Computación cuántica.

Nasser Darwish Miranda

Universidad de La Laguna

Índice general

1 Introducción: El por qué de la computación cuántica.

Antes de comenzar a hablar de la computación cuántica tal vez sea importante señalar que hay mucho más que decir sobre este tema que simplemente ceñirnos a una implementación particular de una máquina para hacer cálculos. El computador cuántico es una idea que crece a medida que se desarolla la teoría cuántica de la información. De este modo cada una de estas entidades crece con ayuda de la otra. Hoy en día el computador cuántico encuentra enormes dificultades para ser construído en la práctica, hasta el punto que determinados autores opinan que es inviable. Si bien yo opto por una postura más optimista, lo cierto es que a medida que vayamos afrontando los problemas que surgen a la hora de crearlo, aprenderemos más y más sobre cómo funcionan los sistemas cuánticos y sobre cómo se comporta la informacion que de ellos somos capaces de obtener.

Hasta hace relativamente poco tiempo nuestra preocupación acerca de los sistemas físicos giraba en torno a, dadas unas determinadas condiciones iniciales, averiguar cómo evolucionarían con el tiempo. Para caracterizar estos sistemas de una manera adecuada fueron surgiendo conceptos, *magnitudes*, con las que éramos capaces de representarlos sin necesidad de referirnos directamente a ellos. Sin embargo, aunque inconscientemente, al hacer esto fuimos desarollando un lenguaje orientado a la información que de los sistemas podíamos extraer, sin que ello implicase al sistema propiamente dicho. No fue hasta la llegada de la mecánica estadística, y más tarde a la *teoría de la información*, que nos empezamos a preocupar por la información como una entidad. Y de hecho es la física la ciencia adecuada para estudiar el comportamiento de la información. Despues de todo, vivimos en un mundo sensorial y creamos el castillo de conocimientos que tenemos sobre él en base a nuestras sensaciones.

La mecánica estadística se vio enfrentada a la necesidad de caracterizar el comportamiento de ciertos sistemas físicos para los que no nos es posible obtener en ningun momento información completa sobre su estado. No sólo era inabordable la medición de cada uno de los parámetros dinámicos de las partículas que componen un sistema macroscópico, sino que además es inabordable la solución del problema de más de dos cuerpos y, aún peor, el tamaño del problema no se escala linealmente con el número de partículas, pues cada una de ellas interactúa con todas las demás. Este asunto será

tratado más adelante, desde otra perspectiva. Lo interesante de la mecánica estadística es que dio significado a una serie de cantidades que procedían de la termodinámica, y que carecían de significado. Abordar el problema de la evolución de un gas desde un punto de vista puramente mecánico, además de muy complejo, por decirlo de alguna manera, es tambien improductivo, pues daría lugar a una caracterización puramente mecánica. De esta forma, ni la entropía ni la presión, temperatura... etc. tendrían aún sentido, pese a que son efectivamente medibles. En este caso es precisamente la energía cinética de cada partícula lo que está fuera de nuestro mundo sensorial, así como el resto de cantidades introducidas por la mecánica, ya sea clásica o cuántica. La idea de entropía, en particular, hace pensar en la información de los sistemas físicos como una entidad física independiente, que evoluciona de acuerdo a unas ecuaciones particulares, como se sabía que hacía el momento lineal, por ejemplo. En la paradoja del demonio de Maxwell, la violación del segundo principio de la termodinámica requería la presencia de un "personaje" capaz de decidir cuándo una partícula iba a atravesar la barrera que separaba dos recipientes, ademas de cual era su estado de movimiento, a fin de aumentar el gradiente térmico. Precisamente eso es disponer de información sobre el sistema, información que desde nuestra escala no es accesible, ya no sólo por las características del sistema, sino tambien por la entropía de éste: si no conocemos el microestado particular del sistema, no podemos abrir y cerrar la barrera cuando se supone que lo haría el demonio de Maxwell. De todas formas no es mi intención entrar a discutir la paradoja en sí, sino hacer ver que la idea de *información* es el centro sobre el que gira la física estadística.

Resultó mucho más evidente la necesidad de caracterizar el comportamiento físico de la información a raíz del surgimiento de la mecánica cuántica. El principio de incertidumbre de Heisemberg

$$\triangle x \triangle p \geq \hbar \tag{1.1}$$

establece una cota superior para el grado de información que podemos extraer de los sistemas físicos. Nos encontramos con que la información desempeña un papel esencial en la descripción de los sistemas cuánticos. Por otro lado, estos sistemas presentan comportamientos particulares, no observables a escala macroscópica, que por una parte deseamos comprender y por otra, si fuera posible, aprovechar para obtener algún tipo de beneficio. Más adelante descubriremos que existen problemas particulares para los que un sistema cuántico aporta una aceleración crítica en el sentido de hacer resoluble un problema que no lo es en la actualidad.

Desde el punto de vista tecnológico, las técnicas empleadas para construir circuitos integrados se acercan a sus límites, debido a la física de los procesos empleados. Hablo por ejemplo de difracción en lo que respecta a la luz con la que realizamos la litografía. Y si bien podríamos olvidar estas limitaciones, ocurre que desde el comienzo todos los computadores han estado basados en las leyes de la mecánica clásica. Estas leyes han alcanzado los niveles

más avanzados de aplicabilidad en la microelectrónica moderna, y aunque podamos desarrollar diseños cada vez más óptimos para los circuitos, éstos se apoyarán siempre sobre un modelo antiguo, como es el de puertas convencionales, y nunca podremos escapar a las limitaciones inherentes a él. Más adelante hablaré sobre estas limitaciones y sobre en que medida pueden resolverse recurriendo a la mecánica cuántica.

Aparte de lo que es el computador cuántico en sí, aprenderemos lo que nos sea posible sobre la teoría cuántica de la información. Encontraremos hechos sorprendentes, que oblgan a matizar resultados clásicos y hacen posibles cosas prohibidas por la relatividad especial. Sobre esto incidiré en el apartado referente al *teletransporte cuántico* y el resto de problemas que un QC sí es capaz de resolver.

Muchas de las fuentes que cito se refieren al computador cuántico como una entidad abstracta, pero otros dedican sus investigaciones hacia cómo construir estos dispositivos en la práctica, y a esto dedico la última parte del trabajo. De cualquier modo, sean más o menos optimistas los autores respecto a la posibilidad de llevar un diseño de computador cuántico a la realidad, todos han dedicado su tiempo y su interés en este asunto, cosa que indica en mi opinión que el tema resultará (espero que tambien este trabajo) gratificante y una ayuda en el estudio de la interpretación de la mecánica cuántica.

2 Teoría cuántica de la información.

Antes de comenzar a hablar sobre computación cuántica presentaré las bases sobre la ciencia que estudia el comportamiento de la información, y sobre su conexión con la mecánica cuántica.

2.1. Teoría clásica de la información.

Comenzaré hablando a nivel básico sobre la teoría clásica de la información. Las ideas que introduciré serán necesarias para entender el resto del trabajo.

2.1.1. Medida de la información.

El primer problema que nos deberíamos plantear es el de la medida de la información. Parece intuitivo decidir en que medida conocemos un sistema, pero necesitamos una formalización. La pregunta puede plantearse en unos términos bastante sencillos:

> *Supongamos que nos dan el valor de un cierto número, X. ¿Cuánta información obtenemos a partir de esto?*

Bien, esto depende de lo que supiésemos previamente sobre ese número. Por ejemplo, digamos que ya sabíamos el valor. En tal situación habremos aprendido exactamente *nada*. Por otra parte, pongamos que sabíamos que el valor X es obtenido al tirar un dado. En este otro caso desde luego que habremos obtenido información. Más tarde hablaré sobre cantidades.

Una observación: una medida de la información es a su vez una medida de la ignorancia, puesto que la información que, dependiendo del contexto, contenga X, es precisamente la que ganaríamos al conocer su valor, y por lo tanto parte de la incertidumbre inicial.

Entropía de Shannon.

Sea X una *variable aleatoria,* que toma el valor x con probabilidad p(x). Definimos el *contenido de información*, o *entropía* de X como:

$$S(\{p(x)\}) = -\Sigma_x p(x) log_2 p(x) \tag{2.1}$$

Ahora queda interpretar esta definición: Que utilicemos base 2 para el logaritmo es simple cuestión de convenio. La entropía es siempre positiva, dado que p(x) está normalizada a la unidad, y por tanto el logaritmo resulta negativo o cero. S es una función de la *distribución de probabilidades* de los valores de X. Es normal que esto sea así, dado que mayor información esperaremos obtener de la cantidad X si los valores que puede tomar son equiprobables que si alguno de ellos es casi seguro. Asumiendo la notación del documento de Steane [1] en lo sucesivo indicaré S({p(x)}) como S(X), no perdiendo de vista lo que en realidad significa. Estamos acostumbrados a hablar de entropía en el contexto de la física estadística, pero definida de este modo adquiere significado en cualquier disciplina donde el grado de conocimiento que tengamos sobre un sistema sea importante, como en biología o economía.

Veamos algún ejemplo sobre cómo funciona esta expresión:

1. Supongamos que sabemos de antemano que X tomará el valor 2, es decir, la distribución de probabilidad es una delta:

   $$p(x) = \delta(2)$$

   o bien, p(x)=0 para x$\neq$ 2 ; p(2)=1. Todos los valores posibles de x son recorridos por el sumatorio, pero sólo cuando x=2 tendremos un término no nulo:

   $$-\Sigma_x p(x) log_2 p(x) = -p(2) log_2 p(2) = -log_2(1) = 0$$

   En otras palabras, no aprendemos nada, como dije antes. X no contiene información.

2. Consideremos la situación en la que el valor de X viene dado al tirar un dado. Los valores posibles para X son los enteros, del uno al seis {1,2,3,4,5,6}, todos ellos en principio con la misma probabilidad, $\frac{1}{6}$. Si sustituímos las cantidades en la expresión (1):

   $$-\Sigma_x p(x) log_2 p(x) = -\Sigma_1^6 [\frac{1}{6} log_2(\frac{1}{6})] = -log_2 \frac{1}{6} \simeq 2{,}58$$

Cuando X puede tomar N valores diferentes, el valor de la entropía se maximiza cuando todos tienen igual probabilidad, p(x)=N^{-1}. Sobre esto ya apunté algo antes. Ganaremos más información al conocer el valor que toma X cuanto menos sepamos en principio sobre cuál puede ser. El que los posibles valores de X sean equiprobables es claro que hace máxima nuestra incertidumbre. La máxima cantidad de información que puede almacenarse en una varible que puede tomar N valores diferentes corresponde a todos con probabilidad $\frac{1}{N}$, es decir:

$$S_{máx}(X) = -log_2(\frac{1}{N}) = log_2(N)$$

$$X \in \{x_i\}_{i=1}^N \tag{2.2}$$

¿Y cuánto vale la unidad de información?

Si partimos del análisis de una variable que puede tomar sólo dos valores con igual probabilidad, al aplicar (2.1) encontramos S(X)=1. Esto va asociado al hecho de haber elegido base 2 en la definición. Otra escala habría llevado a otras medidas, claro, y a una unidad de entropía con un valor diferente, pero somos libres de escoger, al igual que hacemos con cualquier otra magnitud.

Cuando una variable puede tomar sólo dos valores, es claro que la probabilidad de que tome uno de ellos depende de la de que tome el otro. Si X puede valer sólo 1 o 0, entonces p(x=0)=1-p(x=1) , pues el sistema no puede escoger más alternativas. Todo el contenido informativo de X depende entonces de una única probabilidad. Sustituyendo en (2.1) los parámetros correspondientes a un sistema de este tipo:

$$H(p) = -plog_2p - (1-p)log_2(1-p) \tag{2.3}$$

obtenemos la entropía de un sistema de dos estados o, en adelante, simplemente *función entropía*. La función entropía de este modo definida toma valores entre 0 y 1.

Entropía condicional e información mutua.

La entropía condicional se define a partir de la probabilidad condicionada. Representamos la probabilidad de que dado un valor X=x para un parámetro, tengamos Y=y, como p(x|y). A partir de él, la entropía condicional S(x|y) se define como:

$$S(x|y) = -\Sigma_x p(x)\Sigma_y p(y|x)\log p(y|x) = -\Sigma_x\Sigma_y p(x,y)\log p(y|x) \tag{2.4}$$

La segunda de las igualdades se obtiene de

$$p(x,y) = p(x)p(x|y) \tag{2.5}$$

que es la probabilidad de que X=x al mismo tiempo que Y=y.

Podemos interpretar la definición observando que S(x|y) da una idea de cuanta información permanecería oculta en Y si estuviésemos interesados en el valor de X. Conviene observar que Y contendrá menos información en general, nunca más, cuanto más sepamos sobre X, y que ambas variables no tienen por que determinar el valor de la otra en igual medida:

$$S(Y|X) \leq S(Y)$$

$$S(X|Y) \neq S(Y|X) \tag{2.6}$$

La primera de estas expresiones se convierte en una igualdad cuando el valor de X no determina de ningun modo el valor de Y. Sobre la segunda expresión, podemos imaginar, por ejemplo, una relación condicional, pero no biunívoca. De este modo podríamos utilizar:

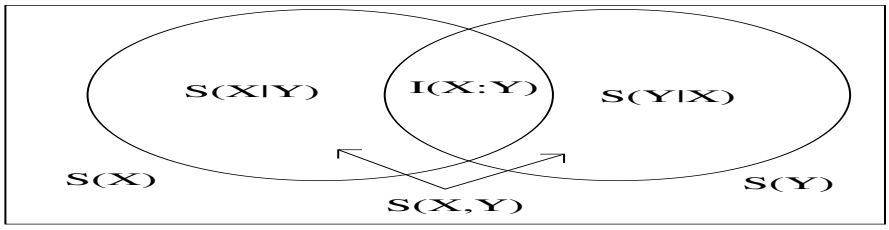
"*Si X toma valor 2, entonces Y tomará valor cero*"

En esta situación, saber que X=2 determina completamente el valor de Y, mientras que conocer el valor de Y no nos dice nada sobre el de X.

A partir de la entropía condicional podemos definir la *información mutua* como :

$$I(X;Y) = \Sigma_x \Sigma_y p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)} = S(X) - S(X|Y) \tag{2.7}$$

Esta cantidad da idea de cuanta información contiene cada variable sobre la otra. Si se trata de variables independientes entoces p(x,y)=p(x)p(y), y la información mutua es *ninguna*. La siguiente figura procede de [1], como casi todo el apartado, pero he creído conveniente incorporarla porque ilustra muy bien estas ideas



Variables con información mutua.

Comprobaré ahora que el conenido informativo de la pareja (X,Y), esto es, la información que obtendríamos al averiguar ambos valores sin conocer inicialmente ninguno, obedece a:

$$S(X,Y) = S(X) + S(Y) - I(X;Y) \tag{2.8}$$

esto es, la suma de la información asociada a ambas variables menos la que comparten, para no duplicarla. Para hacerlo recurriré a la definición de entropía, ec. 2.1:

$$S(Z) = -\Sigma_z p(z) log_2 p(z)$$

donde Z es el hecho compuesto (X,Y). De este modo, Z=z significará que (X,Y)=(x,y). Ahora el problema se reduce a sustituir la probabilidad p(z) en la expresión anterior. Se trata de la probablidad de que simultáneamente X e Y tomen unos valores previstos. Esto podemos hacerlo utilizando 2.5:

$$S(Z) = -\Sigma_z p(x,y) log_2 p(x,y) = -\Sigma_x \Sigma_y [p(x)p(y|x)] log_2 \{[p(x)p(y|x)]\}$$

desarrollando:

$$S(X,Y) = -\Sigma_x p(x) \Sigma_y p(y|x) [\log_2 p(x) + \log_2 p(y|x)]$$

$$S(X,Y) = -\Sigma_x p(x) \Sigma_y p(y|x) \log_2 p(x) - \Sigma_x p(x) \Sigma_y p(y|x) \log_2 p(y|x)$$

Utilizando la definición de información mutua 2.7 puedo poner la expresión anterior como:

$$S(X,Y) = -\Sigma_x\Sigma_y p(x)p(x|y)\log p(x) - \Sigma_x\Sigma_y p(x)p(x|y)\log p(x|y) +$$

$$+\Sigma_x\Sigma_y p(x)p(y|x)\log p(y) - \Sigma_x\Sigma_y p(x)p(y|x)\log p(y)$$

donde he sumado y restado un término al final. Agrupando convenientemente:

$$S(X,Y) = -\Sigma_{x,y}p(x)p(x|y)\log p(x) - \Sigma_{x,y}p(x)p(x|y)\log p(y) - \Sigma_{x,y}p(x,y)\log\frac{p(x|y)}{p(y)}$$

multiplicando y dividiendo por p(x) los argumentos del logaritmo:

$$S(X,Y) = -\Sigma_{x,y}p(x)p(x|y)\log p(x) - \Sigma_{x,y}p(x)p(x|y)\log p(y) -$$

$$-\Sigma_{x,y}p(x)p(x|y)\log\frac{p(x)p(x|y)}{p(x)p(y)}$$

que no es otra cosa que:

$$S(X,Y) = S(X) + S(Y) - I(X;Y)$$

basta con interpretar los términos para comprobarlo, teniendo en cuenta que la probabilidad de obtener X=x o Y=y está condicionada por el valor de la otra variable.

Por otra parte, si bien la informacion puede perderse, esta no puede surgir de ninguna parte. Esto lo expresamos por medio de la *desigualdad del procesaniento de datos* :

$$(X->Y->Z) => I(X;Z) \leq I(X;Y) \tag{2.9}$$

Esto es, si X conduce a Y, e Y conduce a Z (hablamos de una cadena markoviana[1]), entonces la información que comparte X con Z en ningún caso podrá ser mayor que la que comparte con Y. En un proceso de este estilo Z depende directamente de Y, pero no de X: $p(x,y,z) = p(x)p(y|x)p(z|y)$. Una forma de expresar lo que ocurre es, al tratar a Y como un procesador de la información, decir que no puede transmitir a Z más informacion que la que recibe de X.

2.1.2. Compresión de la información.

Es cierto que hemos definido una unidad de información, pero aún no sabemos si proceder así sin más es un modo *adecuado* de cuantificarla, o si existe algún otro modo de hacerlo. Plantearé la siguiente situación:

[1]Una cadena markoviana es aquélla en la que un elemento está relacionado con el anterior y con el siguiente, pero con ningun otro más.

Supongamos que una persona, que en lo sucesivo se llamará Alice, conoce el valor de X, y quiere transmitirlo a Bob. Consideraré solo el caso en que X puede tomar dos valores (sí o no). Podemos considerar que Alice es una fuente de información con un alfabeto de dos símbolos. Mediremos la cantidad de información que Alice envía a Bob contando cuantos bits en promedio necesita enviar para que éste se de por enterado del valor de X. La posibilidad más natural es enviar un 0 o un 1, consiguiendo con ello una tasa de transferencia de un bit por valor comunicado.

Sin embargo, es interesante el caso en que si bien X es una variable aleatoria, uno de sus valores es mas probable que el otro. En este caso podemos proponer un esquema más eficiente que el anterior:

Sea p la probabilidad de de tener X=1, mientras que 1-p será la de que X=0. Entonces Alice puede esperar a n ocurrencias del parámetro X, con n un número relativamente grande. El número promedio de unos que contendría tal secuencia lo denominaré np. Cualquier secuencia de esta duración tendrá *aproximadamente* el mismo número de unos. La probabilidad de obtener cualquier secuencia con np unos:

$$p^{np}(1-p)^{n-np} = 2^{-nH(p)} \tag{2.10}$$

El lado derecho de la ecuación es una generalizacíon del lado izquierdo.

Definiremos una *secuencia típica* como aquella que obedezca a:

$$2^{-n(H(p)+\varepsilon)} \leq p(secuencia) \leq 2^{-n(H(p)-\varepsilon)} \tag{2.11}$$

La probabilidad de los n valores de Alice formen una secuencia típica es mayor que 1-ε. Con n suficientemente grande no importa cómo de pequeño sea ε. Alice no necesita comunicar todos los bits, sino sólo indicar a Bob *cuál de las secuencias típicas* está enviando. Así no hará falta enviar toda la secuencia, sino sólo una etiqueta, sobre la que ambos se habrán puesto de acuerdo.

Puede demostrarse que las secuencias típicas tienen todas la misma probabilidad, y que hay un total de $2^{nH(p)}$ secuencias de este tipo. Para comunicar $2^{nH(p)}$ valores diferentes sabemos que basta transmitir $nH(p)$ bits. Esto es lo mejor que puede hacerse, pues las secuencias típicas son equiprobabeles. Esto, a su vez, significa que el contenido en información de cualquier valor de X en la secuencia original es H(p), lo que está de acuerdo con la definición de entropía 2.1. Para comunicar n valores distintos de X sólo necesitamos enviar nS(X)$\leq$n bits. Cuando el contenido informativo de X es máximo (1) estamos en el caso límite: no podemos dejar de enviar bits sin dejar de enviar información. A esta posibilidad es a la que se conoce como *compresión de los datos*. El teorema que asegura esto se denomina *teorema de la comunicación sin ruido de Shannon*.

Tenemos un inconveniente cuando utilizamos secuencias típicas para comprimir información: Alice primero tiene que decidir que secuencia típica va a enviar, y esto puede llevar mucho tiempo. Podemos elegir otra alternativa:

Imaginemos que Alice espera poco, por ejemplo cuatro decisiones, y comunica este mensaje de la manera más compacta posible. Entonces elige palabras de código más cortas para las secuencias más frecuentes, y palabras más largas para las menos probables. El método de Huffman hace precisamente esto. El ejemplo de la tabla representa la elección para p=1/4:

Codificación BCD	Huffman (compresión)	Hamming (errores)
0000	10	0000000
0001	000	1010101
0010	001	0110011
0011	11000	1100110
0100	010	0001111
0101	11001	1011010
0110	11010	0111100
0111	1111000	1101001
1000	011	1111111
1001	11011	0101010
1010	11100	1001100
1011	111111	0011001
1100	11101	1110000
1101	111110	0100101
1110	111101	1000011

Tabla I: El código Huffman comprime la información, mientras que el código de Hamming detecta y corrige errores. Hablaré de esto último a continuación.

El canal binario simétrico.

Hasta ahora he hablado sobre la comunicación en ausencia de ruido, y por tanto, de errores. Antes de hablar sobre la corrección de los errores es conveniente hacerlo sobre el comportamiento de los canales, pues este es determinante ante los tipos de errores que surgirán. Las ideas que presento aquí adquirirán relevancia en el caso de los sistemas cuánticos, pues en ellos adquiere gran importancia el fenómeno conocido como decoherencia. Si bien no es el momento de empezar a discutir sobre decoherencia, sí lo es para hablar en términos generales de tipos de errores y de estrategias para corregirlos.

