la función f. Sorprendentemente, Deutsch y Josza [4] encontraron un algoritmo cuántico que sólo necesita hacer dos llamadas a f. El truco es usar la compuerta de Hadamard actuando en paralelo, de la forma siguiente:

$$H^{\otimes n}|x\rangle = \frac{1}{2^{n/2}}\sum_{y=0}^{2^n-^o}(-1)^{x\bullet y}|y\rangle \qquad (1)$$

donde $|x\rangle,|y\rangle$ son elementos de la base del cálculo, *x,y* números escritos en binario y $x \bullet y$ es el producto escalar bit a bit. Notemos que el lado izquierdo de (1) se produce con n recursos, y del lado derecho aparecen 2^n recursos (sumandos). Esto es gracias a la bilinealidad del producto tensorial y al hecho de que $H^{\otimes n}$ está actuando en paralelo sobre cada qubit de la base del cálculo. Notemos, además que cada sumando *individual* del lado derecho de (1) se obtiene con una probabilidad de $1/2^n$.

Debido a que las transformaciones de los estados tienen que ser matrices unitarias, la función f no puede implantarse directamente, sino como una transformación lineal. Tal transformación es

$$U_f : C^2 \to C^2, \quad |x\rangle \otimes |y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$$

donde $\oplus$ es la de dos bits complemento a 2, o lo que es lo mismo, `XOR` el o exclusivo. Obsérvese que

$$U_f U_f |x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x) \oplus f(x)\rangle = |x\rangle \otimes |y\rangle \qquad (2)$$

es decir $U^2 = Id$.

El siguiente es el algortimo de Deustch-Jozsa. Tal acepta sólo funciones que son constantes ó balanceadas, y lo que hace es decir si la función de entrada fué balanceada ó fué constante.

DeuJoz(f,n)

1 Preparación: $|\psi\rangle = \left|\underbrace{0\ldots00}_{n+1}\right\rangle$

2 Evolución:

3 $|\psi_1\rangle = (H^{\otimes n} \otimes Id)|\psi\rangle$

4 $|\psi_2\rangle = U_f|\psi_1\rangle$

5 $|\psi_3\rangle = (Id \otimes Z)|\psi_2\rangle$

6 $|\psi_4\rangle = U_f|\psi_3\rangle$

7 $|\psi_5\rangle = (H^{\otimes n} \otimes Id)|\psi_4\rangle$

8 Medición: $|\psi_6\rangle = \{ P_{i_1},\ldots,P_{i_n} \mid i_1 = 1,2;,\ldots i_n = 1,2 \}|\psi_5\rangle$

9 **if** $|\psi_6\rangle = |0\ldots00\rangle$ **then return** *constante*

10 **else return** *balanceada*

Debido a la ecuación (1) se tiene que $|\psi_1\rangle = 1/2^{n/2}\sum_{i=0}^{2^n-1}|i\ 0\rangle$. Por la linealidad de U_f, se obtiene $|\psi_2\rangle = 1/2^{n/2}\sum_{i=0}^{2^n-1}|i\ f(i)\rangle$ mientras que la definición de la matriz Z de Pauli asegura que $|\psi_3\rangle = 1/2^{n/2}\sum_{i=0}^{2^n-1}(-1)^{f(i)}|i\ f(i)\rangle$. La definición de U_f o bien, la ecuación (2), asegura que $|\psi_4\rangle = 1/2^{n/2}\sum_{i=0}^{2^n-1}(-1)^{f(i)}|i\ 0\rangle$. El estado final es

$$|\psi_5\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}(-1)^{f(i)+i\bullet j}|j\ 0\rangle = \sum_{j=0}^{2^n-1}\sum_{i=0}^{2^n-1}(-1)^{f(i)+i\bullet j}|j\ 0\rangle$$

Podemos escribir

$$|\psi_5\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)}|0\ldots00\rangle + \frac{1}{2^n}\sum_{j=1}^{2^n-1}\sum_{i=1}^{2^n-1}(-1)^{f(i)+i\bullet j}|j\ 0\rangle \qquad (3)$$

Si f es balanceada entonces el coeficiente del primer sumando del lado derecho de (3) es cero. Por lo que, después de la medición, que aparezcan estados diferentes de $|0\ldots00\rangle$ es un evento con probabilidad 1. Mientras que si f es constante, entonces el coeficiente de $|0\ldots00\rangle$ de la misma ecuación (3) es 1. Es decir, la probabilidad de que aparezca $|0\ldots00\rangle$ es 1. Por lo tanto, podemos decir que la medición del estado $|0\ldots00\rangle$ indica que la función es constante, y si éste estado no aparece podemos decir, con toda seguridad, que la función es balanceada.

Nótese que se hacen dos llamadas a la función *f* en las lineas 4 y 6 de *DeuJoz*. Se puede modificar ligeramente tal algoritmo para hacer sólo una llamada a f (tal modificación se debe a R. Cleve, A. Ekert, C. Macchiavello y M. Mosca, ver [5, p. 44]).

Bibliografía

[1] M. A. Nielsen y I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambrige, Inglaterra, 2002.
[2] G. Brassard y P. Bratley, Fundamentals of Algorithmics, Prentice Hall, Nueva Jersey, E.U. 1996.
[3] R. Cleve, Introduction to quantum complexity theory, arXiv:quant-ph/9906111.
[4] D. Deutsch y R. Jozsa, {\it Rapid solution of problems by quantum computation}, Proc. Roy. Soc. London A, {\bf 439}, 553-538, 1992.
[5] A. O. Pittenger, {\bf An Introduction to Quantum Computing Algorithms}, PCS 19,Birkhausser, Boston, 2001.