El tipo de canal más sencillo que podemos estudiar se conoce como *canal binario simétrico.* Un canal de este tipo introduce aleatoriamente errores en los bits, cambiando unos por ceros y ceros por unos con igual probabilidad. Además del más sencillo es tambien el más importante, pon un lado debido a que modeliza bien muchos canales que existen en la realidad, y por otro porque cuando no sabemos nada sobre un canal lo más razonable es suponer

que tiene un comportamiento de este tipo. Podemos imaginar otros canales, como uno que sólo sustituya los unos por ceros, pero no al conrario, etc. En los sistemas cuánticos nos preocuparán tambien otros tipos de error de las funciones de onda, debidos a acoplamientos con otros sistemas (esto es, la decoherencia).

Un canal binario simétrico sólo necesita de un parámetro para ser caracterizado, pues la probabilidad de error es la misma tanto para los ceros como para los unos. El parámetro escogido será p, la probabilidad de error por bit enviado. Llamaré X al mensaje enviado por Alice, e Y al mensaje recibido por Bob, quien tiene como objetivo recuperar el original. Imaginemos que el mensaje consta sólo de un bit. A través de la probabilidad condicionada:

$$p(x = 0|y = 0) = p(x = 1|y = 1) = 1 - p$$

$$p(x = 0|y = 1) = p(x = 1|y = 0) = p$$

Estas expresiones dicen: la probabilidad de recibir un bit, habiendo sido enviado el otro, es la probabilidad de error de bit, mientras que la probabilidad de que el error no se produzca es uno menos esa probabilidad, como es de esperar.

Pero cualquier mensaje es mayor de un bit. Entonces podemos recurrir a la *entropía condicional*, definida en 2.4:

$$S(X|Y) = H(p)$$

Con esta expresión no digo más que la información que se pierde en un error de bit es precisamente *toda* la información que el bit contiene. Recurriendo a la definición de *información mutua* 2.7, podremos escribir:

$$I(X;Y) = S(X) - H(p) \tag{2.12}$$

En otras palabras, la información que comparten ambos sistemas es la que contiene uno de ellos menos la información perdida debido a errores en la comunicación. Más ruido en el canal significa un valor de p más alto, lo que se traduce a su vez en mayor pérdida de información y menor información mutua.

Hasta el momento he hablado de *canal* no como una entidad particular, sino como aquello que distingue un sistema de otro. Sin embargo el canal es un sistema, y como tal desearíamos caracterizarlo. Definimos la *capacidad del canal* como la máxima cantidad de información que pueden compartir dos sistemas que se comuniquen a través de él, maximizada a todos los tipos de fuentes:

$$C = max_{\{p(x)\}} I(X;Y) \tag{2.13}$$

La definición observa todo tipo de fuentes. Imaginemos que cierto canal permuta los ceros por unos con una determinada probabilidad, pero deja los unos

sin errores. Una fuente que emita sólo unos no sería afectada por error, pero sí lo sería cualquier otra. En particular, si sólo enviásemos ceros la cantidad de errores introducidos sería máxima. De aquí se explica que no podamos construir una definición de capacidad que no tenga en cuenta la fuente.

La expresión 2.13 tiene como inconveniente su falta de operatividad. El problema de la optimización es complejo, sobre todo a la hora de decidir cuáles son todas las fuentes posibles. Nos gustaría disponer de un método analítico para comparar los canales entre sí. Para el caso del canal binario simétrico, a partir de 2.12 y 2.13 , tomando S(X)=1 (esto es, máximo contenido informativo) ,vemos directamente que la capacidad es:

$$C(p) = 1 - H(p)$$

Códigos correctores de error.

Ante la posibilidad de encontrar errores en una comunicación una estrategia de recuperación consiste en incorporar alguna información adicional (por ejemplo, en cada palabra del código) sobre el formato que esperamos que se reciba. De este modo, si el error aparecido rompe este formato podrá ser detectado fácilmente. Una posibilidad es la paridad: podemos añadir un bit al final de cada palabra, de modo que siempre haya un número par de unos (ceros). Si se produce un error en un uno podríamos entones detectarlo. Sin embargo, si se produce un error en un cero, o si el error aparece en dos unos, sería indetectable. Obviamente, unas estrategias serán preferibles a otras en función del ámbito. No estamos interesados en corregir errores que no esperamos que se produzcan, ni podemos ignorar errores altamente probables.

Será necesario hacer algunas observaciones:

El conjunto {0,1} es un grupo, donde definimos las operaciones +,-,×,÷, con módulo 2 (esto es, 1+1=0) .

Una palabra de n bits es un vector. El conjunto de palabras posibles forma un espacio vectorial ante la suma:

$$(1,0,1) + (0,0,1) = (1,0,0)$$

Aquí podemos observar que la suma equivale a la operación XOR, efectuada sobre cada bit.

El efecto del ruido, entonces, consistente en transformar unas palabras en otras: $\mathbf{u} \rightarrow \mathbf{u'} = \mathbf{u} + \mathbf{e}$, donde **e** es el vector de error. El vector de error tiene componentes nulas salvo allí donde modifica las del vector original. Con esto, podemos definir un *código corrector de error* como un conjunto de palabras construído de modo que:

$$\mathbf{u} + \mathbf{e} \neq \mathbf{v} + \mathbf{f}, \forall \mathbf{u}, \mathbf{v} \epsilon C(\mathbf{u} \neq \mathbf{v}), \forall \mathbf{e}, \mathbf{f} \epsilon E \tag{2.14}$$

siendo E el conjunto de todos los errores que el códogo es capaz de corregir. El código debe como mínimo no alterar un mensaje sin errores, de modo que

e=0 debe estar contenido en E. Un código detector de error es el de Hamming [2,3], representado en la tabla (1).

El código de Hamming es un código [7,4,3]. La notación corresponde a [n,k,d], donde n es el número de bits de cada palabra, 2^k es el número de palabras del código y d es la distancia entre palabras (el número mínimo de bits en que difiere cualquier par de palabras del código). Este código corrige como máximo un bit. Cualquier combinación de los n bits que no sea palabra del código puede identificarse como errónea. Como hay n bits tenemos 2^n combinaciones posibles, de las que 2^k son palabras, de modo que E contiene como máximo 2^{n-k} miembros. No podemos corregir palabras que pasen desapercibidas en el código. Por otra parte, un código de distancia mínima d puede corregir todos los errores que aparezcan en menos de d/2 de los bits transmitidos de cada palabra. La probabilidad de recibir m errores viene dada por la distribución binomial:

$$P(n,m) = C(n,m)p^m(1-p)^{n-m} \tag{2.15}$$

para el número m de unos en una secuencia de n valores binarios. Recordemos que p es la probabilidad de obtener un 1 en un determinado bit. C(n,m) es el número de combinaciones posibles con m unos en secuencias de longitud n:

$$C(n,m) = \left(\begin{array}{c} n \\ m \end{array} \right) = \frac{n!}{m!(n-m)!s}$$

Quedademos satisfechos con un código que sea capaz de corregir un número mayor de errores que el promedio de los que se producen.El *teorema de Shannon*:

> *Si la tasa k/n es menor que C(p) y n es un número suficientemente alto, entonces existe un código binario que permite la transmisión con una probabilidad de error (incorregible) todo lo pequeña que se quiera.*

El problema central de la teoría de la codificación es encontrar códigos de tasas (k/n) elevadas, con distancias mínimas (d) lo mayores posible. Estos requisitos no pueden satisfacerse a la vez, de modo que es necesario buscar un valor de compromiso.

2.2. Teoría cuántica de la información.

2.2.1. Nociones básicas sobre información en mecánica cuántica.

Al principio comenté el hecho de que el tratamiento de la información como entidad independiente adquirió un nuevo significado a raíz del surgimiento de la mecánica cuántica. El apartado que comienza ahora no es más que un

repaso sobre los conceptos más básicos de la física cuántica, pero por completitud he decidido incluirlo. Los postulados de la mecánica cuántica que nos interesan ahora son los siguientes:

1. El estado de un sistema aislado Q se representa por un vector $|\psi$(t)>en un espacio de Hilbert.

2. Las variables, tales como la posición y el momento se denominan *observables* y se representan por operadores hermíticos. En la base de estados propios de X las componentes de la posición y el momento son:

$$\langle x|X|x'\rangle = x\delta(x-x')$$
$$\langle x|P|x'\rangle = -i\hbar\delta'(x-x')$$

3. El vector de estado evoluciona de acuerdo a la ecuación de Schrödinger:

$$i\hbar\frac{d}{dt}|\psi(t)>= H|\psi(t)> \tag{2.16}$$

   donde H es el operador hamiltoniano.

4. El estado de un sistema cuántico inmediatamente despues de que sobre él se efectúe una medida es la proyección del estado anterior sobre el subespacio correspondiente al valor propio obtenido de ésta. En el caso de que este valor propio fuese no degenerado el resultado sería precisamente el vector propio correspondiente.

$$|\psi'(t_0^+)>\sim \Sigma_{i=1}^{g_n}|u_n^i>< u_n^i|\psi(t_0^-)>$$

   Donde no he utilizado el signo igual porque el miembro de la derecha está sin normalizar.

Una consecuencia de los postulados es que la evolución de un sistema cuántico aislado siempre es unitraria:

$$|\psi(t)>= U(t)|\psi(0)>$$

con U, el operador de evolución, unitario:

$$U(t) = exp(-\frac{i}{\hbar}\int Hdt); UU^\dagger = I$$

Pero los sistemas aislados no existen. Modelizar cualquier sistema cuántico por medio de la ec. Schrödinger implica hacer una aproximación más o menos fuerte. Un modo de tratar el problema de los alrededores es simplificándolos como un único sistema, τ, que en cierto modo se comporta como si realizase una medida sobre el estado de Q. Las proyecciones asociadas a la interacción no son unitarias, así que aparece una contribución no unitaria en la evolución. Este fenómeno se conoce como *decoherencia*. Sobre ella y sobre como evitar sus efectos hablaré en la sección 7.

2.2.2. El problema de la simulación.

Despues de recordar las herramientas que necesitaremos, se plantean las siguientes cuestiones:

1. Parece que la naturaleza funciona como un gran procesador de la información, y más a la luz de los postulados de la mecánica cuántica. Un ket, como descriptor de un sistema cuántico en un instante de tiempo, es un paquete de información, y como tal se comporta. Contiene no la información total para especificar por completo el estado del sistema q, sino sólo la que hay disponible, y no información sobre otros sistemas.

2. ¿Puede una computadora simular a la naturaleza en conjunto? Convirtamos la conjetura de Church-Turing en un principio de la física:

   Cualquier sistema físico finito realizable puede simularse con precisión arbitrariamente elevada por una computadora universal con un número finito de estados.

Observemos que el postulado no involucra máquinas de Turing. Hay grandes diferencias entre las máquinas de Turing y los principios que rigen el comportamientoo de los sistemas cuánticos. La idea sobre la que gira la computación cuántica es la posibilidad de llevar a cabo nuevos tipos de cálculos, completamente distintos a los llevados a cabo por los computadores tradicionales. Y hay resultados que llevan a pensar que realmente existe tal posibilidad. El problema de la simulación fue tratado por Feynman en su famoso artículo de 1982, en el que tambien apuntó a la posibilidad de construir un computador basado en las leyes de la mecánica cuántica.

Un computador cuántico en principio parece obvio que serviría para simular sistemas cuánticos. Supongamos que pretendemos simular un sistema cuyo espacio de Hilbert sea de dimensión 2^n mediante un computador clásico. Está claro que necesitaremos 2^n números complejos, las componentes del vector de estado. Un computador cuántico, en cambio, requiere tan sólo de n qubits para hacer lo mismo. Así que a nivel de almanenamiento la ventaja de un computador cuántico sobre uno clásico es obvia. A nivel de cálculo, ni uno ni otro resultarán en general eficientes, pues mientras que para un computador clásico debemos manipular matrices de dimensión 2^n (lo que equivale a número de cálculos exponencial con el tamaño de la entrada, n) un computador cuántico deberá realizar transformaciones unitarias sobre un espacio de 2^n dimensiones, cosa que necesitría un número de puertas que crece en la misma medida.

Se demuestra que existen *algunos* sistemas para los que en efecto un computador cuántico puede proceder de manera más eficiente que uno clásico, si bien esta no es la situación general. Fuentes sobre ello: Lloyd 1996, Zalka 1996, Wiesner 1996, Meyer 1996, Lidar y Biam1996, Abrams y Lloyd 1997, Boghosian y Taylor, 1997.

2.3. Las bases de la computación cuántica: el experimento de Einstein, Podolsky y Rosen.

2.3.1. Primera sorpresa: Es posible medir sin alterar un sistema.

Consideremos parejas de sisetmas cuánticos de dos estados, como partículas de spin 1/2. Llamemos a nuestras partículas A y B, y asociemos los múmeros cuánticos del modo acostumbrado:

$$spinup : m_z = \frac{1}{2}; |\uparrow>$$

$$spindown : m_z = -\frac{1}{2}; |\downarrow>$$

Preparamos el sistema en el estado inicial:

$$|\psi(0)>= \frac{1}{\sqrt{2}}(|\uparrow>|\downarrow> -|\downarrow>|\uparrow>)$$

Entonces ambas partículas se mueven en sentidos opuestos a lo largo de un eje coordenado cualquiera, por ejemplo OY. Por un lado Alice recibe su partícula, y por otro lo hace Bob. El resultado permite predecir con certeza la medida de cualquier componente de $\mathbf{s}_B$, sin perturbar a B. De este modo, las componentes del spin de B están definidas, y por tanto la descripción que hace la mecánica cuántica del sistema es incompleta.

Para hacer la predicción, escogemos cualquier eje η a lo largo del que nos gustaría conocer el momento angular de B. Pero en lugar de medir el spin de B medimos el de A, con un aparato de Stern-Gerlach, alineado con η. El spin del estado singlete que preparamos al principio era cero, de modo que por conservación del spin, el de B debe ser siempre opuesto al de A.

¿Dónde está el problema? Las alternativas habituales:

1. La medida que realiza Alice influye en el estado de la partícula que recibe Bob.

2. El vector de estado del sistema cuántico $|\psi>$no es una propiedad intrínseca del sistema cuántico, sino una expresión del contenido informativo de una variable cuántica. En el estado singlete hay información mutua entre A y B, de modo que el contenido informativo de B cambia cuando conocemos el valor de A. Desde este punto de vista el comportamiento puede describirse usando la teoría clásica de la información.

2.3.2. Segunda sorpresa: Las desigualdades de Bell.

Imaginemos que Alice y Bob miden spines a lo largo de distintos ejes, η_Ay η_B ,contenidos en el mismo plano, digamos XZ. Cada medida tiene dos resultados posibles: spin up , o spin down. Teoría y práctica coinciden en que la

probabilidad de que el resultado encontrado por ambos sea el mismo es

$$P_{eq} = \sin^2((\phi_A - \phi_B)/2) \tag{2.17}$$

En la expresión ϕ_A y ϕ_B son los ángulos que forman respectivamente los ejes η_A y η_B con el eje OZ. Pero no hay manera de asignar propiedades locales a A y B independientemente. Al obsrvar la ec. 2.17 vemos que en la probabilidad hay un altísimo grado de correlación. Lo que ocurre en B depende del ángulo ϕ_A, y al contrario. El máximo grado de correlación corresponde a ϕ_A-ϕ_B=120^o, que da como resultado 2/3.

Este resultado permite imaginar una tarea que los computadores clásicos son incapaces de realizar: comunicar información más rápido de lo que permitiría una señal luminosa. A partir de ϕ_A y ϕ_B , comunicar señales binarias perfectamente correlacionadas con ϕ_A=ϕ_B+180 , anticorrelecionadas con ϕ_A=ϕ_B , y correlacionadas en más de un 70 % con ϕ_A-ϕ_B=120^o.

Estos resultados fueron comprobados en el laboratorio en los años 70 y 80 (Clauser y Shimony, 1978;Aspect et. al. , 1982; Kwiat et. al. , 1995).

Estos últimos resultados nos colocan por fin en el lugar adecuado para estudiar la teoría cuántica de la información con un poco más de profundidad.

2.4. La información en mecánica cuántica.

A continuación introduciré ideas análogas a las vistas antes para los sistemas cuánticos. Las que no veamos en los siguientes apartados las hemos visto ya, o bien no es aún el momento adecuado para desarollarlas.

2.4.1. Qubits.

Un qubit, a groso modo, es un bit, implementado mediante algún observable, como el spin, de un sistemaa cuántico (Schumacher, 1995). Elegimos lógica de dos estados, como en computación electrónica, así que aprovechamos sistemas de dos niveles. Esto nos hace favorables al uso de partículas de spin 1/2, como electrones. Es importante observar que elegir spin semientero nos lleva a estados antisimétricos. Los qubits no se pueden medir como los bits, en el sentido descrito por el postulado de la medida de la mecánica cuántica. Un estado que no podemos medir sin alterarlo parece que no nos sirve de mucho. Esta es una dificultad a la que nos entrentamos, y para la que existen respuestas. Por otra parte, cada nuevo bit duplica el número de estados posibles codificable, bientras que cada nuevo qubit aumenta al doble la dimensión del espacio en el que existen los estados con los que hacemos los cálculos. En ausencia de requisitos de simetría esto supone una infinidad de nuevos estados. Hablaré con mayor detalle sobre los qubits antes de describir procedimientos que pueden desarrollarse con ellos.

2.4.2. Entropía de Von Neumann.

Antes de nada preocupémonos de si el qubit es una buena medida de la información, como vimos que era el bit. Para ello seguiremos pasos similares a los dados en 2.1.

La *entropía de Von Neumann* es una medida de la cantidad de información que obtendríamos si conociésemos el estado particular de cierto sistema cuántico, y se define como

$$S(\rho) = -Tr\rho log\rho \tag{2.18}$$

donde Tr es la operación de traza, que se realiza sobre la matriz densidad, que describe un conjunto de estados de un sistema cuántico. Si comparamos con la entropía de Shannon:

$$S(X) = \Sigma_x p(x) log p(x)$$

encontramos que las definiciones son muy parecidas. Los elementos diagonales de la matriz densidad desempeñan el papel de las probabiliades de que X tome cada valor. Preparemos un sistema cuántico en el estado |x>, descrito por el valor del observable X. La matriz densidad se escribe:

$$\rho = \Sigma_x p(x) |x><x|$$

Los estados |x>no tienen por qué ser ortogonales. S(ρ) se demuestra (Kholevo 1973, Levitin 1987) que es un límite superior para la información mutua I(X;Y) clásica entre X y el resultado Y de la medida del sistema.

Ahora consideremos los recursos necesarios para almacenar y transmitir la información de un sistema cuántico q cuya matriz de densidad es ρ. Al igual que hicimos con la información clásica, nos gustaría reunir un número elevado de sistemas de este tipo (lo que en el otro caso llamamos secuencia típica) y utilizar un nombre que caracterice al conjunto, compactando así la información. La etiqueta será un sistema cuántico más pequeño, que valdrá como unidad de almacenaje o transmisión con todo el contenido informativo. El receptor de ese paquete reconstruirá un sistema q' , cuya matriz densidad será ρ'. Para que el proceso de transmisión (o de recuperación de una información almacenada) tenga éxito la matriz densidad ρ' debe ser lo suficientemente cercana a ρ. Definimos la *fidelidad* como la cantidad que da cuenta de cuánto se parecen ambas matrices:

$$f(\rho, \rho') = \left(Tr\sqrt{\rho^{1/2}\rho'\rho^{1/2}} \right) \tag{2.19}$$

Esta cantidad puede interpretarse como la probabilidad de que q' pase un test que pretendiese determinar si el estado del sistema es ρ. En el caso de ρ y ρ' estados puros (|ϕ><ϕ| y |ϕ'><ϕ'|) , la fidelidad se reduce a:

$$f = | < \phi | \phi' > |^2$$

que es la probabilidad de encontrar el valor propio asociado al autoestado $|\phi>$cuando el sistema se encuentra en el estado $|\phi'>$.

Nos preocupamos naturalmente por encontrar el paquete más pequeño posible que etiquete al conjunto de estados agrupado. Análogamente al caso clásico, buscamos una fidelidad tan próxima a la unidad como sea posible:

$$f = 1 - \varepsilon; \varepsilon << 1$$

Por simplicidad, al igual que antes, nos limitamos a unidades binarias de información: sistemas de dos estados. Para un conjunto de n sistemas de 2 estados existe un vector en un espacio de Hilbert de 2^n dimensiones que especifica por completo su estado. Pero al igual que antes, esperamos que el vector de estado caiga dentro de un *subespacio típico* del espacio de Hilbert, análogamente a como los arrays de bits tomaban valores de secuencias típicas en el caso clásico. Schumacher y Jozsa demostraron que la dimensión de ese subespacio es $2^{nS(\rho)}$, lo que por analogía con el caso anterior conduce a que sólo son necesarios nS(ρ) qubits para transmitir la información de los n sistemas. La dimensión del espacio sobre el que se representan los estados crece exponencialmente con el con número de qubits, y el qubit es una medida de información.

Importante es tener en cuenta que las operaciones de codificación y decodificación no dependen del conocimento que tengamos sobre el estado del sistema. Esto nos salva en cierta medida del problema de la no clonación, y nos libera del hecho de tener que medir para transmitir información.

En el caso de que los estados a transmitir fuesen ortogonales entre sí el problema se reduciría al caso clásico.

Hay contrapartidas cuánticas a las otras cantidades que vimos antes: la *información coherente* desempeña el mismo papel que la información mutua (Schumacher y Nielson, 1996), y podemos desarrollar códigos análogos al de Huffman para comunicación de información cuántica.

2.4.3. Entrelazamiento.

Una propiedad responsable de la potencia de los métodos de cálculo cuánticos es el *entrelazamiento*. El entrelazamiento es un recurso aprovechado en los métodos de factorización y búsqueda de los que hableré más tarde, aunque tambien es responsable de la decoherencia. Una característica tan importante como esta desearíamos ser capaces de cuantificarla. El criterio que utilizaremos es el de Bennett y Shor [4].

Partamos de que Alice y Bob comarten un sistema cuántico (cada uno accede accede a una parte). Sabemos que hay entrelazamiento entre las dos partes cuando el estado del sistema no se puede representar por una superposición de productos tensoriales. Formalmente, hay entrelazamiento cuando la matriz densidad del sistema no se puede escribir como:

$$\rho^{AB} = \Sigma_i p_i \rho_i^A \otimes \rho_i^B \tag{2.20}$$

donde los estados de cada subsistema no interferirían entre sí. Los superíndices A y B se refieren a los subsistemas a los que Alice y Bob tienen acceso. Tanto Alice como Bob pueden preparar cualquier estado no entrelazado por medio de transformaciones locales y de comunicación clásica: primero escogiendo el índice i, para el que deciden probabilidad p_i , despues preparando los estados locales ρ_i^A y ρ_i^B. La manera en que Alice y Bob se ponen de acuerdo sobre estos estados puede tener tan poco que ver con la mecánica cuántica como tiene una simple conversación.

Veamos en cambio que ocurre si Alice y Bob pretendiesen preparar un estado entrelazado. Ahora los agentes necesitarán compartir desde el principio algun grado de entrelazamiento, cosa que pueden conseguir, si no se diese desde el principio, transmitiéndose mutuamente estados cuánticos (obsérvese que el el caso anterior sólo era necesaria comunicación clásica). Ahora sólo nos falta saber cómo hacen Alice y Bob para decidir que los estados a los que acceden estan entrelados.

Supongamos que ρ^{AB} es un estado puro (por ejemplo, $|\psi^{AB}><\psi^{AB}|$ de rango 1). En esta situación la única medida que nos indicaría la presencia de entrelazamiento es la que se conoce como *entropía del entrelazamiento*:

$$E(\psi^{AB}) = S(Tr_B\rho^{AB}) = S(Tr_A\rho^{AB}) \tag{2.21}$$

La entropía de entrelazamiento es la de Von Neumann 2.18 de un estado mezclado cuando uno de los subsistemas se disocia. Veamos qué hemos hecho:

Una buena medida del entrelazamiento debe cumplir una serie de propiedades: dado que el entrelazamiento es un recurso cuántico, éste no podrá incrementarse vía operaciones y comunicación clásica. La mejor situación posible se da cuando dados dos estados ρ_1y ρ_2 con entrelazamientos respectivos E_1y E_2 (E_1>E_2) el segundo estado es siempre identificable a partir del primero vía operaciones locales y comunicación clásica,pero esto puede ser pedir demasiado. Entonces consideramos un límite de esta situación en el que ρ_1y ρ_2 son estados puros. Para cualquier pareja de estados puros compuestos ψ y ψ' (conteniendo los subsistemas A y B) en el límite de n grande un número n de copias independientes de ψ, es decir $\psi^{\otimes n}$pueden transformase por medio de operaciones locales y de comunicación clásica en un estado arbitrariamente cercano a $\psi'^{\otimes n'}$, donde la fidelidad de tiende a 1 y con $(n'/n) -> (E(\psi)/E(\psi'))$[20].

Es interesante observar que la medida de entrelazamiento que hemos hecho es *aditiva*. Si Alice y Bob tienen subsistemas con entrelazamientos E_1y E_2, entonces el sistema global que comparten tiene entrelazamiento E_1+E_2.

La cantidad de entrelazamiento de estado puro que se necesita para crear un estado mezclado se sabe que es en general menor que la que se puede obtener de ese estado. A la primera cantidad la llamaremos *entrelazamiento de formación*, y a la segunda *entrelazamiento destilable*. A nivel cuantitativo, la definición de entrelazamiento de formación involucra el número de pares EPR (2.3) necesarios para crear copias de un cierto estado con alta fidelidad, y la

del entrelazamiento destilable, el número de pares EPR casi perfectos que se pueden obtener con elevada fiabilidad de las copias del estado.

2.4.4. Puertas cuánticas.

Antes vimos qué era una puerta lógica, y cómo con ellas podemos realizar operaciones elementales con los bits. Para este caso seré un poco más formal.

Una puerta cuántica es un dispositivo capaz de realizar operaciones unitarias sencillas sobre los qubits (Deutch, 1985, 1989)

Un ejemplo, el siguiente: un sistema que actúe sobre el estado |0>sin alterarlo, y sobre |1>, convirtiéndolo en $e^{i\omega t}|1>$ puede contruirse con "algo" que opere sobre el qubit de acuerdo a:

$$P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\omega t} \end{pmatrix} \tag{2.22}$$

He dicho "algo", y no he concretado más, porque podemos utilizar campos magnéticos, radiación electromagnética u otra clase de fenómenos para modificar estados cuánticos. Un mismo dispositivo, capaz de generar campos que apunten en direcciones distintas puede actuar como diferentes puertas, de modo que el término puerta aquí es más abstracto que en electrónica.

Algunas puertas cuánticas son las siguientes:

$$I = |0><0| + |1><1| \tag{2.23}$$

$$X = |0><1| + |1><0| \tag{2.24}$$

$$Z = P(\pi) \tag{2.25}$$

$$Y = XZ \tag{2.26}$$

$$H = \frac{1}{\sqrt{2}}[(|0> + |1>)<0| + (|0> - |1>)<1|] \tag{2.27}$$

I es la identidad, X es la complementación, H es el operador de Hadamard[2].

Una puerta que actúa sobre un qubit tiene la forma:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

y transforma |0>en a|0>+b|1>, y |1>en c|0>+d|1>.

Nos llama la atención el hecho de que si bien para un único bit sólo hay dos puertas posibles (obviamente, la identidad y la complementación: $I = x$ o

[2]La transformación de Hadamard $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ es capaz de pasar de qubits a combinaciones simétricas y antisimétrias: |0><->$(1/\sqrt{2})$(|0>+|1>) ; |1><->$(1/\sqrt{2})$(|0>-|1>).

bien $X = \overline{x}$), en el caso de un qubit hay infinitas puertas posibles. La pueta NOT cuántica se etiqueta como X porque coincide con la matriz $\sigma_x$de Pauli. El conjunto {I,X,Y,Z} constituye un grupo bajo la operación de producto.

No nos preocupamos de cómo construir estas puertas. De hecho, existen puertas con las que es posible implementar cualquier función, de igual modo que podíamos hacer tradicionalmente con puertas AND y OR. De eso hablaré cuando describa diferentes modelos de computador. En el autómata celular veremos cómo podemos hacer que un array de qubits realice operaciones sin necesidad de incorporar puertas lógicas. De hecho, las puertas lógicas sólo conducen a un modelo de computador.

Nos interesaremos en particular por las puertas que puedan escribirse como

$$f = |0><0| \otimes I + |1><1| \otimes U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix} \tag{2.28}$$

donde I es la identidad para un qubit, como vimos antes, y U es alguna otra puerta de una única entrada tambien. A estas puertas se las denomina *U controlada*s, porque el efecto del operador U sobre el segundo qubit depende de si el primero es |0>o |1>.

Un ejemplo especialmente interesante de puerta de este tipo es la que tiene la siguiente forma matricial:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.29}$$

Analicemos esta matriz:

Para el estado de entrada |00>tenemos el elmento $CNOT_{00}$, así que:

$$|00> \rightarrow |00>$$

si la entrada es |01>, entonces la salida es $CNOT_{11}|01>$, es decir:

$$|01> \rightarrow |01>$$

pero cuando el primer bit vale 1, tenemos otra situación. Ahora el segundo bit cambia:

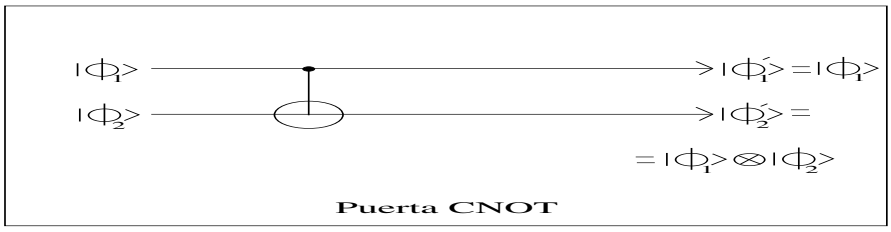
$$|10> \rightarrow |11>$$

$$|11> \rightarrow |10>$$

Como he indicado, esta puerta recibe el nombre de CNOT, o puerta NOT controlada. Se demuestra que esta puerta, en combinación con el conjunto de puertas de un bit constituye un conjunto de primitivas de la computación cuántica, es decir, que puede realizarse cualquier cálculo a partir de ellas [6, 4].

2.4.5. Reversibilidad en computación.

Una puerta que actúa sobre un único qubit efectúa una operación reversible, dado que al tratarse siempre de matrices unitarias, podemos encontrar la inversa. Así, a partir de la salida es posible obtener la entrada. Además si pretendemos que la dimensión del espacio de los estados de salida coincida con la de los de entrada, debemos mantener siempre todas las líneas. En la figura tenemos el ejemplo para la puerta CNOT:



Puerta CNOT

La puerta CNOT es reversible incluso a nivel clásico, dado que a la salida tenemos información suficiente para reconstruir la entrada.

Si quisiéramos un conjunto universal reversible con puertas clásicas necesitaríamos al menos una puerta de tres bits, como la de Toffoli, que efectúa para a,b,c la operación{ a->a;b->b;c->a·b }. De hecho, esta puerta puede simular puertas AND y OR. Pero además tambien permite invertir las operaciones para el caso cuántico, y permite por sí sola tambien en este caso realizar todas las operaciones posibles.

2.4.6. Teorema de no clonación.

Este resultado se relaciona directamente con el postulado de medida en mecánica cuántica, y es una de las dificultades que presenta la construcción del computador cuántico. Un enunciado posible:

Es imposible clonar un estado cuántico desconocido

Copiar un estado a través de una medida es imposible, pues al medir lo modificamos. Pero podemos ser más rigurosos.

Para copiar un estado cuántico $|\phi>$necesitamos que cierta pareja de sistemas evolucione de acuerdo a U($|\phi> |0>$)=($|\phi> |\phi>$). U es el operador de evolución, que no debe depender de $|\phi>$, pues eso significaría que conocemos $|\phi>$, y que hemos preparado un mecanismo para copiarlo, caso excluído en el enunciado. Así, tendremos tambien $U(|\psi> |0>) = (|\psi> |\psi>)$, para $|\psi> \neq |\phi>$.

Entonces consideremos el caso $|\gamma> = (|\phi> + |\psi>)/\sqrt{2}$. En este caso, la evolución generaría:$U(|\gamma> |0>) = (|\phi> |\phi> + |\psi> |\psi>)/\sqrt{2} \neq (|\gamma> |\gamma>)$.

La operación de clonación U se ha descrito de la manera más general posible, de modo que la demostración es totalmente general.

Así que no podemos bifurcar un canal para obtener dos salidas idénticas a una entrada dada. En un computador electrónico este detalle es fundamental, de modo que cualquier computador cuántico debe diseñarse de una manera radicalmente distinta.

Si la clonación fuera posible entonces las correlaciones EPR (véase sección 2.3) podrían utilizarse para transmitir información a velocidad superior a la de la luz. Este teorema por tanto hace consistente aquel resultado con la relatividad especial.

3 Definición de computador cuántico.

3.1. Definición de computador.

Llegado este punto, justo antes de comenzar a hablar sobre qué es o puede ser un computador cuántico, es obvia la necesidad de dejar claro qué es un computador. Tambien es normal que tengamos que aprender un poco sobre la computación en general, y este es el momento de hacerlo. Definiremos:

> *Un computador es un procesador de información de propósito general, es decir, que puede utilizarse para resolver no sólo un único tipo de problema.*

Tal dispositivo sería deseable entonces que sirviera pera resolver la mayor cantidad de problemas posible, y conseguirlo requiere que averigüemos *que problemas* podemos resolver, para más adelante ocuparnos de *cómo* hacerlo.

3.2. Teoría clásica de la computación.

Nuestro enfoque en este momento requiere responder las primeras preguntas que aparecieron cuando los computadores comenzaron a surgir: ¿qué problemas son computables y cuáles no lo son? y ¿cuáles son los recursos necesarios para la computación?

A propósito de la primera cuestión, se dice que una computación es *ineficiente* si mientras el *tamaño del problema* (la cantidad de información necesaria para especificarlo) crece linealmente, los recursos necesarios para resolverlo lo hacen de manera exponencial. Un ejemplo: si la información se codificase utilizando una única palabra:

1 -> 1
2 -> 11
3 -> 111
...

la cantidad de memoria necesaria para guardar los datos crece exponencialmente con la cantidad de información que vayamos a manipular, cosa que no ocurre al utilizar un esquema binario. Ese es el motivo de que en la práctica se elija esta alternativa.

El problema relativo a cómo procesar la información se reduce al tener un cuenta un resultado conocido, que dice que no es necesario manipular todos

los bits de una vez, sino que *es posible realizar cualquier operarión sobre un conjunto de bits manipulándolos de dos en dos.*

3.2.1. Puertas lógicas.

Una puerta *lógica binaria* es un sistema que a partir de dos entradas (x,y) devuelve un valor f(x,y) función de ellas. Hay dos bits de entrada, lo que lleva a cuatro combinaciones posibles en la entrada. A cada una de esas cuatro combinaciones puede responderse con un cero o con un uno. Esto lleva a 16 posibles funciones:

input	f_0	f_1	f_2	f_3	f_4	f_5	...	...	f_{11}	f_{13}	f_{14}	f_{15}
00	0	0	0	0	0	0	...	...	1	1	1	1
01	0	0	0	0	1	1	...	...	0	1	1	1
10	0	0	1	1	0	0	...	...	1	0	1	1
11	0	1	0	1	0	1	...	...	1	1	0	1

Tabla II: todas las posibles funciones combinacionales para entrada binaria.

Concatenando puertas lógicas en diferentes etapas se genera el modelo de red de computador.

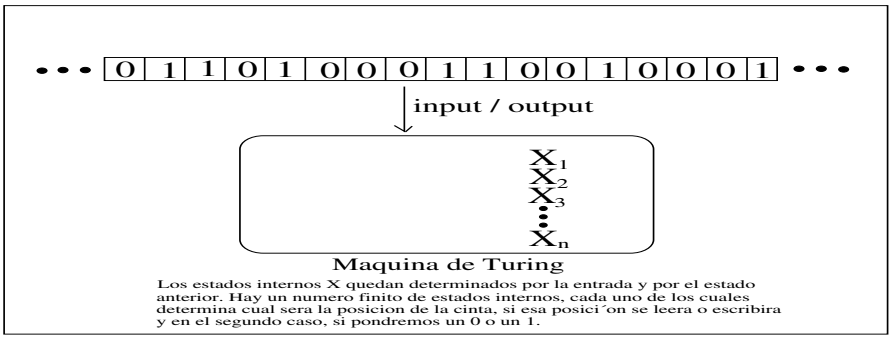
Un computador es un sistema capaz de operar sobre información, transformándola de algún modo. Podemos construir computadores casi con cualquier cosa que se nos ocurra: los dedos, engranajes, válvulas, dispositivos de estado sólido, neuronas, cadenas de ADN o ARN[7]...etc. Unos esquemas funcionarán de modo más eficiente que otros, o más deprisa, pero dejando de lado el computador cuántico, todos estos sistemas están basados en un único modo de trabajar, la *lógica combinacional* ,o en última instancia, la mecánica clásica. En principio, para computar problemas necesitamos un modo de *almacenar y manipular símbolos*. El computador en sí, por encima de esta idea, puede ser abstraído, y no importa realmente (si no fuera por requisitos prácticos) cómo lo construyamos... esto es una cuestión fundamental. El computador cuántico es un esquema interesante porque rompe esta idea. Un cálculo que con los dedos puede realizarse en tres etapas podrá ser realizado tambien en tres etapas (o un número de este orden) en un computador electrónico. Ciertamente un computador electrónico llevará a cabo los pasos que la operación requiere más deprisa, pero esto no incrementa la velocidad del cálculo de modo crítico, ésta sencillamente aumenta liealmente. Un computador cuántico, por contra, es capaz de realizar las mismas operaciones que un computador convencional, pero lo hace de modo diferente. Para algunos tipos de cálculo no aporta ninguna ventaja, pero para otros es capaz de reducir el número de pasos necesarios de manera crítica. De esto y otras cosas va la siguiente sección.

3.2.2. La máquina de Turing.

La *máquina de Turing* se debe al matemático Alan Turing, y es una abstracción sobre lo que puede ser en general un computador clásico. Se trata de una implementación particular (abstracta) sobre lo que es en la práctica todo computador. En cierto sentido se puede decir que desempeña el papel de la máquina de Carnot en termodinámica.

Un computador en general, de los que podemos encontrar en cualquier lugar consiste en esencia en un procesador, capaz de realizar un número limitado de operaciones (lectura/escritura en distintos medios, básicamente) y una memoria direccionable (esto es, con un modo de referenciar la posición en que un dato particular se encuentra, como el índice de un libro). De este modo, el procesador es inicializado de algún modo en un estado particular, y en función de qué encuentre en la posición de memoria actual, y del estado en que se encuentre el mismo realizará una operación de lectura o de escritura en alguna otra posición (o medio), y cambiará de estado. Con esta idea, un conjunto de palabras adecuadamente seleccionado puede conseguir que la máquina así diseñada opere sobre ellas de modo que el resultado tenga algún sentido; esto es la programación.

En el caso de la máquina de Turing la memoria direccionable era sustituída por una cinta que puede moverse de derecha a izquierda y de izquierda a derecha, un número entero de posiciones. No hay ningun otro lugar donde se pueda leer o escribir información. En la posición que queda ante el procesador habrá una palabra (un cero o un uno, no hace falta más) y con esa palabra, más el estado interno en que se encuetre en el instante actual, el procesador es capaz de decidir si escribe o si lee, y a que nuevo trozo de la cinta irá a buscar la siguente instrucción o dato. Si la operación es de escritura entonces tambien decidirá si poner un uno o un cero.



Máquina universal de Turing.

Es requisito que haya un número finito de estados internos. A tales estados los representaré con el número binario d[T]. Turing demostró que existe una máquina U, que cumple:

$$U(d[T], x) = T(x) \tag{3.1}$$

que necesita dar un número de pasos lineal en la longitud de d[T] para simular cada paso de T, donde T es cualquier otra máquina. Es decir, no tenemos ralentización exponencial cuando una máquina de Turing simula el comportamiento de otra. Antes dije que cualquier computador puede realizar la misma clase de operaciones en un número similar de pasos, siempre que se base en el mismo esquema. El esquema decide, como vimos antes, si el problema es en sí computable o no, de modo que para toda una misma clase de máquinas (clásicas o cuánticas, quiero decir) los problemas abordables son los mismos y el tiempo de computación similar. Sobre el grado de abordabilidad de los problemas hablaré en el siguiente apartado, pero debe quedar claro que un problema abordable en una máquina lo es en todas las demás, pues cualquier máquina puede modelizarse con una máquina de Turing, sin cambiar la cantidad de pasos necesarios.Esto se resume en la *conjeturas de Church-Turing:*

> *Cualquier función que pueda ser considerada de "modo natural" como computable puede ser computada por una máquina universal de Turing.*

Si bien la conjetura no ha podido ser demostrada, hasta el momento ha resistido todos los intentos de encontrar un contraejemplo. Veremos en detalle qué ocurre con esto y con el computador cuántico.

3.2.3. Complejidad computacional.

La posibilidad de que una tarea pueda ser llevada a cabo computacionalmente puede analizarse a partir de algún tipo de clasificación. Un modo de hacerlo consiste en definir el *grado de complejidad computacional*. Este parámetro se define como el número de pasos que una máquina de Turing debe dar para resolver el problema. En el modelo de red, del que hablé antes por encima el grado de complejidad una medida del número de etapas que debe atravesar una señal de entrada hasta llegar a la salida en forma de respuesta. La complejidad introduce las clases de problemas:

1. Problemas de clase **p**: El algoritmo existe, y es función polinómica de L, el tamaño de la entrada (cantidad de información que especifica el problema).Un problema de tipo **p** se considera abordable computacionalmente.

2. Un problema es de clase **np** cuando podemos encontrar su solución en tiempo que crece polinómicamente con L, pero con muchos más pasos. Aunque es obvio que un problema de clase **p** lo es tambien de clase **np**, la recíproca no ha sido demostrada.

3. Problemas no computables por tiempo de cálculo: La imposibilidad de resolverlos algorítmicamente se deriva del hecho de que el número de etapas que requieren crece exponencialmente con el tamaño de la entrada.

4. Problemas esencialmente no computables. Aquí la situación es aún peor. Se comprueba que para ciertos casos *no existe* ningun algoritmo capaz de llegar a una solución. El ejemplo por excelencia es el *problema de la detención*. Hablaré del problema en en siguiente apartado.

3.2.4. El problema de la detención.

Todos los que hemos programado unas pocas líneas de código sabemos lo que es un bucle, y con toda seguridad tambien nos habremos encontrado con algún bucle sin fin. En un bucle sin fin el computador queda atrapado en una secuencia de estados que se repite cíclicamente, bloqueado por alguna condición que suele corresponder a un error de programación.

Un problema interesante relacionado con la detención de una máquina se formula de esta manera:

Supongamos que un computador se ha preparado para que siga estas instrucciones: " Si x es igual a la suma de dos números primos añadele 2. Si no lo es, imprime el valor de x y para. Comienza por x=8"

Un algoritmo como este permite obtener números primos de manera sistemática. Esto, como vimos antes, puede ser muy útil. Pero no es el propio algoritmo el que nos interesa ahora, sino la pregunta *¿se detendrá alguna vez un algoritmo como este?*. Bien, si se detuviera, la conjetura de Goldbach tendría un contraejemplo. Sin embargo, la conjetura de Goldbach no deja de ser una conjetura. Si supiéramos que es estrictamente cierta, entonces sabríamos que el algoritmo no se detendrá nunca. Por contra, si no se cumpliese siempre, entonces el algoritmo se detendría... y aún no sabemos cual de ambos casos se da. Conclusión: no sabemos si un algoritmo como este se detendrá alguna vez. Contado así parece que se trata de una simple cuestión de profundizar en el estudio, pero no es el caso. El problema de la detención es esencial.

Si dispusiéramos de un modo de decidir si un algoritmo se detendrá o no, entonces seríamos capaces de resolver problemas como el anterior. Bueno, según Steane[1], resolveríamos todas las matemáticas.

Entonces supongamos que tal algoritmo existe, y formalmente lo describiríamos diciendo que es capaz de decir si una máquina de Turing se detendrá o no en función de la entrada. El problema:

> "Dados x y d[T](esto es, la descripción de la máquina), ¿se detendrá una máquina de Turing T que reciba x como entrada?"

Si existe un algoritmo capaz de tal cosa entonces es posible crear una máquina de Turing T_H de modo que se detenga sólo en el caso de que T(D[T]) no lo haga. La nueva máquina recibe como entrada precisamente d[T], que incluye la descripción de T y de su entrada.

$$T_H(d[T]) sedetiene \iff T(d[T]) nosedetiene \tag{3.2}$$

Pero... hay un pero. ¿Cómo hace T_H para describirse a sí misma? Si utiliza d[T_H], entonces:

$$T_H(d[T_H])sedetiene \Longleftrightarrow T_H(d[T_H])nosedetiene \tag{3.3}$$

una contradicción. Concluímos que no existe ningún método general para saber si una máquina de Turing se detendrá o no. El problema de la detención no es computable. Este resultado está directamente conectado con el teorema de Godël: las matemáticas no pueden reducirse a una única idea fundamental, hay diferentes resultados que no se pueden conectar unos con otros.

3.2.5. Criptografía RSA.

Un ejemplo habitual de problema que no es computable es la descomposición en factores primos. Un número que no sea primo se puede expresar como producto de números primos, naturalmente más bajos. Podemos ir dividiendo el número que tenemos que factorizar sucesivamente por los enteros menores que él, hasta encontrar uno que de resto cero. Procederíamos entonces de este modo con los factores encontrados, y así sucesivamente, hasta que el problema quede resuelto. Esto suena muy bien, hasta que nos enfrentamos a la posibilidad de que los factores primos sean tambien números grandes. Los mejores métodos conocidos en la actualidad requieren 42 días a 10^{12} operaciones por segundo para factorizar un número decimal de 130 dígitos, pero si duplicamos L el tiempo aumenta en un factor 10^{25}, es decir, un millón de años.

El hecho de que la factorización sea un problema intratable ha sido utilizado como base para los sistemas criptográficos modernos, tales como el RSA (Rivest, Shamir y Adleman). Para cualquier mensaje, podemos obtener fácilmente una versión encriptada

$$E = M^s mod(c) \tag{3.4}$$

donde s y c son dos números enteros grandes, que pueden darse a conocer. El modo de desencriptar el mensaje consiste en calcular

$$M = E^t mod(c) \tag{3.5}$$

donde el valor de t puede obtenerse sin problemas a partir de s y de los factores de c (Schroeder, 1984). Se suele utilizar c=pq, donde p y q son dos números primos grandes, conocidos sólo por quien publicó el valor de c. El usuario que conoce p y q es el único que puede leer los mensajes, mientras que cualquier otro puede enviárselos de manera segura.

3.3. Teoría cuántica de la computación.

Para esta sección he reservado las generalizaciones sobre computación clásica que quedaron pendientes.

3.3.1. El principio de Church-Turing y el QC.

El principio de Church-Turing era la generalización del de Turing, en la que no se exigía que la máquina que realizase las computaciones fuese en particular una máquina de Turing, sino una de tipo más general (véase sección 2.2.2). Nos preocupamos ahora de si un QC es o no un computador universal, en el sentido de si verifica este principio. Utilizaré el siguiente esquema para justificar tal cosa:

1. El estado de cualquier sistema cuántico finito viene representado por un vector de estado en un espacio de Hilbert. De esta forma, un número finito de qubits tambien puede representarlo con un nivel arbitrario de precisión.

2. La evolución de cualquier sistema cuántico viene dada por una transformación unitaria del estado. Un computador cuántico puede simular cualquier transformación unitaria con la precisión que queramos, de modo que tambien podrá representar la evolución de un sistema cuántico.

Nos encontramos una dificultad (Myers, 1997) para aquellas tareas en las que no podemos decidir en cuantos pasos se llevará a cabo la operación. Para un computador cuántico no tenemos un modo general de decidir si se ha detenido, cosa que no supòne un problema en el computador clásico. La solución (Deutch, 1985) aparrece cuando sólo tenemose en cuenta operfaciones para las que sabemos de antemano el númenro de pasos, o para las que hemos dedicadao un qubit a señalarnos que la operación se realizó ya. Nielsen y Chuang (1997) demostraron que no existe un conjunto de puertas fijadas que actuando sobre loos datos y sobre el programa sea capaz de realizar cualquier operación, pero a nosotros nos interesa un computador donde un computador clásico comprola qué puertas son aplicadas sobre un registro cuánticos.

3.3.2. Procedimientos cuánticos.

Sabemos que un computador clásico es capaz de simular el comportamiento de los sistemas cuánticos, de modo que es razonable tener dudas sobre si la computación cuántica es capaz de aportar algo más que la clásica. La clave ya comenté antes que está en el grado de complejidad computacional, que si bien puede permitir plantear un problema en términos de computación clásica, puede impedir que éste se resuelva en un tiempo razonable. El nivel de complejidad que un computador cuántico es capaz de afrontar para determinados problemas es mucho más alto que el clásico para máquinas de la misma escala.

Existen problemas que ya sabemos cómo resolver de un modo más eficiente por medio de un QC, como son los de factorización y búsqueda, de los que hablaré en el siguiente capítulo. Hay además recursos que disponibles que

hacen pensar en posibilidades muy importantes, asociadas a la posibilidad de una fuerte paralelización de las operaciones.

Personalmente, prefiero el término *procedimiento* frente al de *algoritmo* a la hora de referirme a lo que un procesador cuántico de la información es capaz de hacer, pues a mi entender el segundo término está habitualmente asociado a la idea de conjunto de operaciones que se realiza secuencialmente a partir de un estado inicial, manipulando información completamente determinada en cada etapa. El término procedimiento tal vez sea un sinónimo de algoritmo, pero no suele ser utilizado del mismo modo, y me permite dejar la puerta abierta a un tipo de operaciones más general, en el que tal vez la información se manipula de otra manera, al tiempo que ésta puede tener otro aspecto mientras se opera con ella.

4 Los problemas que resuelve el computador cuántico.

De entrada hemos descubierto que el computador cuántico no es aquella panacea capaz de resolver todos los problemas que se nos ocurran. Existe aquel grupo de problemas que es irresoluble por naturaleza, como el de la detención (ver sección 3.2.4). Sin embargo, tenemos una ventaja enorme cuando hablamos de mecánica cuántica: El espacio de los estados crece exponenciamente con el número de qubits, mientras que lo hace linealmente con el número de bits. Esto se debe a que si bien n bits pueden combinarse de 2^n maneras diferentes, una combinación de n qbits admite todas las combinaciones lineales posibles de vectores de estado, cada uno de los cuales supone a su vez las 2^n combinaciones.

Mejor que hablar tanto será tratar de ilustrarlo con un ejemplo. Lo natural será escoger el caso más sencillo: 2 qubits frente a dos bits. Los estados posibles fruto de la combinación de 2 bits son:

$$x_1 x_2 = 00; 01; 10; 11$$

es decir $2^2 = 4$ estados. Ahora veamos qué ocurre si disponemos de 2 qubits. En principio parece evidente que disponemos de estos estados:

$$\{|q_1, q_2>\} = \{|0,0>; |0,1>; |1,0>; |1,1>\} \tag{4.1}$$

pero esto es sólo la base. Esto es, *la dimensión del espacio* ha crecido exponencialmente con el número de bits. En este espacio es posible encontrar toda clase de combinaciones de elementos de la base:

$$|\phi>= a|q_1, q_2> + b|q_1', q_2'>$$

siempre que esté normalizada. Si hablamos de partículas idénticas estos estados deben ser de simetría bien definida. Es habitual trabajar con electrones (fermiones), por lo que los estados posibles serán antisimétricos, o fotones (bosones) para los que los estados posibles son simétricos. Los qubits, desde un punto de vista intuitivo da la impresión de que son capaces de barrer muchos casos posibles con una cantidad de recursos que no sería ni de lejos suficiente para un computador electrónico. Esto, que he dicho de un modo tan informal, veremos que se traduce en hechos, como los métodos de Shor

y de Grover. Por un lado, el primero es capaz de encontrar la descomposición en factores primos de un número en un tiempo que crece linealmente con el tamaño del número a factorizar, aprovechando el crecimiento exponencial de la dimensión del espacio con el que se trabaja. En un computador clásico el tiempo vimos que crece exponencialmente con el tamaño de la entrada. El algoritmo de búsqueda de Grover, por otra parte, tiene propiedades similares en lo que se refiere a buscar elementos determinados en listas grandes, donde los tiempos de búsqueda son tambien mucho mayores en los computadores clásicos. El apartado titulado "aplicaciones a la inteligencia artificial" surge debido a esta misma intuición y es enteramente personal. En él, a partir del hecho intuitivo que he mencionado, pretendo concluir que es posible encontrar métodos para resolver problemas tales como las búsquedas de soluciones en problemas como los juegos, o el análisis de situaciones en las que se abren abanicos de posibilidades, y trato de encontrar algún método.

En principio, la conclusión es que la mecánica cuántica es un herramienta muy poderosa de aceleración de cálculos, y poco más. Pero esto no es trivial, desde el momento que vemos que esta aceleración puede ser exponencial, y por tanto saca problemas del dominio irresoluble al dominio p. El entrelazamiento hace posibles algunas otras cosas tambien. Estas son algunas aplicaciones interesantes.

4.1. El método de factorización de Shor.

En la sección 3.2.5 señalé cómo los métodos modernos para descomponer números grandes en factores primos son considerados ineficientes, y no permiten en general obtener descomposiciones en tiempos razonables. Este problema, aún bastante particular, resulta de gran interés, y tiene solución conocida en el campo de la computación cuántica. Existen otros muchos problemas donde un QC es más eficiente que un computador clásico, la mayoría de ellos sin descubrir. Esto constituye un campo de búsqueda activo.

Ocupémonos de los algoritmos de *búsqueda del periodo de una función* y, como caso particular, *descomposión en factores primos.*

4.1.1. Búsqueda del periodo de una función.

Partamos de la función $f(x)$, cuyo periodo es r:

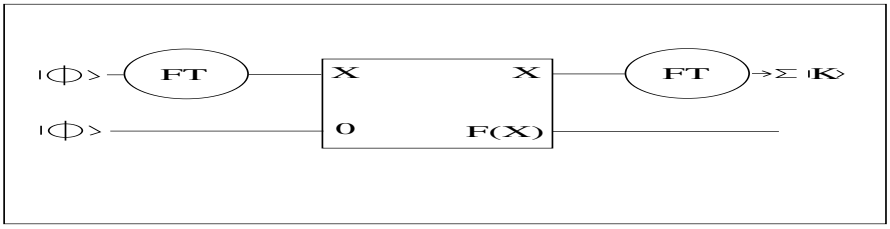
$$f(x+r) = f(x)$$

Partamos de dos suposiciones adicionales: $f(x)$ puede computarse eficientemente en x, y sabemos que r cumple:

$$\frac{N}{2} < r < N$$

para un N dado. En un 9computador clásico, en ausencia de un método analítico para hallar el periodo de la función, lo mejor que podemos hacer es evaluar $f(x)$ en alrededor de N/2 puntos, y buscar dónde comienzan a repetirse los resultados. El número de operaciones crece exponencialmente con log N, que es la información necesaria para especificar N.

En la siguiente figura muestro el modo en que un QC resolvería el problema:



Dispositivo para ejecutar el algoritmo de Shor.

Los requisitos para ejecutar el algoritmo son 2n qubits, más del orden de n más para almacenamiento intermedio (espacio de trabajo), con n=$\lceil 2 \log N \rceil$, donde el símbolo $\lceil ... \rceil$ significa "el entero inmediatamente superior" al argumento.

1. Utilizaremos dos registros de n qbits cada uno, que llamaremos **x** e **y.** Prepararemos ambos registros en el estado inicial |0>.

2. Aplicaremos la operación H (transformada de Fourier) a cada qubit del registro **x**. El estado obtenido:

$$\frac{1}{\sqrt{\omega}} \Sigma_{x=0}^{\omega-1} |x> |0>; \omega = 2^n \tag{4.2}$$

   donde |x>significa, por ejemplo, |0011001>, con 0011001 la representación binaria de **x**. Denominamos a {|0>,|1>} la *base computacional.*

3. Hacemos entonces pasar los registros **x** e **y** por una red de puertas de modo que se efectúe la transformación

$$U_f \left(\frac{1}{\sqrt{\omega}} \Sigma_{x=0}^{\omega-1} |x> |0> \right) = \frac{1}{\sqrt{\omega}} \Sigma_{x=0}^{\omega-1} |x> |f(x)> \tag{4.3}$$

   Este proceso es reversible, dado que el estado del miembro derecho está biunívocamente determinado por el del miembro izquierdo, de modo que U_f puede ser una transformación unitaria.

4. Hemos obtenido el valor de $f(x)$ para $\omega = 2^n$ valores de una sola vez. Esto se conoce como *parlelismo cuántico.* La dependencia con n es exponencial, de modo que el grado de paralelismo es enorme. Con sólo n=100 tenemos $2^{100} \simeq N_A$ procesadores clásicos.

5. Nos enfrentamos al inconveniente de no tener un modo directo de alcanzar los valores almacenados en el estado 4.3. Un modo de obtener información es medir los estados del registro **y**, donde almacenamos $f(x)$, pero eso sólo nos dará un valor de f, debido al colapso del estado en el subespacio correspondiente al autovalor medido. Imaginemos que hemos medido el registro **y**, y obtuvimos $f(x) = u$. Entoces todo el registro **y** colapsará en el estado |u>, asociado a un valor determinado para todas las componentes del registro. Pero sabemos que en el registro hay información sobre 2^n evaluaciones de f. El estado total sería:

$$\frac{1}{\sqrt{M}}\Sigma_{j=0}^{M-1}|d_u + jr > |u >$$

donde $d_u + jr$, con j=0,1,2...M-1 son todos los valores de **x** para los que $f(x) = u$. El periodo de $f(x)$ conlleva que en el registro **x** aparece una superposición de M$\simeq \omega/r$ estados, con valores de x separados un periodo r. El offset es d_u, y depende del valor de u obtenido al medir el registro **y**.

6. Lo único que queda es extraer la periodicidad del estado contenido en el registro **x**. Esto se hace directamente, haciendo la transformada de Fourier del estado, y midiendo a continuación. La transformada de Fourier discreta es la siguiente operación:

$$U_{FT}|x >= \frac{1}{\sqrt{\omega}}\Sigma_{k=0}^{\omega-1}e^{i2\pi kx/\omega}|k > \tag{4.4}$$

Conviene observar en este momento que la operación 4.2 es un ejemplo de transformada de Fourier, como dije antes sin justificar, donde se actúa sobre el estado |0>. Hemos supuesto que r es un divisor de ω, de modo que M=ω/r es una división exacta. Esta simplificación puede hacerse innecesaria (Shor 1994, 1995a, Ekert y Josza 1996).

7. En lo sucesivo no nos interesa lo que haya almacenado en el registro **y**. La aplicación del operador U_{FT} sobre el estado del registro **x**:

$$U_{FT} = \frac{1}{\sqrt{\omega/r}}\Sigma_{j=0}^{\frac{\omega}{r}-1}|d_u + jr >= \frac{1}{\sqrt{r}}\Sigma_k\widetilde{f}(k)|k >$$

donde

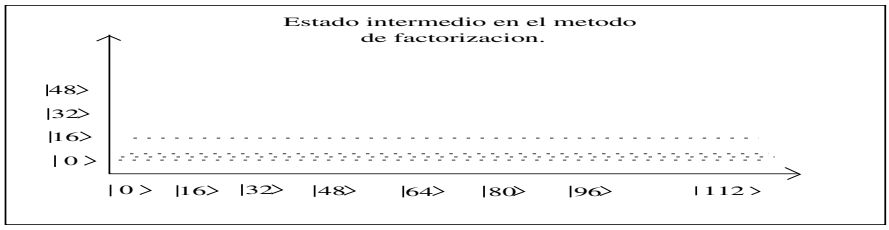
$$|\widetilde{f}(k)| = \begin{cases} 1 & ; \quad k \quad es \quad múltiplo \quad de \quad \omega/r \\ 0 & ; \quad los \quad demás \quad casos \end{cases}$$

8. Ahora queda obtener el valor de r (el periodo) a partir del resultado. Sabemos que x=$\lambda\omega$/r, con λ desconocido. Si λ y r no tienen factores comunes podremos despejar x/ω , que será una fracción irreducible, y a partir de ella y de x, obtener tanto r como λ. Si λ y r tienen algún factor común, cosa poco probable para valores grandes de r, el algoritmo falla, y debemos

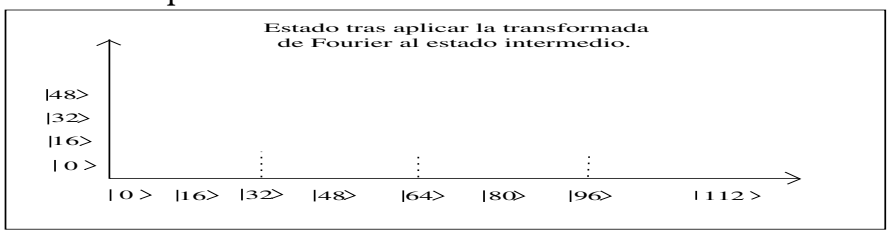
repetir todos los pasos desde el principio. Tras un número de repeticiones del orden de log r la probabilidad de obtener el resultado correcto se hace tan alta como queramos (Ekert y Josza 1996).

Observamos que el límite de eficiencia del método de Shor viene dado por la evaluación de $f(x)$. Por contra, el número de puertas lógicas requeridas para la búsqueda del periodo crece polinómicamente en n, en lugar de hacerlo exponencialmente.

Los pasos intermedios vienen representados en estas figuras:



Primera etapa



Segunda etapa

4.1.2. Factorización de enteros grandes.

El problema de la factorización puede reducirse al que acabamos de resolver en el apartado anterior. Ahora la función cuyo periodo buscamos es:

$$f(x) = a^x mod(N) \tag{4.5}$$

donde N es el número a factorizar y a<N se elige aleatoriamente. A partir de teoría de los números (Ekert y Josza 1996) puede demostrarse que el periodo r es un número par, y que $a^{r/2} \pm 1$ tiene un factor común con N. Así encontramos los factores que estamos buscamos. A partir de la cantidad r encontramos el factor común $a^{r/2} \pm 1$ por medio del algoritmo de Euclides.

Sabiendo esto, el modo de proceder consiste en hallar raíces admitiendo resto hasta N dando potencias $(((a^2)^2)^2...)^2$ seleccionando así las potencias de correspondientes a la representación binaria de a, para finalmente multiplicarlas. Las redes para ejecutar los algoritmos de búsqueda de factores primos se han descrito (por ejemplo, Micuel et. al. 1996) y se sabe que requieren del orden de 300(log N)^3 puertas.

Para números con 130 dígitos encontramos que un computador cuántico no aporta ventajas sobre uno clásico (unas 7 horas con una tasa de conmutación

del orden de un MHz), pero con 260 dígitos un computador que ejecutara el algoritmo anteriormente descrito tardaría 8 veces más, mientras que para un computador clásico el problema se hace intratable.

El método de Shor hace uso de toda la potencia de la mecánica cuántica: *entrelazamiento*, *interferencia* y *paralelismo cuántico*. Interesa observar cómo la transformada de Fourier 4.4que aplicamos puede considerarse como permitir que se produzca interferencia entre los estados superpuestos en el registro **x**.

4.2. Codificación superdensa.

La idea de la *codificación superdensa* es aprovechar el entrelazamiento como una fuente de información. A primera vista, lo natural parece que es utilizar los qubits como si se tratara de bits convencionales, enviando la información por medio de combinaciones de ellos. Pero los qubits permiten mucho más.

La forma simple de enviar la información no requiere explicación: si Alice quiere hacer llegar a Bob la secuencia {1101011}, enviará siete qubits preparados en el estado | 1101011>. Un bit, un qubit.

Pero Alice y Bob podrían compartir desde el principio un par entrelazado de qubits, preparados en el estado

$$\frac{1}{\sqrt{2}}[|00> + |11>]$$

Este par procede de alguna fuente intermedia. Ahora Alice puede enviar dos bits con un único qubit (Benett y Wiesner, 1992). Observemos el siguiente hecho: Tenemos cuatro estados ortogonales entre sí, que dan lugar a lo que se conoce como base de Bell:

$$|s_1> = \frac{1}{\sqrt{2}}[|00> + |11>]$$

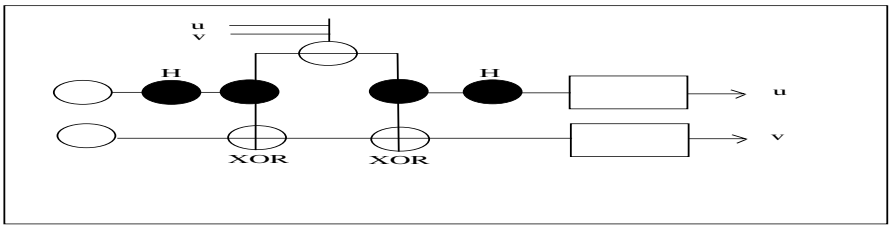
$$|s_2> = \frac{1}{\sqrt{2}}[|00> - |11>]$$

$$|s_3> = \frac{1}{\sqrt{2}}[|01> + |10>]$$

$$|s_4> = \frac{1}{\sqrt{2}}[01> - |10>]$$

Estos estados pueden obtenerse uno a partir del otro por medio de operaciones sobre un único qubit. Para hacerlo, Alice no necesita más que efectuar una de las operaciones {I,X,Y,Z}, vistas en la sección 2.4.2. Hay cuatro operaciones posibles. Una cantidad de información que puede tomar cuatro valores diferentes equivale a dos bits, pero efectuamos operaciones sobre un único qubit. Así Alice puede enviar dos bits manipulando un único qubit.

Bob necesita ahora saber que información le ha sido enviada. Para ello hará actuar la puerta XOR sobre las parejas recibidas. Al medir el resultado se encontrará con dos posibilidades, (|00>±|11>)/$\sqrt{2}$ o bien (|01>±|10>)/$\sqrt{2}$. Para averiguar el signo de la superposición Bob hará actuar el operador de Hadamard sobre el estado final.En la figura presento el diagrama asociado a este proceso:



Red para codificación superdensa.

El tiempo corre de izquierda a derecha. La codificación superdensa es difícil de realizar en la práctica, pero es la base de un método de comunicación seguro, dado que el qubit enviado por Alicia sólo es interpretable para el poseedor del otro elemento de la pareja.

4.3. Teletransporte cuántico.

El *teletransporte* es en esencia lo que su propio nombre indica. Puede no ser necesario enviar un qubit para hacer llegar información de un punto a otro.

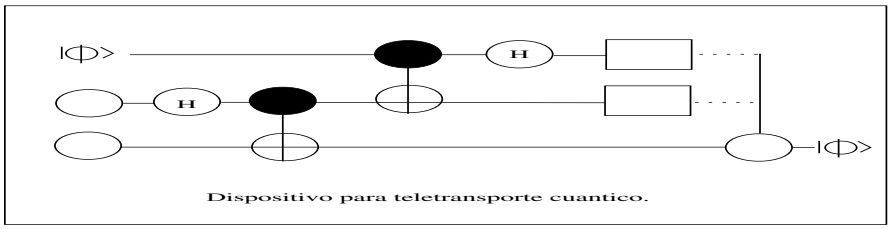
Alice está interesada en hacer saber a Bob el valor de un qubit particular, digamos |ϕ>=|0>, que ella conoce. No necesariamente hay que hacer llegar el qubit |0>hasta Bob. La posibilidad que pasa por medir antes el qubit, de todas formas, en caso de que éste fuera desconocido para Alice, y de enviar despues la información destruiría el estado inicial. Y ya sabemos que no se puede copiar un estado que no es conocido. Así que Alice siempre conoce el estado del qubit.

El *teletransporte cuántico* (Bennett et. al. 1993, Bennett 1995) utiliza el entrelazamiento para resolver estas dificultades.

> Supongamos que Alice y Bob comparten un par entrelazado en el estado (|00>+|11>)/$\sqrt{2}$. Alice pretende transmitir a Bob un qubit en un estado *desconocido.* Este estado será representado como
>
> $$|\phi>= a|0> +b|1>$$
>
> El estado inicial de los tres qubits será:
>
> $$[a|0> +b|1>] \otimes \frac{1}{\sqrt{2}}[|00> +|11>] =$$

$$= \alpha|000> + \beta|100> + \alpha|011> + \beta|111>$$

desde luego, normalizado. Alice mide en la base de Bell (apartado anterior) los primeros dos qubits, que son aquél que es en principio desconocido, y que se desea transmitir, y su parte del par entrelazado. Esto se puede hacer mediante el esquema de la figura:



Dispositivo para teletransporte cuantico.

Red para Quantum Teleporting.

Primero Alice aplica las operaciones XOR y de Hadamard, y despues de esto el estado resultante es :

$$|00>(\alpha|0> + \beta|1>) + |01>(\alpha|1> + \beta|0>) +$$

$$+|10>(\alpha|0> - \beta|1>) + |11>(\alpha|1> - \beta|0>)$$

Inmediatamente despues mide sus qubits. De acuerdo al postulado de la medida, el estado inmediatamente a continuación de ésta es el colapso sobre uno de los cuatro estados de la base de Bell, que contiene dos bits de información. Enviamos estos dos bits a Bob, que con ellos será capaz de decidir que operación {I,X,Y,Z} debe aplicar a su qubit, para pasarlo al estado $|\phi>$=a|0>+b|1>. Así Bob ha sido capaz de recuperar el qubit sin que éste fuese en sí transmitido.

No es posible clonar un estado que no se conoce, y no se ha podido hacer llegar a Bob el qubit sin que Alice lo perdiese. Por otra parte, $|\phi>$contiene información completa sobre el estado del qubit de Alice, de modo que no se ha perdido información. De estos dos hechos se deriva que el término *teletransporte* sea adecuado para esta situación.

4.4. El algoritmo de búsqueda de Grover.

Otra aplicación de la potencia de la mecánica cuántica en la resolución de problemas computacionalmente pesados es la búsqueda de elementos en listas. El método es una variante del de búsqueda del periodo de una función. Nos ocupamos de listas desordenadas, que es donde un sistema clásico de búsqueda no tiene más remedio que recorrerlas de algún modo buscando el

elemento por medio de repetidas comparaciones. Existen diferentes alternativas, que mejoran la eficiencia de las búsquedas, pero cada una de ellas parte de situaciones particulares en las que rentabiliza el número de operaciones. Un ejemplo relativo a la seguridad informática: una lista de contraserñas habitualmente se almacena encriptada en algún archivo de sistema, de modo que cuando cualquier usuario teclea su contraseña esta se encripta de nuevo (con un coste computacional insignificante) y se compara con la versión encriptada de la lista. Desencriptar una contraseña codificada por ejemplo vía RSA, es actualmente un problema inabordable cuando la clave es suficientemente compleja. Las personas que pretenden acceder a un sistema protegido de esta manera sin permiso habitualmente optan por hacer un ataque por *fuerza bruta*, esto es, recorrer todas las combinaciones posibles de caracteres una por una hasta encontrar una que sea una contraseña. Esta búsqueda a menudo se enfoca de un modo diferente: teniendo en cuenta que muchas personas utilizan palabras con sentido en algún idioma, utilizan *diccionarios*, que no son otra cosa que bases de datos de palabras en algún idioma, que lleva mucho menos tiempo recorrer, de modo que por regla general el coste computacional de romper la seguridad de un sistema informático puede disminuir considerablemente. *Si* estas condiciones se rompen el algoritmo puede hacerse muy ineficiente. De esta forma, si ningun usuario utiliza una clave basada en lenguaje natural el intruso recorrerá el diccionario completo sin obtener resultados satisfactorios.

Del ejemplo anterior extraemos la siguiente conclusión: si no sabemos nada sobre una lista no hay motivo para escoger un enfoque en lugar de otro, salvo el de maximizar las posibilidades de éxito. Esto incluso conlleva hacer más pesada la computación, pues nos obliga a hacer todas las comparaciones posibles.

Desde la perspectiva de la mecánica cuántica podemos utilizar el algoritmo presentado por Grover en 1997. El problema que resuelve puede representarse del siguiente modo:

> *Partimos de una lista desordemada* $\{x_i\}_{i=1}^{N}$*, en la que tratamos de localizar un elemento particular,* $x_j = t$*.*

Si bien un algoritmo clásico, recorriendo una lista de N elementos requiere en promedio realizar N/2 comparaciones, el método de Grover necesitará sólo hacer $\sqrt{N}$. Bennett demostró que esto es lo mejor que se puede hacer (Bennett et. al. , 1997). El método no supone trasladar el problema a una nueva clase en el sentido de peso de computación, pero sí supone una aceleración tanto más significativa cuanto mayor sea la lista.

Aquí describo el algoritmo paso a paso:

1. Suponemos cada elemento de la lista etiquetado con un índice i, como expresé en el planteamiento del problema. Suponemos tambien que hay

una operación unitaria que permite saber si el elemento actual es el que estamos buscando. El operador aplicado se denotra por S:

$$S|i>=|i>; i \neq j$$

$$S|i>=-|j>; i = j$$

donde j representa el índice del elemento buscado. Observemos que una estrategia para la resolución de problemas pesados es hacer una búsqueda aleatoria donde por comparación tratamos de determinar si ciertop elemento es una solución al problema. Esto es el ejemplo que presenté en la introducción, o puede ser la búsqueda de la solución de una ecuación diferencial suponiendo buen comportamiento de las funciones implicadas. Estos problemas vimos antes (sección 3.2.3) que pertenecen al dominio NP.

2. Al igual que en el método de búsqueda del periodo, inicializamos un registro en una superposición de estados:

$$|\psi(\theta)>=\sin\theta|j>+\frac{\cos\theta}{\sqrt{N-1}}\Sigma_{i\neq j}|i> \tag{4.6}$$

los índices siguen correspondiendo a lo indicado en el paso (1). Este estado superpone con igual peso todos los elementos de la lista, pues partimos de $\theta_0 = 1/\sqrt{N}$. No decimos nada, sólo preperamos un registro donde todos los elementos están igualmente representados, tanto el que buscamos como los demás. De hecho, como el que estamos buscando es en principio un elemento cualquiera, simplemente hemos construido el estado $|\psi>=(1/\sqrt{N})\Sigma_k|k>$, donde los vectores |k>barren toda la lista. Nótese que el subíndice indica estado *inicial*.

3. A continuación aplicaremos el operador unitario S, que invertirá el signo del elemento que estamos buscando.

4. Por último, aplicamos la transformada de Fourier (4.4) al estado resultante, lo que invierte el signo de todas las componentes excepto |0>. Entonces aplicamos de nuevo la transformada de Fourier. El efecto de esta secuencia de transformaciones, expresado en un único operador:

$$U_G|\theta>=|\psi(\theta+\phi)> \tag{4.7}$$

donde $\sin\phi = 2\sqrt{N-1}/N$.

5. Encontramos que el coeficiente del elemento que buscamos es ligeramente mayor que el de los demás elementos de la superposición.

6. Sabiendo lo anterior, aplicaremos el operador U_G m veces, con $m = (\pi/4)\sqrt{N}$. Poco a poco el ángulo θ se va acercando a $\pi/2$, lo que hace cada vez más importante el coeficiente de |j>en la superposición, es decir, cada vez nos acercaremos más al elemento |j>.

7. Tras m iteraciones la probabilidad de error al medir el registro es del orden de N^{-1}.

Sólo hay un problema: aplicar demasiadas veces la tranformación U_G disminue la probabilidad de éxito. Para evitarlo debemos conocer m, es decir, el tamaño de la lista.

4.5. Aplicaciones a la inteligencia artificial.

Imaginemos que buscamos la salida de un laberinto. Un laberinto es precisamente un juego. Se puede resolver con un número determinado de jugadas, que son decisiones sobre que pasillo escoger en cada bifurcación, y hay un objetivo claro: encontrar la salida. Imaginemos que utilizamos un ratón. Sabemos que cuando está debidamente entrenado, un ratón es capaz de encontrar la salida, en un laberinto de un tamaño que cabe en la mesa de un laboratorio. Pero una partida de ajedrez permite en cada movimiento un número enorme de movimientos (pasillos). A su vez, los movimientos del oponente están por determinar, y el objetivo, pese a ser accesible desde varios caminos, requiere un enorme esfuerzo computacional. Un ratón, despues de todo, es capaz (no haré mayores suposiciones) de recorrer todos los pasillos y dar la vuelta cada vez que no puede continuar. Un computador electrónico hace en esencia lo mismo, salvo cuando incorpora heurísticas que pueden hacer más eficientes las búsquedas.

Imaginemos ahora un laberinto enormemente complejo. Hablo de un nivel de complejidad tal que un ratón que pretendiese encontrar la salida moriría por el camino. Esto es, para mí, un problema inabordable con métodos de tipo paso-a-paso. Pero imaginemos que la superficie del laberinto está inclinada, y que lo llenamos con agua. El agua saldría en poco tiempo por el otro lado. Más deprisa encontraría la salida un gas, empujado por una depresión en la salida. Pero no sabemos preguntarle a estos sistemas por dónde han pasado. No ocurre lo mismo con un sistema cuántico. Baso esta afirmación en la idea de *paralelismo cuántico* que mencioné en la sección 4.1.1. Podemos preparar un estado inicial de alguna manera que a cada paso almacenara información sobre que alternativa escogió. Esto suena familiar si recordamos el experimento de Stern-Gerlach. Los electrones con un spin determinado son filtrados, mientras que los de spin perpendicular llegan sin problemas al detector. Podemos imaginar que hay una manera de filtrar unos estados de una combinación inicial, en función de la alternativa escogida. Así, podríamos preparar un estado inicial, que se propagase a través de una serie de filtros, que tendrían sobre él el efecto de una medida, proyectándolo sobre un subespacio determinado. Adicionalmente, la propia implementación del problema incluiría la descripción de cuándo dejar de permitir que uno de los estados se propagase, con lo que al final sólo recibiríamos partículas en un estado dotado de información

sobre las trayectorias que resuelven el problema. El estado de llegada habrá excluído los autovalores diferentes a los escogidos en las etapas de decisión, y una medida de estos operadores de nuevo nos dirá todo lo que queremos saber. No nos preocupa modificar el estado del sistema, porque se trata de la última etapa, salvo en el caso de que queramos saber varias cosas, por haber usado varios filtros. Si ese es el caso podemos buscar observables asociados a subespacios independientes. En función de la tecnología de que dispongamos seremos capaces de utilizar distintas clases de filtros, y en función de éstos será conveniente limitar el número de etapas.

Volvemos, si se quiere ver de esa manera, a resolver problemas de clase np por medio de búsqueda de soluciones en lugar de intentar calcularlas y, como era de esperar, la obvia referencia al método de búsqueda de Grover. Sin embargo, me parece buena idea observar que tal método no sólo sirve para buscar raíces a ecuaciones ni números de teléfono, sino que permite tomas de decisiones para problemas complejos en tiempos muy cortos.

4.5.1. Juegos de un movimiento.

Supongamos que tratamos precisamente de resolver el problema del laberinto, y digamos que en cada nudo sólo hay dos posibilidades: girar a la derecha o girar a la izquierda. En este caso hay un árbol (binario) de decisiones para resolver el problema, y encontrar la salida consiste únicamente en encontrar la rama correcta y seguirla desde el principio hasta el final. Una posible manera de etiquetar las estrategias posibles es implementando en unarray un número alto de decisiones, de modo que sea poco probable quedarnos a la mitad. Me explico: supongamos que el laberinto mide cien metros de largo por cien de ancho, y sabemos que cada puerta tiene un ancho de al menos un metro. Esto, si hubiese puertas en todas las posiciones no permite que el número de éstas exceda de las diez mil. Un análisis más detallado haría bajar significativamente el número de puertas que esperamos etner que atravesar para encontrar la salida. El vector que utilizaremos para encontrar la salida sobreestimará el número de decisiones, de modo que no permitirá que la rama correcta del árbol termine en una puerta del interior del laberinto, mientras que todas las decisiones sobrantes serán desestimadas. Incluso, por medio de interferencia con los estados finales podríamos llegar a decidir en cuántas etapas encontró la salida el computador.

Dada esta situación, podemos tomar la siguiente tabla de estrategias:

000...001	Toma la primera desviación a la derecha y el resto a la izquierda.
000...010	Gira a la derecha sólo en el segundo nodo.
000...011	A la dcha. en los 2 primeros nodos, a la izq. en los demás.
...	...
111...111	Gira a la derecha en todos los nodos.

Una base de qubits adecuada (tal vez no la mejor) puede ser

$$\{|000..,001>;|000..,010>;|00..,100>;...|1000..,0>\} \tag{4.8}$$

Introduciremos la definición del laberinto en la computadora de tal modo que cuando un camino se cierra, y con ello una rama del árbol, los autoestados que llegaron hasta ahí son filtrados.

En esta situación, bastará inicializar el array en un estado que sea superposición de todos los elementos de la base, y aplicar sobre él la definición del laberinto, para encontrar el conjunto de posibles soluciones. De hecho, si hay más de un camino oara encontrar la salida, entonces el estado final será la superposición de estas trayectorias, aunque en el proceso de medida el sistema se decida por una.

Visto esto, tal vez el lector aún sae pregunte ¿ y por qué no realizar estas operaciones sobre un array de bits en lugar de un array de qubits? La respuesta es simple: no podemos preparar el array de bits en un estado que sea superposición de estrategias, de modo que tendríamos que ir aplicando sucesivamente la definición de laberinto sobre todos los posibles arrays diferentes hasta encontrar aquél para el que al final no se obtuviese un array de ceros. En el método cuántico, en cambio, sólo necesitamos un número de operaciones proporcional al tamaño del laberinto para encontrar la salida. De nuevo nos hemos servido del principio de superposición.

4.5.2. Juegos de varios movimientos.

Por un lado podemos imaginar juegos de varias jugadas como el ajedrez, en el que un movimiento va seguido por otro del oponente, de modo que a cada paso (en el caso más sencilllo) será necesario reevaluar la situación. Podríamos imaginar cada alternativa codificada como un autovector en cierta base. Esta base podría, para hacerlo más simple aún, podría ser 4.8, pero reinterpretando los elementos. Ahora cada vector corresponde a una decisión diferente, dentro de las reglas del juego. El procesador enfrentado al humano podrá encontrarse en un estado que sea superposición de alternativas, de modo que se encuentra listo para evaluar los resultados de distintas decisiones. Por otra parte, las probabilidades asociadas a cada componente podrán ser una función de los objetivos, de modo que se filtren las alternativas claramente conducentes a la derrota. El movimiento del contrincante hará que, de acuerdo a las normas, ciertos movimientos pasen a estar prohibidos, lo que nos fuerza en el diseño a proveer un modo de filtrar las componentes de la superposición que no se permite ejecutar. Entonces el sistema se encontrará en un estado tal que permite la toma de decisiones coherentes con las reglas del juego, pero de modo no determinista. La amplitud de cada una de las probabilidades puede ser función de una base de datos con numerosas partidas, de modo similar a como hace un computador actual, pero con la ventaja de no obligar al sistema a

responder de un modo determinista. De hecho, esta misma ventaja, unida a algún medio de modificar la definición de la función ante el resultado de una partida puede dar a la máquina la posibilidad de *aprender* nuevas estrategias. Ajustar coeficientes es lo que hace una red neuronal, pero éstas no preveen la *improvisación* ante situaciones nuevas.

4.5.3. Algunas conjeturas sobre la naturaleza.

En este momento muchos harían conjeturas sobre si éste es uno de los mecanismos por el que nosotros mismos aprendemos y tomamos decisiones, como hizo Roger Penrose en su momento. Lo cierto es que aunque no parezca haber rastro de los "microtúbulos" en los que decía que los efectos cuánticos se verían amplificados a escala macroscópica en las neuronas, y pese a no ser el tema sobre el que trata este trabajo, me permitiré decir que tal vez no estemos ante el modo que la naturaleza escogió para los seres vivos, pero sí ante una máquina capaz de incorporar todos los elementos que yo reconozco en la toma de decisiones. Esto no quiere decir otra cosa que probablemente hayamos encontrado un sistema que modeliza ciertos esquemas de pensamiento. Esto, combinado con el principio de Church-Turing (véase sección 2.2.2) me lleva a la defensa de la IA-fuerte, siempre y cuando las máquinas de las que estemos hablando sean lo suficientemente generales.

Hasta ahora la idea de IA-fuerte siempre había tropezado con la aparente falta de potencia de los métodos de computación tradicionales. Siempre fui escéptico frente a tales objecciones, y no dejo de serlo, a la luz del hecho de que las diferencias entre la computación clásica y la cuántica son sólo de rendimiento. Es obvio que el conjunto de ecuaciones que determinan la evolución cuántica de un sistema pueden resolverse, como se ha venido haciendo hasta ahora, en un computador clásico. Sin embargo, si unimos el requisito de eficiencia, el principio de Turing se ve en dificultades, y se acepta la posterior generalización, y si consideramos que el principio de Church-Turing es válido (cosa que no tenemos por qué hacer, pero que yo hago) nos encontraremos con que nosotros mismos debemos ser modelizables por algún tipo de procesador general de la información.

5 Una aplicación llevada a la práctica:Criptografía cuántica.

He optado por dedicar un capítulo completo a la criptografía cuántica por una razón en particular: ésta ya *es* una realidad. Por ahora no deja de ser algo que se realiza de modo experimental, pero ya se ha llevado a la práctica. Por otra parte, si bien las comunicaciones forman parte de los sistemas computacionales hasta el punto de que éstos no son posibles sin ellas, lo contrario no ocurre, habiendo así un nivel de aplicabilidad que justifica por completo el desarrollo de esta técnica en ausencia del computador cuántico. Podemos utilizar mecanismos cuánticos de codificación para asistir la comunicación entre computadores electrónicos o entre usuarios humanos que, por ejemplo, hablan por teléfono.

5.1. Justificación de la criptografía cuántica.

En la sección 3.2.5 estudiamos un método de comunicación considerado actualmente seguro, el RSA. Este método, al igual que todos los que actualmente resisten todos los ataques, basan su fiabilidad en la clase a la que pertenece el problema de descomponer un número en factores primos: no existen computadores electrónicos capaces de descifrar un mensaje de este tipo sin la clave privada en un tiempo razonable. Pero a la luz del método desarrollado por Shor vemos que la existencia del computador cuántico rompería por completo la seguridad de los códigos actuales. Esto puede querer decir que, en ausencia de computadores cuánticos, emplear una tecnología más cara para resolver un problema que en la práctica no existe resulta como mínimo inadecuado. ¿Por qué entonces hacerlo? Básicamente porque la posibilidad de construir computadores cuánticos es aún una incertidumbre, y hay numerosas instituciones (como pueden ser las militares) que no se pueden permitir arriesgar la seguridad de sus datos ante la evolución de la tecnología.

De cualquier modo, el enfoque de este trabajo es relativo a la física, y por tanto a la posibilidad de realizar tareas nos importa mucho más que la viabilidad de las mismas. Desde mi punto de vista, aún así, todo el computador cuántico gira en torno a problemas de viabilidad: su misma realización pretende hacer viables tareas que no lo eran hasta ahora. Lo que es en mi opinión interesante de la criptogrfía cuántica, aparte de como problema particular de

la teoría cuántica de la información, es que responde afirmativamente a la pregunta:

> *¿Existe algún modo de comunicar información que sea inherentemente seguro, es decir, cuya seguridad resida en las propias leyes de la física?*

Los códigos tradicionales basaban su seguridad en una cuestión de rendimiento. Los requisitos para descifrarlos crecen exponencialmente cuando lo hace linealmente el esfuerzo empleado en proteger la información, pero no hay ningún principio físico que niegue la posibilidad de que esto se consiga. La criptografía cuántica, en cambio, es totalmente segura desde este punto de vista.

5.2. Descripción de una transmisión.

Los métodos criptográficos cuánticos pueden agruparse en dos categorías, que describo a continuación:

5.2.1. Distribución de clave cuántica.

En este procedimiento la llave secreta, en lugar de ser generada a partir de números primos, se obtiene mediante estados cuánticos. Una estrategia posible viene descrita a continuación:

Alice envía 2n qubits a Bob, cada uno de ellos preparado en uno de los estados {|0>, |1>, |+>, |->} de modo aleatorio. A continuación Bob recibe la información, y mide los estados en una base particular, ya sea {|0>, |1>} o bien {|+>, |->}. Finalmente Alice indica públicamente a Bob la base utilizada para preparar cada qubit. En promedio Bob habrá usado para hacer las medidas la misma base que Alice al preparar los estados la mitad del tiempo. Los únicos resultados con sentido son precisamente estos. Por último, Alice y Bob comparten la misma cadena de n bits, donde por ejemplo |0>y |+>se asocian a “0” y |1>y |->lo hacen a “1”. A esta cadena la llamamos RQT, por *raw quantum transmission* (es decir, algo así como transmisión cuántica literal).

Un espía que pretendiese interceptar la cadena de qubits dejaría evidencia de su presencia, pues la información no puede clonarse, y para hacerse con ella ésta no podrá alcanzar nunca a su receptor legítimo. Si el interceptor opta por realizar medidas y reenviar la información aproximadamente la mitad de las veces acertará con la base en la que hacerlas, de modo que no perturbará el

qubit correspondiente, pero la otra mitad su intervención será evidente a la hora de comparar lo que Bob recibió al principio con la cadena RQT: de los n bits de la cadena la cuarta parte será inconsistente con el estado cuántico.

Si Alice y Bob comparten públicamente n/2 bits elegidos aleatoriamente de la cadena RQT y encuentran que coinciden en sus valores, podrán estar seguros de que nadie ha interceptado el mensaje. La probabilidad de que con n=1000 bits Alice y Bob hayan escogido sólo bits que no hayan sido modificados en el ataque es de $(3/4)^{n/2} \simeq 10^{-125}$.

Hay otras estrategias posibles, y descifrar cadenas puede ser aún más complicado para un espía en función de ellas. Por otra parte, tenemos ruido, haciendo que algunos qubits varíen impredeciblemente su estado.

5.2.2. Comunicación segura en presencia de ruido.

Si hay ruido en la comunicación aparecerán errores en la secuencia RQT que no estarán asociados a ningun ataque. Esos errores esperamos que sean escasos, pues los errores se espera que se produzcan de manera esporádica, mientras que los debidos a un ataque afectan a la mitad de los qubits y, por diseño, a la cuarta parte de los bits de la secuencia RQT. Así que, bajo el esquema anterior, una tasa de error inferior al 25 % significa que la línea no ha sido pinchada.

El siguiente paso es corregir los errores. Esto puede hacerse compartiendo resultados de bits de paridad en subconjuntos del mensaje elegidos aleatoriamente.

El último paso es extraer de la clave otra más pequeña, compuesta por valores de paridad obtenidos a partir de la original. De esta clave, de alrededor de n/4 bits, un espía conocería 10^{-6}de un bit (Bennett et. al. 1992).

5.2.3. Bit commitment.

Partamos de la situación en la que Alice decide el valor de su qubit de modo que Bob puede a partir de cierto instante estar seguro de que ella ha decidido el valor, pero no puede saber de que valor se trata hasta algun instante posterior, escogido tambien por ella. En el marco clásico esto podría hacerse guardando el mensaje dentro de una caja fuerte, para, despues de elegido el momento en que Alice permita a Bob leerlo, facilitarle la combinación.

El modelo cuántico del quantum commitment consiste en una implementación particular de un protocolo en el que Alice envía un qubit a Bob, y sólo ella puede elegir el momento de decirle en que base fue preparado. Ni Bob ni un tercero pueden permitirse tratar de medir el qubit sin conocer la base, pues

ademas de no poder interpretarlo echarían a perder el estado cuántico dotado de la información.

5.3. Realizaciones prácticas.

La realización que se ha llevado a la práctica es la distribución de clave cuántica. Uno de los primeros experimentos, que fue realizado por Bennett Y Bassard en 1989, demostró la fiabilidad de la idea. De ahí a los 23 Km de distancia que Zbinden comunicó por debajo del lago Geneva (Zbinden et. al. 1997) pasaron sólo ocho años.

En el experimento de Zbinden los qubits eran estados de polarización de pulsos láser. Los pulsos contenían 0.1 fotones en promedio, es decir, pulsos con más de un fotón eran muy poco probables. Más de un fotón por pulso implica duplicidad de la información, y abre el camino a un tercero para interceptar el mensaje sin ser detectado. La tasa de error del sistema era del 1.35%. Vimos antes que para la estrategia descrita, un error menor del 25% permite estar seguro de que cierto mensaje no ha sido robado y la tasa de error en este caso es claramente inferior, de modo que el sistema es adecuado para la comunicación segura.

5.4. Observaciones.

Tambien hay más alternativas. Los pares EPR (2.3) pueden usarse, de modo que Alice y Bob pueden preparar estados y realizar medidas a lo largo de diferentes ejes, pero un espía introduciría correlaciones EPR que serían detectables.

Las pruebas sobre las que descansa la fiabilidad de los métodos criptográficos son púramente teóricas. Existen principios de los que se sigue hasta que punto es invulnerable un sistema cuántico de encripción. Por contra, no hay manera de demostrar experimentalmente esto.

El mecanismo conocido como bit commitment se ha demostrado recientemente que no es seguro (Mayers 1997, Lo y Chau 1997), debido a la posibilidad de engañase a los participantes haciendo uso del entrelazamiento.

6 El computador cuántico.

Comenzaré por hablar de qué es precisamente un computador cuántico, para luego entrar en detalles sobre sus componentes y estructura.

6.1. El computador cuántico.

Recurriré a la definición de Deutch (1985, 1989) para modelizar los computadores cuánticos (de ahora en adelante, QC, por *quantum computer*).

> *Un computador cuántico es una colección de n qubits sobre los que es posible:*
>
> 1. *Cada qubit puede prepararse en un estado conocido |0>*
> 2. *Los qubits pueden medirse en la base {|0>,|1>}*
> 3. *Sobre cualquier subconjunto de qubits de tamaño fijo podemos aplicar una (o un conjunto de) puertas universales.*
> 4. *Los qubits sólo evolucionarán de la manera prevista en las puertas.*

El último punto es el que trae más problemas, debido a la existencia de decoherencia. No podemos esperar de un sistema cuántico convencional que su evolución se produzca de manera totalmente controlada. Si bien esta es una limitación física, podríamos modificar el enunciado de modo que fuese menos restrictivo, al tiempo que lo suficiente para poder seguir llevándonos a una definición útil. Por ejemplo, podríamos conformarnos con un sistema cuya evolución fuese al menos en cierta medida controlada, de modo que una parte de la información que contiene evolucionase de un modo previsible.

6.2. Modelos de computador.

La discusión sobre modelos particulares de computador está aquí orientada hacia la construcción, pero no desde el punto de vista físico, sino de diseño. Si no se hizo antes este sería un buen momento para consultar el apartado 3.2.2 , sobre la máquina de Turing. La máquina de Turing era un apartado que iba a continuación en la versión preliminar de este trabajo, debido a que desde luego es un diseño particular de computador. Sin embargo, debido tanto a motivos históricos como al hecho de que introducía conceptos que eran necesarios más atrás, decidí colocarla al principio.

6.3. El modelo de circuito cuántico.

El modelo de red es el más utilizado en computación convencional. Se basa, como dije antes, en la concatenación de etapas de puertas lógicas (no necesariamente binarias). Este modelo es fácilmente traducible a un mundo de puertas lógicas cuánticas, si bien no aporta nada nuevo sobre el modelo tradicional, y tropieza con muchas dificultades. La ventaja que tiene no es otra que el hecho de ser un modelo más maduro, y por tanto con más posibilidades de llevarse al laboratorio. El inconveniente consiste enque lo que hacemos es trasladar un diseño que surgió para sistemas clásicos al campo de los sistemás cuánticos, por lo que el modelo en sí no explota las particularidades de este dominio.

Aquí no debemos entender las puertas lógicas como en los circuitos electrónicos. En un circuito electrónico un a puerta lógica era algún dispositivo físico que encuentra una señal eléctrica a su paso, y que es capaz de permitirle o no el paso en función de uns determinadas circunstancias.

En una red cuántica, en cambio,las puertas lógicas no pueden realizarse de esta manera por varias razones. Entre otras cosas, no podemos clonar a voluntad un cierto estado, mientras que en un circito electrónico esto es inmediato. Por otra parte, la naturaleza de los qubits (habitualmente magnética) requiere que utilicemos por ejemplo campos magnéticos para manipularlos.

Podemos entonces preparar una especie de trayectoria que vaya recorriendo todo el sistema, salpicada de regiones donde producimos campos magnéticos durante tiempos tan cortos que podamos estar seguros de que sólo afectan a un qubit, y tan bien sincronizados que además sabremos a que qubit afectan. Deberíamos hacer tantos de estos dispositivos como etapas tenga la operación que hayamos previsto realizar. Todo esto resulta absurdamente complicado.

Lo razonable es dejar los qubits fijos en el espacio, y operar sobre ellos con un único conjunto de actuadores que produzcan los respectivos campos magnéticos, o el efecto que queramos aprovechar para modificar el estado del arreglo. Si se tratara de campos magnéticos, obligando a los spines a orientarse de determinada manera, no sería necesario que nos preocupáramos de si éstos están activos un poco más de tiempo de la cuenta, pues cada electrón acabaría en el mismo estado final. Así emos que la idea de red en computación cuántica tiene muy poco que ver con la de la computación tradicional.

6.4. El autómata celular cuántico (QCA).

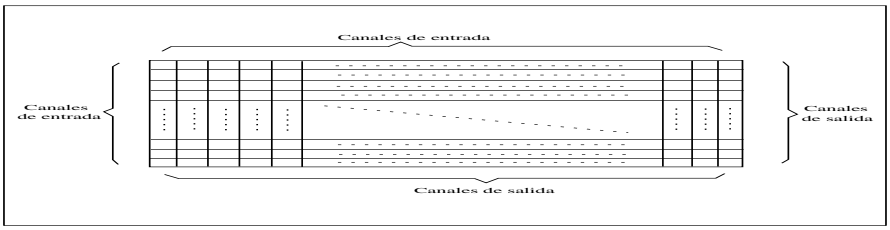
A diferencia de los modelos anteriores, el autómata celular cuántico está diseñado de modo que aprovecha el comportamiento de los sistemas a escala cuántica. El modelo de red o circuito cuántico, por ejemplo, no trata de ser más que una adaptación del modelo de circuito tradicional, aunque como tal tropieza con dificultades, como lamiposibilidad de clonar los estados y el hecho de que debemos pensar en las operaciones como operadores actuando sobre

registros cuánticos, y no necesartiamente como etapas en la propagación de señales. Hablaré en detalle sobre este modelo porque me parece de particular interés.

6.4.1. Nociones generales sobre el QCA.

Básiacamente, un autómata celular consiste en un array de quantum dots, acoplados mediante interacción culombiana. En cada celda, los electrones tienen un estado bien definido, y determinado por la interacción con las celdas vecinas. La probabilidad de efecto tunel será despreciable. Más adelante veremos que no es complicado en principio conseguir que un sistema de este tipo realice diferentes tipos de operaciones. De hecho, si hay algo difícil en el autómata celular es la conexión con el exterior, en mucho mayor medida que la implemantación de las operaciones.

En la definición de computador cuántico (sec. 6.1) veíamos que es necesario proveer un método para inicializar los qubits en un estado conocido. Esto lo haremos por medio de unos electrodos de control. El estado de todo el sistema viene dado por unas condiciones de contorno que son precisamente las que producen los canales de entrada, y es ahí donde los electrodos de control irán situados. Sin entrar aún en la descripción de cada celda, un autómata celular en dos dimensiones es el sistema representado en la siguiente figura:



Autómata celular cuántico.

Existe un problema, asociado a la interconexión de dispositivos cuando el tamaño disminuye ([21]), que resuelve el esquema propuesto. Como cólo suministramos entradas desde los bordes del array, a medida que este se hace más pequeño las energías características asociadas se hacen cada vez más importantes, y con ello se hace factible realizar computacione s a mayores temperaturas. Se considera posible llevar de esta manera la computación cuántica a escala molecular.

En lo que se refiere a este trabajo, me aproximaré al autómata celular primero desde su comportamiento independiente del tiempo, para luego ver cómo evoluciona con el paso de éste.

Para el caso independiente del tiempo, lo que debemos hacer es buscar el estado fundamental de los arrays de quantum dots. Recordando algo sobre el modelo de Ising, lo primero que observamos es que *no importa cómo evolucione* el array , sino hacia dónde evoluciona. Esto hace aún más interesante

el modelo QCA, dado que podemos obviar una serie de operaciones que sería inabordable a poco que el array aumentara de tamaño. La computación puede llecvarse a cabo correctamente sin necesidad de controlar todos los detalles de la evolución del sistema, ésta se realiza preparando el sistema en un estado tal que la solución de su evolución corresponda a la del problema propuesto.

La necesidad de estudiar la dinámica del autómata celular no está asociada tanto a la realización de las operaciones como a la de averiguar la velocidad con que éstas serán llevadas a cabo. Por otra parte, existe la posibilidad de que bajo ciertas condiciones iniciales el sistema no alcance la solución, sino algún estado metaestable a medio camino entre ella y el estado inicial, y queremos saber si el estado final será alcanzado o no. Vemos, de todas formas, que este estudio no necesita alanzar la profundidad de ver en detalle cómo evoluciona cada celda, sino que más bien nos preocupas decidir cuándo una señal ha alcanzado el extremo del array, o cuándo podemos decir que la operación ha sido finalizada.

6.4.2. Acoplamiento con el exterior.

Nos centraremos en dos límites para el acoplamiento con el exterior:

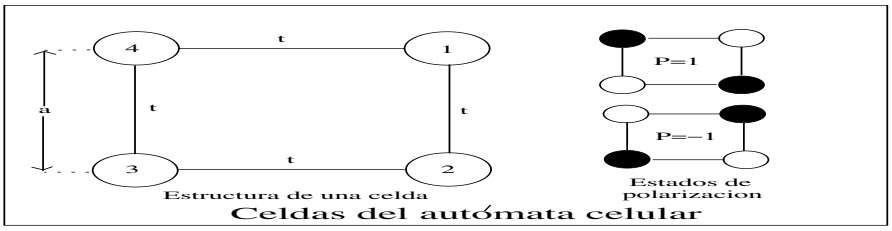
1. La evolución (elástica) del sistema tiene lugar en una escala de tiempo mucho más larga que el acoplamiento (inelástico) entre el autómata y el medio exterior. Si el acoplamiento con el exterior es lo bastante alto como para forzar el tránsito al estado fundamental entonces la dinámica del sistema puede describirse mediante la tasa dada por la regla de oro pasa el scattering a partir del estado inicial, sin necesidad de saber qué ocurre en las etapas intermedias.

2. Nosotros nos interesamos por el otro límite, en el que el acoplamiento (inelástico) con el exterior tiene lugar en escalas de tiempo mucho mas largas que el acoplamiento (elástico) entre dots. Esta situación hace aceptable la aproximación de sistema aislado, y el estudio de la evolución mediante la ecuación de Schrödinger (ec. 2.16).

Una vez establecido que nuestro autómata evoluciona como n sistema aislado, resta decir que se sabe que es posible detectar el estado de carga de una celda de modo no invasivo [12], desde el punto de vista clásico. Una medida siempre provocará el desplome del estado cuántico sobre el subespacio saociado al valor encontrado, de modo que no podrá ser no invasiva desde el punto de vista cuántico. Aún así, en el QCA lo que se hace es medir el estado de las celdas de los bordes una vez que se ha alcanzado el estado de equilibrio, y esto se hace a lo largo de un intervalo de tiempo largo comparado con el de fluctuación de los estados de cada celda, obteniéndose así un valor esperado.

6.4.3. Descripción del autómata celular.

En un autómata celular no asociaremos cada celda a un único quantum dot, sino que haremos construcciones como la de la figura:



Celdas de nuestro QCA.

En este caso el esquema para cada celda consiste en cuatro quantum dots por cada celda. Hay otros modelos, como el que incluye un quantum dot adicional en el centro. En el interior de la celda hay dos electrones. Como dije antes, no hay efecto tunel. A medida que las celdas se hacen más pequeñas, la separación entre niveles energéticos se hace mayor, lo que acelera la respuesta temporal de los sistemas.

Haré referencia a los resultados obtenidos por Douglas y Lent [12], para lo que se hace necesario que me refiera a su celda estándar. Esta celda parece mostrar un buen comportamiento, y por eso fue elegida por ellos. Para este caso, la distancia entre dots (t, en la figura anterior) del interior de una celda es de 20nm, mientras que los centros de las celdas distan 60nm entre sí. Para esta construcción, la energía de efecto tunel es de 0.3MeV, mientras que el resto de parámertos fueron escogidos de acuerdo a los del GaAs.

El hamiltoniano empleado para el análisis de la evolución de tales sistemas fue el de *Hubbard extendido* :

$$H^{cell} = \Sigma_{i,\sigma}(E_0 + V_i)\hat{n}_{i,\sigma} + \Sigma_{i>j,\sigma}t_{i,j}(\hat{a}^{\dagger}_{i,\sigma}\hat{a}_{j,\sigma} + \hat{a}^{\dagger}_{j,\sigma}\hat{a}_{i,\sigma}) +$$

$$+\Sigma_i E_Q \hat{n}_{i,\uparrow}\hat{n}_{i,\downarrow} + \Sigma_{i>j,\sigma,\sigma'}V_Q\frac{\hat{n}_{i,.\sigma}\hat{n}_{i,\sigma'}}{|\overrightarrow{R}_i - \overrightarrow{R}_j|} \tag{6.1}$$

En este hamiltoniano no entran en juego los grados internos de libertad de la celda. Veamos lo que significa cada término:

1. Los operadores $\hat{a}_{i,\sigma}$ y $\hat{a}^{\dagger}_{i,\sigma}$ destruyen y cean electrones con spin σ en la posición i (son operadores de creacdión y aniquilación). esto hace que el término $\hat{n}_{i,\sigma} = \hat{a}^{\dagger}_{i,\sigma}\hat{a}_{j,\sigma}$ sea el operador número, que da el número de electrones con spin σ que opupan la i-ésima celda, y, por tanto $(E_0 + V_i)$ es la energía de cada electrón. Así, el primer término de la ecuación es la energía de cada quantum dot aislado. El potencial V_i es generado por las distribuciones de carga exteriores a la celda, de modo que es aproximadamente una constante dentro de ella.

2. El segundo término da cuenta del efecto tunel, con $t_{i,j} = 0{,}3MeV$, la energía de efecto tunel, y el término entre paréntesis el operador responsable de que el electrón j pase a la opsición i o al contrario (destrucción de uno, creación del otro). Para saltos en diagonal tomamos $t_{i,j} = 0$.

3. El tercer término expresa la energía necesaria para poner dos electrones con spines contrarios en el mismo quantum dot, E_Q cada vez que esto ocurra.

4. El último término de la ecuación es puramente culombiano, y se refiere a la interacción entre electrones del interior de la misma celda.

La ecuación de Schrödinger independiente del tiempo

$$\hat{H}^{cell}|\psi_i >= E_i|\psi_i > \tag{6.2}$$

es la que nos permitirá encontrar el estado estacionario de la celda. Ahora nos queda elegir la base adecuada para tratar el problema. Los autoestados de $\hat{H}^{cell}$ serán en principio los de la base para *dos electrones de spin contrario y cuatro posiciones*:

$$\left\{|\phi_1 >= \left| \begin{array}{cccc} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{array} \right\rangle ; |\phi_2 >= \left| \begin{array}{cccc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array} \right\rangle ... |\phi_{16} >= \left| \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{array} \right\rangle \right\} \tag{6.3}$$

Esta base desde luego necesita una explicación:

En la fila superior de cada vector aparecen las posiciones ocupadas por electrones con spin *up* con un uno, y las no ocupadas de esta forma con un cero. Hacemos exactamente lo mismo para los electrones con spin *down* en la fila de abajo. Los números de columna corresponden a las posiciones con que numeré los quantum dots en la figura anterior. Un electrón apuntando hacia arriba en la primera posición y otro apuntando hacia abajo en la trecera vendrían representados como

$$|\phi >= \left| \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array} \right\rangle$$

En esta base el hamiltoniano es una matriz de 16×16, de la forma $< \phi_i|\hat{H}|\phi_j >$, mientras que el estado fundamental de la celda veine expresado como:

$$|\psi_0 >= \Sigma_j \psi_j^0 |\phi_j > \tag{6.4}$$

donde $|\phi_j >$ son los distintos elementos de la base.

La base anterior puede sin embargo reducirse, teniendo en cuenta una serie de consideraiones. Al tomar efecto tunel despreciable, el número de electrones está restringido a dos por celda. Dado que el término $t_{i,j}$ valía cero para posiciones enfrentadas por una diagonal, el estado fundamental esperamos que sea precisamente este para la celda. Estos son los dos estados de polarización

contemplados a la derecha en la figura. Si la energía de efecto tunel se hace comparable a las energías de interacción culombianas del problema los electrones perderán rápidamente su estado de localización , y los estados de polarización dejarán de estar bien definidos. Así pues, para nosotros los términos de efcto tunel del hamiltoniano serán pequeños frente a los demás, y el estado del sistema muy próximo a los de la figura.

Definiremos la *polarización de la celda* como:

$$P = \frac{(\rho_1 + \rho_3) - (\rho_2 + \rho_4)}{\rho_1 + \rho_2 + \rho_3 + \rho_4} \tag{6.5}$$

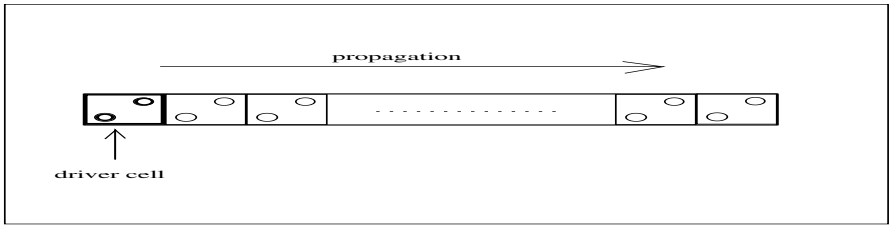
donde $\rho_i =< \psi_0|\hat{n}_i|\psi_0 >$ es el valor esperado del operador número para la posición i en el estado fundamental $|\psi_0 >$. es precisamente esta función la que toma valores ± 1 para los estados de la figura. De ahora en adelante P=1 equivale a un qubit "1", mientras que P=-1 equivale a un qubit "0".

Ahora veamos qué efecto tiene una celda sobre sus celdas vecinas. Los resultados de Douglas y Lent, para una serie de simulaciones, fueron los siguientes:

Partiendo de fijar el estado de una de las celdas (denominada celda de control, o de entrada) , procedieron a buscar el estado fundamental de la celda adyacente directamente, por medio de la diagonalización de 6.1. El efecto resultó ser muy próximo a una función escalón, en la que P_2 toma valor 1 por pequeño que sea el valor de P_1, mientras este sea positivo, y toma valor -1 con cualquier valor negativo de P_1. Vemos que un valor muy pequeño de polarización se convierte inmediatamente en un valor extremo. Esto permite evitar que una señal se pierda por el camino cuando es transmitida por un hilo de celdas.

El siguiente paso fue precisamente simular el hilo de celdas, para lo que debíamos hacer algunas aproximaciones más si queríamos que el problema fuese tratable. Lo más simple fue la *aproximación entre células de Hartree*. En esta aproximación se incluyen los efectos de correlación e intercambio dentro de cada celda, pero se desprecia la posibilidad de que se produzcan entre celdas diferentes. Todo el efecto de las celdas vecinas se recoge en el potencial V_i del primer término de 6.1. Así podemos encontrar el estado fundamental de una celda examinando sólo el hamiltoniano local. El procedimiento escogido para conocer el estado de todas las celdas fue fijar la polarización de todas menos de la que se analiza, para a continuación hacer lo mismo con cada una de las demás. Esto se realiza iterativamente hasta el momento en que el estado de todas las celdas permanezca sin cambios de una etapa a la siguiente. El inconveniente es que proceder así nos lleva al estado de equilibrio, pero no nos informa sobre la dinámica.

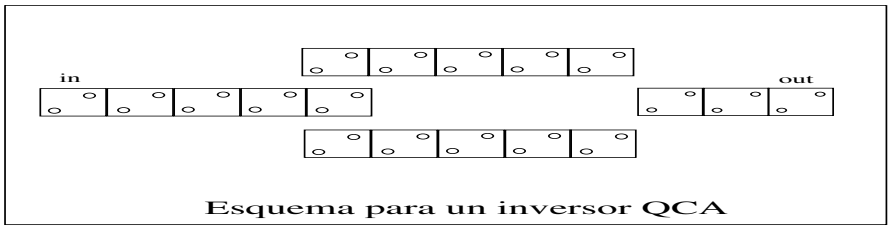
La evolución de un sistema como el anterior obedece a lo que puede verse a continuación:



Equilibrio en un array de celdas controlado.

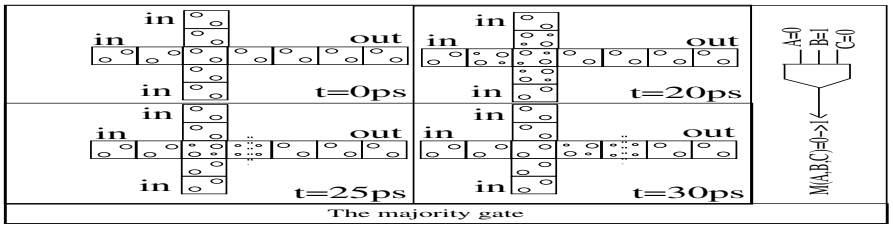
Lo que llama la atención es que el estado no se pierde, sino que avanza de una celda a otra. La celda sombreada es la de control, que mantuvimos en en el estado P=1. La hipotética pérdida de polarización debida a cualquier efecto no será importante a menos que ésta se invierta, dado que cada celda amplifica P hasta su valor de saturación.

Si quisiéramos realizar una operación sencilla, tal como una inversión, podríamos recurrir a la siguiente estrategia:



Esquema para inversión de una señal.

Pero no sólo necesitamos realizar operaciones sobre un único qubit, como sabemos. Aquí vemos una puerta generalizada en funcionamiento:



Puerta generalizada en QCA.

con estas operaciones es claro que podríamos, en principio realizar cualquier operación más complicada. Por otra parte, vemos que lo único que debamos hacer es introducir las entradas adecuadas desde las líneas conectadas a los bordes del autómata.

Dado que este apartado pretendía ser puramente descriptivo, y que sólo el autómata celular sería suficiente para un trabajo bastante extenso, no he creído conveniente continuar aquí con la descripción de la evolución. De cualquier

modo, es precisamente el hecho de que no es necesario entrar en ese tema para conocer el estado de equilibrio lo que hace al autómata celular un modelo tan potente.

6.4.4. Problemas del QCA.

En principio, una limitación básica para la construcción del QCA es la púramente tecnológica. En primer lugar, nos enfrentamos a la necesidad de crear un array de quantum dots, que son de por sí nanométricos, en el que las distancias entre unos y otros lo deben ser tambien. Este problema, a pesar de todo, no parece excesivamente importante, dado que se trata de simple miniaturización, la búsqueda de una técnica que haga posible reproducir a escala muy pequeña un patrón sencillo. Dado que la construcción de quantum dots es factible en la práctica, es razonable pensar que este tipo de construcciones lo llegarán a ser tambien. El otro problema surge cuando nos proponemos por un lado controlar todas las entradas, cada una de las cuales tiene escala nanométrica, pero necesita un dispositivo capaz de polarizarla adecuadamente para preparar el estado inicial y cada salida, donde necesitamos realizar medidas sobre todos las celdas de un lado del array. Si tratamos de realizar operaciones en el sentido descrito antes debemos enfrentarnos a saber que filas interactúan con que columnas, y necesitamos controlar exactamente las entradas y salidas correspondientes, lo que significa ser capaces de manipular y leer las entradas y salidas de una en una, al tiempo que sincronizarlas. Esto es técnicamente más complicado, y está relacionado con el problema de la interconexión, al que me referiré más adelante.

7 Construcción del computador cuántico.

Por ahora no hablamos de computadores cuánticos, sino de *procesadores cuánticos de la información,* siendo incluso esta denominación pretenciosa. Al hablar de computador cuántico nos referimos a una máquina de propósito general, capaz de ejecutar cualquier tarea simplemente preparándola de modo adecuado. Un procesador de información puede ser una máquina con una tarea mucho más específica, tal como realizar operaciones matemáticas, o adaptar señales eléctricas para digitalizar sonidos. Las tareas que la mecánica cuántica computa de manera más eficiente que la clásica no son todas; de hecho, son una pequeña parte, cuya aplicabilidad en el fondo es más bien reducida. Un usuario corriente no está preocupado de simular sistemas físicos ni de romper claves seguras en las comunicaciones de otros usuarios. De hecho, la mayoría tampoco aspiran a utilizar criptografía cuántica en sus mensajes, dado que en ellos no hay un contenido cuya protección merezca inversiones económicas tan fuertes.

Vistas las cosas de este modo parece que, al menos como primer fin, el objetivo a nivel tecnológico en computación cuántica es el de construir máquinas capaces de ejecutar específicamente aquellas tareas en las que la mecánica cuántica suponga una verdadera ventaja, independientemente del coste, dado que los primeros usuarios serán instituciones públicas o grandes compañías. Algo parecido ocurrió con la computación electrónica, aunque su evolución desembocó finalmente en la situación que conocemos hoy en día.

7.1. Decoherencia. Códigos cuánticos detectores de error.

7.1.1. Significado de la decoherencia

Ningun sistema cuántico está realmente aislado. Ya indiqué en 2.2.1 que utilizar la ec. Schorödinger para describir la evolución de los sistemas era hacer una apoximación. Para establecer una definición, diré:

> *Decoherencia es el fenómeno a través del que las componentes de un vector de estado* $|\psi>=\Sigma_i c_i|\phi_i>$ *se desfasan unas respecto a otras por efecto de la interacción con el entorno, destruyéndose en el proceso la información condensada en el entrelazamiento.*

Sabemos que los alrededores introducen una componente no unitaria en el operador de evolución:

$$< componentenounitaria >$$

Añadiré ahora que el entrelazamiento con los alrededores puede destruir un estado en tiempos que hacen impesable cualquier tipo de uso computacional. La pérdida de coherencia en escalas de tiempo tan breves tiene relación directa con el hecho de que no observemos fenómenos cuánticos a nuestra escala. Queremos que el computador cuántico lleve a nuestra escala fenómenos que sólo se producen a nivel microscópico, así que debemos encontrar un modo de preservar la coherencia. Ambas cosas se tratan a continuación.

7.1.2. Códigos cuánticos de detección de error

En conexión con la sección 2.1.2 (detección de errores) comenzaré definiendo la *matriz de chequeo de paridad*. Decimos que un código detector de errores es *lineal* si es cerrado ante la suma:

$$\mathbf{u} + \mathbf{v}\epsilon C, \forall \mathbf{u}, \mathbf{v}\epsilon C$$

Un código de este tipo queda totalmente especificado por su matriz de chequeo de paridad, H, que es un conjunto de n-k palabras de bits linealmente independientes, que satisfacen:

$$H \cdot \mathbf{u} = 0, \forall \mathbf{u}\epsilon C$$

Esto se traduce en la práctica en:

$$H \cdot (\mathbf{u} + \mathbf{e}) = H \cdot \mathbf{u} + H \cdot \mathbf{e} = H \cdot \mathbf{e} \tag{7.1}$$

Al último término se lo denomina síntoma de error, y la clave de la idea es que delata el hecho de que un error se produjo sin necesidad de que Bob lea el mensaje (**u**), cosa que lo alteraría. De modo automático puede sustraerse el vector de error identificado, **e**, del mensaje original sin que por ello sea necesario leerlo en ningun momento.

Ahora pasaré de este nivel de formalidad a algo mucho más concreto. En primer lugar, la idea de introducir redundancia en los mensajes transmitidos tropieza con el teorema de no clonación (2.4.6). Si bien podemos pensar en un emisor que preparase múltiples copias de un cierto estado cuántico, en el proceso de medida el receptor no tendría modo de comparar cada no de estos estados con los demás, pues al medirlos los echaría a perder, y distintos estados cuánticos conducen a idénticas probabilidades para cada valor propio, como ocurre con $(1/\sqrt{2})(|0>+|1>)$ y $(1/\sqrt{2})(|0>-|1>)$.

Reproduciré el ejemplo empleado por Bennett y Shor en [4] para describir cómo se construyen los códigos cuánticos detectores de error. Primero partamos de la idea clásica de que para corregir un error basta repetir la información:

$$|0> \longrightarrow |000>$$

$$|1> \longrightarrow |111> \tag{7.2}$$

si sólo hubiésemos duplicado la información, al producirse un error no sabríamos cuál de las alternativas es la correcta, mientras que aquí podemos suponer que la información correcta es la que aparece más veces. Aquí no enviamos varios estados, sino que empaquetamos el estado original (perteneciente a un espacio de dimensión 2) en un espacio de Hilbert de dimensión más elevada (8, en particular). La medida se hará una sola vez, y no tropezaremos con la imposibilidad de clonar el estado. Este código ya evita el único error que nos preocupa de la información clásica (esto es, la inversión de bits). Si en una comunicación se produce el error |0>->|1>, este código será capaz de resolverlo. Pero la principal fuente de potencia computacional de la mecánica cuántica reside en el *entrelazamiento*, lo que quiere decir que esperamos conservar la fase de cada qubit en los estados que contienen la información. Una inversión de bit puede representarse por la aplicación del operador:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{7.3}$$

que hace la transformación |0>->|1>y |1>->|0>. Si se produce por ejemplo una inversión de fase en el qubit |1>:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{7.4}$$

tendremos un cambio de fase en la palabra completa. El código descrito en 7.2 ha incrementado la vulnerabilidad frente a cambios de fase, pues al proteger los estados |0>y |1>codificamos $(1/\sqrt{2})(|0> \pm |1>)$ como $(1/\sqrt{2})(|000> \pm |111>)$, donde un cambio de fase en cualquier qubit estropea la combinación y, al haber más qubits, es más fácil que el fallo se produzca. Un código cuántico de detección de errores (en adelante QECC) debe protetger el subespacio compoleto que contiene la información. Si observamos el comportamiento del operador de Hadamard2.27, encontramos el hecho de que la información puede estar contenida tanto en los bits como en las fases, y que se puede pasar de una representación a otra. La aplicación de la transformación de Hadamard convierte los errores de bit en errores de fase y los de fase en errores de bit. Apliquemos esta transformación a los estados de la base 7.2. El resultado:

$$\begin{aligned} |0> &\longrightarrow \frac{1}{2}(|000> + |011> + |101> + |110>) \\ |1> &\longrightarrow \frac{1}{2}(|111> + |100> + |010> + |001>) \end{aligned} \tag{7.5}$$

Ahora la protección contra errores de bit se ha convertido en protección contra errores de fase. Un error de fase en el tercer qubit, por ejemplo, lleva a un estado ortogonal $\frac{1}{2}(|000> - |011> - |101> + |110>)$ a los de la base, y al

salirse de ella el error se hace fácilmente identificable. Como era de esperar, al aplicar la transformación de Hadamard para hacer más seguro el código ante errores de fase, perdemos de nuevo la seguridad ante errores de bit. Esto se ve claramente si tenemos en cuenta que un error en cualquier qubit de una de las dos palabras lleva a la otra. Si no nos salimos de las palabras del código no tenemos forma de saber si se ha producido un error.

Pero no todos los cambios de fase son inversiones

En general un error de fase se representa por la aplicación del operador:

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \tag{7.6}$$

Las inversiones de fase son desde luego un caso particular de este tipo de errores, donde $\theta = \pi$. En la construcción anterior sólo hemos considerado inversiones de fase, de modo que da la impresión de que prácticamente no hemos hecho nada. Sin embargo ocurre que el código anterior no sólo protege contra inversiones de fase, sino tambien contra todos los errores de fase.

Cualquier estado cuántico puede variar en un *factor de fase global* sin perder su contenido informativo. Esto nos permite escribir el error de fase como:

$$\begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{-i\phi} \end{pmatrix}; \phi = \frac{-\theta}{2}$$

Para darnos cuenta de que tal cosa es posible consideremos este error actuando sobre el primer bit del estado |0>codificado:

$$|0> \longrightarrow e^{i\phi}(|000> + |011>) + e^{-i\phi}(|101> + |110>) =$$

$$= cos\phi(|000> + |011> + |101> + |110>) +$$

$$+ sin\phi(|000> + |011> - |101> - |110>)$$

Lo que aparece tras la ocurrencia del error es la superposición del estado |0>codificado sin errores con amplitud de probabilidad $cos\phi$, y de un estado codificado con un error de fase en el primer bit, cuya amplitud de probabilidad es $sin\phi$. Al hacer la medida del estado resultante encontraremos el estado sin errores con probabilidad $cos^2\phi$ y el estado con error con probabilidad $sin^2\phi$. La medida provoca el colapso de la función de onda, de modo que si encontramos la medida errónea sabremos cómo corregir el error en el estado del sistema. Esto prueba que cualquier error de fase es corregible con la técnica anteriormente descrita.

Previniendo errores de fase y de bit al mismo tiempo.

Una vez que conocemos una estrtategia para evitar tanto un tipo de error como otro, nos interesa el modo de evitar ambos a la vez. Para hacerlo, combinaremos ambas técnicas, triplicando el número de qubits. El siguiente código deja clara la idea:

$$|0> \longrightarrow \frac{1}{2}(|000> + |011> + |101> + |110>) \longrightarrow$$

$$\longrightarrow \frac{1}{2}(|000000000> + |000111111> + |111000111> + |111111000>)$$

$$|1> \longrightarrow \frac{1}{2}(|111> + |100> + |010> + |001>) \longrightarrow$$

$$\longrightarrow \frac{1}{2}(|111111111> + |111000000> + |000111000> + |000000111>) \qquad (7.7)$$

Obsérvese que lo único que diferencia el código definitivo del intermedio es la triplicación de cada qubit. La redundancia proteje contra los errores de bit, mientras que la etapa anterior (redundancia procesada con el operador de Hadamard) evita los errores de fase. Las técnicas empleadas no interfieren entre sí: por ejemplo, un error de bit es corregido vía chequeo de la redundancia sin afectar con ello a la fase de la superposición.

¿Que otros errores no hemos tenido en cuenta?

No sólo hay errores de fase y de bit, de hecho hay un espacio de errores completo. De los infinitos errores posibles sólo hemos visto dos posibilidades. Afortunadamente, siendo capaces de corregir cualquier error de bit o de fase podremos corregir tambien todos los demás tipos de errores. La matriz identidad y las matrices de Pauli

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}; \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad (7.8)$$

forman un espacio para *todas* las matrices de 4×4. La matriz identidad no hace falta decir que corresponde a la ausencia de error. No hacía falta, pero lo acabo de decir. La matriz $\sigma_x$corresponde a una inversión de bit, la matriz σ_z corresponde a un error de fase y σ_y representa una combinación de ambos errores.Como podemos corregir los diferentes errores asociados a las matrices de Pauli, y cualquier otro error es resultado de combinarlas, podemos corregir toda clase de errores. Encontramos que si podemos corregir cualquier producto tensorial de k matrices actuando sobre qubits diferentes entonces podemos corregir cualquier error que se produzca sobre k qubits.

Cuando realizamos la codificación estamos separando el espacio de los estados con la información del de los posibles errores en dimensiones ortogonales

del espacio de Hilbert. La ventaja de esto es que al hacer un chequeo para comprobar la presencia de errores no alteramos la parte del vector de estado que contiene la información, con lo que aún nos será posible corregirlos.

Existen otros códigos detectores de error, como por ejemplo el de Steane [17], que utiliza siete qubits en lugar de nueve. Relmente hay infinitos códigos posibles. Buscando el espacio de los códigos posibles apareció un código de cinco qubits[18, 19].

7.1.3. Capacidad de los canales cuánticos.

Aunque ganamos prestaciones al desarrollar los QECC, la contrapartida es que ahora entendemos muy poco sobre la capacidad de los canales cuánticos, en contraste con lo que ocurre con los clásicos.

En concordancia con la definición clásica (sección 2.1.2) decimos:

> *La capacidad sin asistencia de un canal cuántico ruidoso Q(N), donde N es el canal, es la mayor tasa para la que con un número n de qubits arbitrariamente grande y un ε arbitrariamente pequeño cualquier estado $|\psi>$pueda recuperarse con fidelidad mayor que 1-ε tras ser leído en el destino.*
>
> $$\lim_{\varepsilon\to 0}\lim_{n\to\infty} sup\{\frac{n}{m} : \exists_{m,\varepsilon,D}\forall_{\psi\in H_{2^n}} < \psi|DN^{\otimes m}\varepsilon(|\psi><\psi|)|\psi >> 1-\varepsilon\} \tag{7.9}$$
>
> *donde ε es el superoperador de codificación que pasa n qubits a m entradas de canal, y D es el superoperador de descodificación, que convierte m salidas del canal en n qubits.*

La capacidad clásica de un canal cuántico se define como:

$$\lim_{\varepsilon\to 0}\lim_{n\to\infty} sup\{\frac{n}{m} : \exists_{m,\varepsilon,D}\forall_{\psi\in\{|0>,|1>\}^n} < \psi|DN^{\otimes m}\varepsilon(|\psi><\psi|)|\psi >> 1-\varepsilon\}$$

que senillamente se distingue de la otra en que ahora los estados pertenecen a la base $\{|0>,|1>\}^n$, excluyendo la superposición. A simple vista podemos decir que Q(N)≤C(N).

7.2. Otros problemas: la interconexión.

El problema al que me refiero aquí hasta ahora no ha sido tenido en cuenta, pero era inevitable que apareciese. El dominio de la mecánica cuántica es un mundo a escala nanométrica, mientras que el de los usuarios de un hipotético QC es macroscópico. Es de suponer que para explotar las ventajas de la mecánica cuántica no escalaremos dispositivos tales como los cables

que salgan fuera del procesador, ni, desde luego la interfaces connosotros mismos, pues no hemos previsto encoger. Así que nos encontramos con un procesador de información constituído a partir de elementos de escala nanométrica, y tiempos de respuesta característicos de los sistemas cuánticos conectado a unos hipotéticos detectores y electrodos de control que deben ser capaces por un lado de controlar individualmente las entradas y medir las salidas, y por otro asociarse de modo que puedan funcionar coordinados para, por ejemplo, la medida del estado de un registro cuántico, tal vez constituído por miles de qubits, y llevar información de un dominio a otro sin cometer errores. Todo esto teniendo en cuenta que a escala macroscópica los tiempos característicos son siempre órdenes de magnitud mayores.

7.3. Alternativas para la construcción del computador cuántico.

Una vez llegados a este punto es momento de comenzar a hablar sobre *como* construir computadores cuánticos, tanto de qué se ha hecho como de que se considera hacer en el futuro. El tema abandona la teoría de la información, para ocuparse de aspectos más convencionales; podríamos decir que se trata de ingeniería. Esto nos limita desde la perspectiva de que no basta con encontrar un modo de fabricar un QC, sino que además esto debe conseguirse dentro de costes que se consideren aceptables. Tambien puede significar que en la práctica se escoja una alternativa de fabricación en lugar de otra más eficiente por motivos relacionados con los medios de producción, etc.. Sin embargo, la búsqueda de sistemas físicos capaces de servirnos supone la necesidad de resolver el problema de la evolución de sistemas cuánticos determinados, y en ese sentido estamos haciendo física.

A nivel elemental tenemos bastante idea sobre cómo realizar distintas operaciones sobre qubits. Desde luego, tal cosa depende de cómo hayamos escogido implementar los qubits en el QC. Cualquier transición entre niveles energéticos que fuéramos capaces de estimular podría ser un modo de actuar sobre los qubits, si es que hemos decidido almacenarlos de esta manera.

Nos enfrentamos a una de las mayores dificultades en la construcción del QC: la eslala. Un procesador cuántico de la información debe ser controlable, de modo que podamos realizar controladamente operaciones sobre los qubits, de acuerdo a la definición de la sección 6.1, pero al mismo tiempo debe ser lo bastante grande como para que sea útil. En la sección 4.1.2 dije que un QC no aporta ninguna ventaja sobre un computador convencional al factorizar números de 130 dígitos o menos.

El reto consiste en descubrir que sistemas físicos podemos emplear para realizar cálculos, que cumplan los requisitos anteriores. Lo primero que se nos viene a la cabeza es tratar de fabricar procesadores de estado sólido, del mismo modo que en computación clásica, pero tropezamos con la decoherencia.

En el interior de un sólido el acoplamiento entre vecinos es fuerte, de modo que cualquier estado que consiguiésemos producir se perdería en tiempos del orden de picosegundos. En concreto, donde la decoherencia actúa tan deprisa en en la destrucción de la fase de las superposiciones de estados, que son la clave para conseguir por ejemplo que la factorización se realice tan deprisa.

Dos de las alternativas posibles parece que pueden utilizarse para manejar decenas de qubits: se trata de las trampas iónicas (propuestas por Zirac y Zoller en 1995), y la de aprovechar la "bulk" resonancia magnética (de Gershenfeld y Chuang en 1997 y Cory et. al. en 1996).

7.3.1. Trampas iónicas.

La descripción de un prcesador que se aprovecha de trampas iónicas puede hacerse del siguiente modo:

1. Disponemos de una *"Trampa de Paul"*, que en esencia es una región de alto vacío (hablamos del orden de 10^{-8} Pa), donde una cadena de iones se mantiene confinada utilizando una combinación de campos eléctricos oscilantes y estáticos.

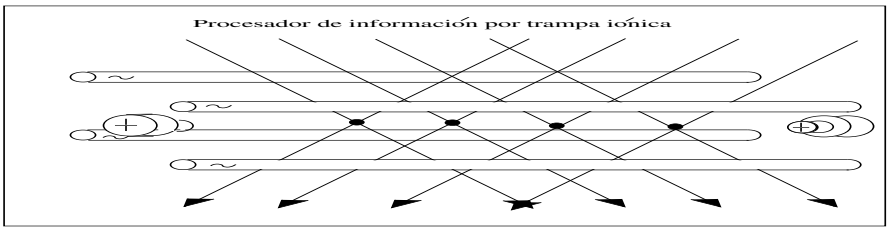
2. Hay un haz láser que se desdobla por medio de *desdobladores de haces* y *moduladores acustoópticos*. Obtenemos por este medio un par de haces para cada ión.

3. Cada ión tiene dos estados de vida media lo más larga posible. Podemos usar los subniveles de estructura fina del nivel fundamental. Llamaremos a los estados correspondientes |g>y |e>. Dado que van asociados a energías diferentes (en la base de estados de estructura hiperfina es donde podemos distinguirlos) , son necesariamente ortogonales.

4. Los pares de hacer láser se utilizan para inducir transiciones Ramman coherentes entre los niveles en los que hemos codificado los qubits. Esto permite aplicar operaciones sobre cada qubit, pero no operaciones sobre más de uno, tales como las puertas binarias o ternarias.

5. Para realizar operaciones sobre más de un qubit, tales como las binarias, aprovecharemos la repulsión culombiana entre los iones. Para hacerlo utilizamos un resultado de Zirac y Zoller que analizaremos a continuación.

La siguiente figura muestra un esquema del dispositivo:



Dispositivo ideado por Zirac.

Los haces de fotones que utilizamos para modificar el estado de los iones además de energía transportan momento. El momento suministrado a los iones se traduce en que toda la cadena se mantiene en estados vibracionales globales, que naturalmente se encuentran cuantizados, puesto que la trampa iónica mantiene confinados a los iones. Esto se conoce como *efecto Mössbauer*. Los estados de la cadena corresponden a números enteros de cuantos de energía de vibración, precisamente *fonones*.

El nivel fundamental de vibración corresponderá al estado |n=0>, mientrar que el primer nivel excitado lo hará con |n=1>, y así sucesivamente.

Supongamos que queremos realizar la operación Z controlada (recuérdense las operaciones 2.25, 2.22 y 2.29). Para realizar esta operación entre los iones x_i y x_j desde el estado vibracional fundamental |n=0>, hacemos que una pareja de haces de fotones lleven a cabo sobre el ión x_i la transición:

$$|n=0>|g>_i \longrightarrow |n=0>|g>_i$$

$$|n=0>|e>_i \longrightarrow |n=1>|g>_i$$

de modo que x_i siempre acaba en el estado interno fundamental. El estado de movimiento del i-ésimo ión queda así inicializado. A continuación aplicamos una pareja de haces sobre x_j de modo que se produzca la transición:

$$|n=0>|g>_j \longrightarrow |n=0>|g>_j$$

$$|n=0>|e>_j \longrightarrow |n=0>|e>_j$$

$$|n=1>|g>_j \longrightarrow |n=1>|g>_j$$

$$|n=1>|e>_j \longrightarrow -|n=1>|e>_j$$

esto es, invertir el estado de x_j sólo cuando nos encontremos al ión en el primer estado vibracional y en el nivel interno |e>.

Ahora aplicaremos de nuevo el puso inicial sobre x_i. El efecto de los fres pulsos se resume en:

$$|n=0>|g>_i\,|g>_j \longrightarrow |n=0>|g>_i\,|g>_j$$

$$|n=0>|g>_i\,|e>_j \longrightarrow |n=0>|g>_i\,|e>_j$$

$$|n=0>|e>_i|g>_j \longrightarrow |n=0>|e>_i|g>_j$$
$$|n=0>|e>_i|e>_j \longrightarrow -|n=0>|e>_i|e>_j$$

donde hemos actuado solamente sobre los estados internos de los iones, aunque para hacerlo nos hayamos aprovechado de los estados vibracionales.

La puerta Z controlada, junto con las transformaciones sobre un único qubit tambien constituye un conjunto de primitivas de la computación cuántica, análogamente al descrito en la sección2.4.4. Más referencias sobre el tema hay en [6].

Recordemos (apartado 6.1) que tambien debemos ser capaces de preparar la cadena en el estado |000...>, y que debemos ser capaces de medir el estado final. La primera operación la realizaremos por medio de *bombeo óptico* y *enfriamiento láser*, mientras que la medida puede hacerse con técnicas como el *salto cuántico* y el *shelving electrónico*. En 1995 Monroe consiguió aplicar todas estas técnicas a un sistema con un único ión atrapado (Monroe et. al. 1995b).

Para llevar los iones al nivel energético más bajo de estructura hiperfina hace falta llevar la temperatura por debajo de la millonésima de grado Kelvin. La principal fuente de decoherencia aquí es el calentamiento debido a la interacción entre el movimiento de la cadena y el ruido en los electrodos. Aún no se sabe como evitar este problema.

7.3.2. Resonancia magnética nuclear.

La siguiente propuesta se esquematiza en esta figura:

Procesador por RMN.

En el interior de la cápsula tenemos moléculas con un esqueleto interno de alrededor de 10 átomos, fijados a algunos otros por enlaces químicos. Para los núcleos de estas moléculas hay un momento magnético asociado a su spin nuclear. Estos spines serán los que utilizaremos como qbits. Las moléculas de este tipo son sometidas a un elevado campo magnético, y sus estados son manipulados por medio de pulsos magnéticos de duración controlada.

El problema en esta situación es que no hay modo de preparar una molécula en un estado inicial determinado. Entonces, en lugar de una única molécula, utilizamos un fluído con alrededor de 10^{20} moléculas, y medimos el spin promedio, cosa que puede hacerse si el momento magnético de los núcleos es lo bastante elevado como para producir un efecto medible.

El campo magnético no tiene el mismo valor en todos los puntos del recipiente, de modo que la evolución de cada procesador molecular es ligeramente diferente. Aplicamos entonces una técnica llamada de *spin-echo*, lo que permite invertir el efecto de la evolución libre de cada spin, sin que el efecto de las puertas cuánticas desecho. El pago por hacerlo es el aumento de la dificultad de implementación de muchas operaciones seguidas.

Volvamos al problema de la preparación del estado inicial. El líquido con el que operaremos se encuentra en equilibrio térmico, de modo que las probabilidades de ocupación de los distintos estados de spin obedecen a la distribución de Boltzmann. Además, partimos de la base de que las energías de estos estados son muy parecidas, con lo que las probabilidades de ocupación lo serán tambien. La matriz densidad de alrededor de 10^{20} spines nucleares se parece mucho a la matriz identidad.

$$\Delta = \rho - I \tag{7.10}$$

la matriz Δ (la pequeña diferencia) es la que se utiliza para almacenar la información. Esta no es la matriz densidad, pero se transforma del mismo modo que ella bajo pulsos magnéticos adecuadamente escogidos. De este modo, podemos llamar a este sistema un *computador cuántico efectivo*.

Actualmente somos capaces de manipular experimentalmente una cantidad de spines suficienmtemente elevada como para realizar operaciones con unos pocos qubits. Esto hace pensar que los primeros procesadores de información cuánticos aprovecharán esta técnica.

Lo malo es que no podemos aumentar indefinidamente la escala de computación esperando un comportamiento igualmente bueno. Con n qubits la señal pierde potencia en un factor 2^{-n}, de modo que no podemos aumentar indefinidamente el número de qubits. Otro problema que aparece es que al poder manejar solamente estados promedio de los spines nos vemos limitados para aplicar técnicas de corrección de errores.

7.3.3. Impurezas en semiconductores.

7.3.4. Quantum dots.

A continuación exploraré otra posibilidad para la construcción del QC, spines electrónicos en puntos cuánticos. Además de puntos cuánticos podríamos hacer uso de otros medios de confinamiento, tales como átomos o moléculas. Sin embargo, dado el grado de desarrollo de las técnicas experimentales asociadas a los quantum dots parece que esta será una de las primeras formas en que veremos contruir a los computadores cuánticos.

Los *puntos cuánticos* tambien se denominan *átomos artificiales*, debido a que son capaces de mantener electrones estados ligados, del mismo modo que los átomos... pero son mucho más fáciles de controlar. Los dispositivos de este tipo sabemos que permiten incrementar el número de qubits, y que la decoherencia no es tan importante como en otros esquemas.

En estructuras de GaAs podemos hacer variar el número de qubits de uno en uno (S. Tarucha et. al., 1996). Longitudes magnéticas del orden de 1nm se obtienen con campos de 1T, y esa es la escala de los quantum dots. Con quantum dots acoplados observamos efectos como la formación de un estado deslocalizado, "molecular". El entrelazamiento nos va a permitir realizar las operaciones que describí antes.

Elegimos entonces a los spines electrónicos como nuestros qubits, y a los quantum dots como los responsables del confinamiento. Ahora necesitamos una fuente de entrelamiento que sea *determista*. Dicho de otro modo, un modo de hacer que los qubits interactúen entre ellos (por ejmplo, a través de una XOR). Dos sistemas aislados no podrán nunca influir el uno sobre el otro. Podemos acoplar los spines durante un tiempo para conseguir esta interacción. Si tenemos en cuenta por lado la repulsión de Coulomb y por otro el principio de exclusión de Pauli, llegaremos a que el estado fundamental de una pareja de electrones acoplados es un singlete, que desde luego tiene un elevado grado de entrelazamiento.

Estamos interesados en la realización de las operaciones sobre los qubits, y a la luz de lo que acabamos de ver parece buena idea esudiar el hamiltoniano de acoplamiento. El hecho de tratarse de un singlete conlleva una energía de canje, asociada a la interacción entre spines:

$$H_s = J(t)\mathbf{s}_1 \cdot \mathbf{s}_2 \tag{7.11}$$

Supongamos que hacemos actuar a la energía de intercambio, de modo que tengamos:

$$\frac{1}{\hbar}\int J(t)dt = \frac{J_0\tau_s}{\hbar} = (2n+1)\pi; n = 0, 1...$$

Entonces la evolución del sistema vendrá dada por el operador unitario:

$$U(t) = T(e^{\frac{i}{\hbar}\int_0^t H_s(\tau)d\tau}) \tag{7.12}$$

Esta evolución corresponde al *operador de intercambio,* U_{sw} , que intercambia ambos electrones.

Lo interesante está en esta igualdad:

$$U_{XOR} = e^{\frac{i\pi}{2}S_z^1} e^{\frac{i\pi}{2}(S_z^1 - S_z^2)} U_{sw}^{1/2} \tag{7.13}$$

Aquí hay dos operaciones diferentes, $U_{sw}^{1/2}$, donde U_{sw} es el operador de intercambio, y una rotación de un único qubit, $e^{i\pi S_z^1}$. El resultado, nada menos que la aplicación de una puerta XOR. El estudio de la implementación de funciones con qubits, bajo este esquema (y otros parecidos) se reduce entonces al del mecanismo de acoplamiento J(t), y de su control externo[13].

7.3.5. Cavidades ópticas de alta calidad.

El problema ahora no es el procesamoiento de la información, sino la comunicación. En una cavidad optica de calidad suficiente podremos tener acoplamientos fuertes entre un único ión y un modo de radiación electromagnética. Esto permitiría aplicar operaciones entre los modos del campos y los iones, cosa que por ejemplo puede usarse para comunicar trampas iónicas.

Para acoplar los iones en una trampa sencilla podemos utilizar campos electromagnéticos, como alternativa al método de los fonones, descrito en 7.3.1.

8 Conclusiones.

Llegados a este punto queda poco por decir, más alla ed hacer un a simple recapitulación sobre todo lo que hemos visto hasta ahora.

8.1. Lo que el QC es capaz de hacer.

Hemos visto un capítulo completo dedicado a operaciones que diferencian al computador clásico del computador cuántico. Sabemos que es capaz de aprovechar la posibilidad de *superposición* en forma de *paralelismo cuántico*. Por este motivo decidí en su momento que me gustaba más hablar de procedimientos que sobre algoritmos. No obstante, pese al manejo simultáneo de mucha información y a que las etapas en la computación no son las mismas que las de un computador tradicional, no deja de ser cierto que realizamos con él una serie de operaciones secuencialmente, por mucho que algunas de sus etapas se lleven a cabo de golpe. Al fin y al cabo, basta decidir que es una etapa y que no lo es para que la idea tradicional de algoritmo siga siendo válida.

Lo realmente importante que aprendimos fue a distinguir entre diferentes tipos de problemas, al tiempo que vimos que si bien tanto un tipo de computadores como otros podían resolver los mismos, para algunos dominios particulares el computador cuántico resultó ser mucho más eficiente. Un QC puede resolver problemas del dominio np en un tiempo polinómico en las dimensiones de éstos, mientras que un computador tradicional lo hará en un tiempo exponencial. Lo normal es que para hacerlo el QC aproveche su capacidad de explorar rápidamente el espacio de soluciones.

Tambien hay problemas fuera del dominio np que el computador cuántico puede resorver en un tiempo polinómico respecto a la entrada, mientras que un computador tradicional lo hará en un tiempo tambien exponencial.

Las operaciones que distinguen a los QC vimos durante el trabajo se apoyan en una operación conocida como transformada de Fourier discreta, que aparece en 4.4. Esta operación, a su vez, descansa en la interferencia entre componentes de una combinación lineal de estados que sólo es posible en el ámbito de la mecánica cuántica, gracias al principio de superposición. Así es como el principio de superposición nos permite condensar en un único estado toda una serie de alternativas, obteniéndose la propiedad conocida como *paralelismo cuántico*, que es básicamente lo que permite reducir drásticamente el número de pasos necesarios para determinadas tareas.

El método de factorización de Shor es un ejemplo de aplicación de la potencia de la mecánica cuántica a un problema tradicionalmente considerado como inabordable. Buscar factores primos en números grandes requería explorar los enteros inferiores a éstos, y cada vez que añadimos un dígito nuevo el tamaño del problema se multiplica por diez. Este es un típico problema donde el número de pasos crece exponencialmente con el tamaño de la entrada. Un computador cuántico enfrentado al mismo problema y provisto de la herramienta facilitada por Shor sólo necesitará un tiempo polinómico en el tamaño de la entrada. Hablando en orden de magnitud, si duplico el número de dígitos duplicaré el tiempo de computación. Basta imaginar lo que ocurriría si duplicásemos el número de dígitos en la entrada de un computador tradicional para que nos demos cuenta de la potencia que hemos conseguido. La base de la criptografía moderna descansa sobre la supuesta inabordabilidad del problema de factorización, como expliqué en el apartado 3.2.5.

La aplicación de la criptografía y la necesidad de la conservación de secretos ha creado muchas espectativas sobre el algoritmo de Shor, pero no hace que deje de ser la solución a un problema muy particular. De hecho, hoy se utilizan tambien números primos en la generación de números pseudoaleatorios, de gran utilidad por ejemplo en simulación, pero un computador cuántico nos permitiría obtener directamente números aleatorios puros.

Más interesante a mi entender es la generalización que permite encontrar el periodo de una función, pues encontrar correlaciones en conjuntos de datos puede reducirse a esto, y a partir de nevos patrones podríamos aprender mucho sobre la naturaleza. A veces parece que olvidamos que las computadoras son una herramienta, y al hacer de la computación una ciencia no nos damos cuenta de que en realidad su cometido es asistirnos en nuestra labor científica en todas las disciplinas (incluyendo, claro está, la propia computación, pero tambien la física). En relación a este tema hablé en 2.2.2, donde traté sobre si un computador cuántico sería o no más adecuado para modelizar la naturaleza. La respuesta vino en forma de conjetura aceptada en general, conocida como *principio de Church-Turing*.

El método de búsqueda de Grover, al igual que el método de Shor, aprovecha tambien el paralelismo cuántico, pero permite resolver una variedad de problemas más amplia. Tal como aparecía en las referencias sobre las que trabajé, parecía como si encontrar un método de búsqueda sólo sirviese para encontrar números de teléfono, mientras que buscar factores primos parece algo así como el santo grial de la computación. Sin embargo, los listines de teléfono ya vienen ordenados alfabéticamente, y una persona puede en muy pocos pasos resolver ese problema sin necesidad de un computador cuántico. Sin embargo, la posibilidad de analizar rápidamente un espacio de soluciones y detectar elementos en él resulta útil para resolver ecuaciones complejas, así como para la toma de decisiones y el tratamiento de problemas difíciles de formalizar. La búsqueda de factores primos en enteros grandes no parece que sea tan importante en el estudio de fluídos turbulentos ni en nuestra vida cotidiana.

Dado que de todas formas apareció el método de descomposición, se hizo urgente asegurar otra manera de proteger la información. La respuesta a este problema se conoce como criptografía cuántica, y además no reside en una característica tecnológica, como puede ser una cierta limitación, superable en el transcurso del tiempo, sino en un principio mismo de la física, el *teorema de no clonación*. La criptografía cuántica es ya hoy una realidad, hasta el punto de que dentro de pocos años será de uso generalizado, al menos para algunas instituciones.

Junto con la criptografía cuántica vimos lo que era el *teletransporte*. Nos tropezamos con un problema asociado al no localismo de la función de onda, una incompatibilidad con el principio de la relatividad especial que establece que no se puede propagar una señal a velocidad mayor que la de la luz. La deslocalización de la función de estado hace posible la existencia de estados entrelazados que conectan a observadores separados en el espacio. La alteración de una parte del sistema entrelazado provoca un efecto instantáneo en la otra, y este efecto alcanza cualquier posición. Tal vez el problema esté en qué es una señal y qué es informción, pero los resultados experimentales no dejan lugar a dudas: el teletransporte es un realidad.

8.2. Lo que el QC no es capaz de hacer.

Vimos durante el trabajo que el computador cuántico es capaz de sacar problemas de sus dominios de complejidad, hasta el punto de hacer viables operaciones que no lo eran hasta ahora. La palabra clave es *viables*, pues siempre hablamos de tiempo de computación. No nos preocupamos de si el problema se podía resolver en un minuto o en un tiempo superior a la edad del universo, sino simplemente de si se podía resolver. Y vimos que existen problemas que no se pueden reslver. En particular vimos un ejemplo en 3.2.4. Existe toda una clase de problemas de este tipo, y en su momento aprendimos que es imposibilidad no es en absoluto circunstancial, fruto tal vez de una falta de estrategias, o de un mal planteamiento del propio problema, sino que era en sí misma esencial. Nuestra conclusión entonces fue que cualquier problema que pudiese ser resuelto en un computador clásico lo sería tambien en un QC, y vicebersa. Insisto en que no nos preocupamos de en cuánto tiempo.

Por otra parte, aceptamos como válido el *principio de Church-Turing,* que nos dice que si un sistema es realizable en la naturaleza, entonces tambien lo es en un computador con un número finito de estados internos. De esta forma, no esperamos encontrar en la naturaleza ningun sistema que no sea modelizable en un QC, de modo que parece que el problema de la simulación sí quedaría resuelto. A la luz de esto, los problemas sin solución esperamos que sean, por así decirlo, un poco extraños.

Por otra parte, hay numerosos problemas para los que un computador cuántico no aporta ninguna ventaja sobre uno clásico. Uno de estos problemas es

la evaluación de funciones, sin ir más lejos, pero en realidad slos problemas de este tipo son la mayoría, de modo que a un usuario doméstico puede resultarle muy poco interesante adquirir un computador cuántico para su trabajo doméstico, cuando un computador electrónico cubre todas sus necesidades con un coste previsiblemente mucho más bajo. Por el mismo motivo, a un fabricante puede resultarle muy poco atractivo faricar QC para usuarios domésticos, y es previsible que estos no se lleguen a ver nunca. Tambien es cierto que toda novedad en computación se dirige primero al sector profesional, y que un computador dedicado a juegos en nuestros hogares tiene la potencia de cálculo de un supercomputador de hace pocos años. Esto es una observación que puede hacerse, aunque tambien es cierto que en este caso no hablamos de fuerza bruta a nivel de potencia de cálculo, sino de un nuevo diseño, orientado a la resoluciónde una clase particular de problemas, y a no ser que descubramos uno de particular interés para todos, lo más probable es que los computadores cuánticos no salgan del ámbito de la simulación y poco más.

8.3. ¿Seremos capaces de construir un QC?

Los últimos trabajos en relación a las posibles realizaciones del QC hacen pensar primero que de momento el día en que veamos un procesador de propósito general capaz de explotar las características de la mecánica cuántica está aún lejos, y por otro que realmente no hay motivo para suponer que no llegará. Ya se han realizado experimentos en los que se podía controlar un par de qubits, y hay varios esquemas diferentes tanto a nivel de arquitectura como a nivel físico. De hecho, aún no hay un modelo estándar para el que los investigadores se hayan puesto de acuerdo en trabajar. Cada línea de desarrollo soluciona problemas asociadas a las demás, al tiempo que tropieza con nuevos problemas. Sobre los problemas que aparecen ahora cabe decir varias cosas:

1. Un problema a menudo se resuelve cambiando de perspectiva, y no insistiendo en una idea que puede no ser la más adecuada. El hombre intentó volar primero batiendo las alas, pues era un esquema que estaba acostumbrado a ver funcionar, pero consiguió volar cuando creó algo mucho más sencillo: el globo. La computación electrónica no avanzó definitivamente hasta que los tubos de vacío fueron superados, y los dispositivos de estado sólido que los sucedieron tenían muy poco que ver con ellos. A veces solucionar un problema requiere un nivel de distanciamiento relativamente alto, y especular sobre si un problema carece de solución suele ser resultado de esta falta de distanciamiento.

2. La definición de un problema se hace mediante premisas, y la premisa para tratar la construcción del QC en la actualidad son una serie de tecnologías que están dejando paso a otras nuevas (como puede ser la computación biomolecular [7]).

3. La tercera razón, y la más importante de todas, es la siguiente:

   No existe (al menos no conocemos) ningun principio en la naturaleza que prohiba la existencia de los procesadores cuánticos de la información.

De hecho, procesadores cuánticos de información hay (la propia naturaleza) y tambien se han construído, aunque a escala muy pequeña. Es el problema de la escala el que más nos preocupa ahora, pero no hay razón para pensar que este problema no pueda ser resuelto.

9 Apéndice: técnicas mencionadas en el trabajo.

9.1. Preparación del procesador.

9.1.1. Bombeo óptico.

9.1.2. Enfriamiento láser.

9.2. Técnicas de medida.

9.2.1. Salto cuántico.

9.2.2. Shelving electrónico.

9.3. Control de la evolución.

9.3.1. Spin-echo.

Bibliografía

[1] "Quantum Computing", Andrew Steane, Department of Atomic and Laser Physics, University of Oxford, July 1997

[2] "Error detecting and error correcting codes", Hamming R. W., Bell Syst. Tech, 1950

[3] "Coding and information theory", 2nd ed. (Pretience Hall, Englewood Cliffs), 1986

[4] "Quantum Information Theory", Charles Bennett, Peter W. Shor. September 4, 1998

[5] "Quantum computing", Peter Shor, 1991

[6] "Elementary Gates for Quantum Computation", A. Barenco, C.H. Benett, R. Cleve, D. DiVicenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, H. Weinfurter, Phys. Rev. A vol 52, pp 3457-3467, 1995

[7] "Alternative Computational Models: A comparison of Biomolecular and Quantum Computation", John H. Reif, (FST&TCS98), 120-121, December, 1998

[8] "Quantum Computation", Samuel L. Braunstein

[9] "Scheme for Reducing decoherence in quantum computer memory", Peter Shor, Phys. Rev. A 52 (1995), pp 2493-2496.

[10] "Algorithms for Quantum Computation: Discrete Logarithms and Factoring", Peter Shor, In: Proceedings, 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, November 20-22, 1994, IEEE Computer Society Press, pp 124-134.

[11] "A Bibliography of Quantum Cryptography", Gilles Brassard, December 3, 1993

[12] "Dinamic Behaviour fo Quantum Cellular Automata", P. Douglas Tougaw and Craig S. Lent, Journal of Applied Physics.

[13] "Quantum Computing And Quantum Communication With Electrons in Nanostructures", Daniel Loss, Guido Bukard and Eugene V. Sukhorukov.

[14] "Quantum Computing With Trapped Ions, Atoms and Light", A. M. Steane and D. M. Lucas, October 30, 2001.

[15] "Speed of Ion Trap Quantum Information Processors", A. Steane, C. F. Ross, D. Stevens, A. Mundt, D. Leibfried, F. Schmidt-Kaler, R. Blatt, November 7, 2001.

[16] "Quantum Information Processing Using Quantum Dot Spins and Cavity-QED", A. Imamo$\bar{g}$lu, D. D. Awshalom, G. Bukard, D. P. DiVicenzo, D. Loss, M. Sherwin, A. Small, November 6, 2001.ss.

[17] "Multiple Particle Interference and Quantum Error Correction",A Steane, proc. R. Soc. London A, vol 452, pp 2551-2577, 1996 (quant-ph/9601029).

[18] "Mixed State Entaglement and Quantum Error Correction", C.H. Bennett, D.P. DiVicenzo, J. Smolin & W.K. Wooters, Phys. Rev A, vol 54, pp 3824-3851, 1996 (quat-ph/9604024).

[19] "Perfect Quantum Error Correction Code", R. Laflamme, C. Miquel, J.-P. Paz & W.H. Zurek, Phys. Rev Lett. 77, 198-201, 1996.

[20] "Nonlocality Criteria for Quantum Teleportation",N. Gisin, Phys. Rev Lett. A vol. 210, pp 151-156, 1996.

[21] "Cellular Automata Machines: A New Enviroment for Modelling", T. Toffoli & N. Margolus, MIT Press, Cambridge, Massachusets, 1987.