Universidad Nacional Autónoma de México

Facultad de Ciencias

Aleatoriedad Algorítmica y Generación de Números al Azar Usando Sistemas Cuánticos

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
FÍSICO

P R E S E N T A:
ALDO FERNANDO GUADALUPE SOLIS LABASTIDA

DIRECTOR DE TESIS :
DR. JORGE GUSTAVO HIRSCH GANIEVICH

2015

Ciudad Univeritaria, D. F.

Hoja de datos jurado

1. Datos del Alumno
Solis
Labastida
Aldo Fernando Guadalupe
56-34-82-10
Universidad Nacional Autónoma de México
Facultad de Ciencias
Física
306035609

2. Datos del tutor
Dr.
Jorge Gustavo
Hirsch
Ganievich

3. Datos del sinodal 1
Dr.
Carlos
Villarreal
Luján

4. Datos del sinodal 2
Dr.
Alfred Barry
U'ren
Cortés

5. Datos del sinodal 3
Dr.
Jesús
Garduño
Mejía

6. Datos del sinodal 4
Dr.
Isaac
Pérez
Castillo

7. Datos del trabajo escrito.

Aleatoriedad algorítmica y generación de números al azar usando sistemas cuánticos

67 p

2015

Agradecimientos

Agradezco a mis padres, Fernando y Cristina, y a mis hermanas, Karina y Giselle por todo el apoyo y consideración que me han dado.

Agradezco a mi asesor el Dr. Jorge Hirsch por guiarme a través de este trabajo y formarme como físico.

Agradezco al Dr. Alfred U'Ren y al equipo del Laboratorio de Óptica Cuántica del ICN, en partícular al M. en C. Alí Angulo, por el apoyo recibido para la realización experimental de este trabajo.

Agradezco al Conacyt por el apoyo económico para la realización de este trabajo con la beca para titulación del proyecto SEP-Conacyt 166302.

Índice general

Introducción

Todos los días alrededor del mundo las personas nos enfrentamos a algún tipo de situación o fenómeno que consideramos tiene "azar" sin embargo dar una caracterización global de este concepto es algo difícil.

Con una evolución tecnológica tan grande como la que se ha presentado en los últimos tiempos, muchos aspectos de la vida cotidiana han sido llevados a este nuevo ámbito tecnológico como los juegos de azar, donde usando números al azar se logra el carácter de impredictibilidad en el juego. Por otro lado el azar al que estamos acostumbrados ha influido en el diseño de nuevas tecnologías como la criptografía o los métodos de Monte-Carlo donde se requiere de principio una buena fuente de números al azar.

Sin embargo introducir el azar en este mundo no es una tarea fácil, escribir un programa que logre generar números al azar ha demostrado ser una tarea difícil pues si bien se pueden alcanzar buenos resultados para ciertas aplicaciones para otras esta solución ya no funciona.

Es ahí donde buscamos que el azar ya conocido del mundo se introduzca a la tecnología. Los números aleatorios generados por hardware han sido una alternativa que ha tomado mucha fuerza debido a sus buenas propiedades como generadores de números al azar, sin embargo esto presenta un problema tecnológico, en primer lugar hacer una interfaz entre el fenómeno aleatorio y la generación de los números, y por otro lado los números generados de esta manera pueden no tener las propiedades que deseamos de una fuente de números al azar.

Finalmente, de nuevo por el gran avance tecnológico, hoy día se pueden hacer de manera sencilla mediciones en sistemas cuánticos que antes sólo era posible con arreglos experimentales muy complejos, lo que representa una nueva posibilidad para la generación de números al azar. Los generadores de números al azar cuánticos tienen, según la teoría cuántica, una impredictibilidad intrínseca que les otorga una gran ventaja frente a otros tipos de

generación, lo anterior no sin problemas pues debe asegurarse este carácter a través de un estudio riguroso y con los problemas técnicos y teóricos que esto conlleva.

Para tener una aproximación fundamental con las ideas de azar en este trabajo nos apegamos a una definición completamente matemática del azar que está basada en una formalización de la noción de información y rescata algunos aspectos de estos conceptos que se dejan de lado en algunos acercamientos. En el primer capítulo se expone este concepto de aleatoriedad de manera intuitiva así como algunas de sus características principales.

En el segundo capítulo relacionamos la definición formal del primer capítulo con un método aplicable a datos obtenidos de sistemas reales, esta relación está dada por un teorema que nos ofrece una condición para probar la aleatoriedad.

Como primera aplicación de este método, en el tercer capítulo obtenemos números aleatorios de un sistema electrónico.

En el capítulo cuarto señalamos algunos puntos finos entre lo incomputable y la aleatoreidad en mecánica cuántica.

Como segunda aplicación de este método llevamos estas ideas a un sistema cuántico donde utilizamos tiempos de llegada entre fotones como base para la generación.

Finalmente revisamos las propuestas más importantes de generación de números al azar que están relacionados con esta técnica.

Capítulo 1

Teoría Algorítmica de la Información (TAI)

La información es un concepto que, debido al desarrollo tecnológico actual, es intuitivo para muchas personas. Hoy día palabras como bit, byte, etc. forman parte del vocabulario coloquial, y frases como:"mi conexión es de 5 megabytes" o "Un disco de 5 terabytes" se escuchan frecuentemente en distintos ámbitos. Si bien todo mundo habla naturalmente de cantidades de información vale la pena hacerse las siguientes preguntas ¿Cómo se mide la cantidad de información? ¿Cómo puedo saber cuanta información tiene una foto que tomé o incluso el texto que estoy escribiendo en este momento? ¿A todos los objetos que puedo "meter" en una computadora les puedo asociar una cantidad de información?¿Puedo asociarle una a aquellos que no puedo "meter"?

Tomemos la segunda pregunta, supongamos una imagen en una computadora, preguntemos al sistema operativo que cantidad de información tiene:

```
$ du -b imagen.jpg # cantidad de bytes del archivo imagen.jpg
465448 imagen.jpg
```

el sistema operativo me dice que ocupa 465,448 bytes y apliquemos un cambio de formato a la imagen, donde la resolución de la imagen es la misma antes y después de la transformación, veamos la cantidad de información que ocupan:

```
$ jpegtopnm imagen.jpg > imagen.bmp
$ du imagen.bmp imagen.jpg
15116561 imagen.bmp
465448 imagen.jpg
```

Como podemos ver las cantidades no coinciden y difieren bastante relativamente a sus magnitudes.

El ejercicio anterior puede parecer una simpleza pues el hecho de que estas cantidades no coincidan es común y estamos acostumbrados a él, sin embargo, ambos archivos al abrirlos muestran la misma imagen con la misma resolución, entonces ¿cómo puede medirse la información de una imagen si, como vemos, dependiendo del formato tendrá una u otra cantidad de información?.

Una respuesta posible a lo anterior es que el archivo con más bytes tiene "información innecesaria" es decir que no es indispensable para poder reconstruir la imagen en la pantalla pero está ahí por otras razones, lo cual nos lleva a un buen camino para definir la cantidad de información. Para poder diferenciar entre la información que es "necesaria" y la que no busquemos aquel archivo que tenga la mínima cantidad de bytes y propongamos esa como la cantidad de información de ese objeto.

1.1. La Mínima Descripción

Si bien las ideas anteriores pueden parecer sencillas, al intentar dar un formalismo matemático a ellas se debe de usar una de las herramientas matemáticas más potentes del siglo pasado: la teoría de la computación. En el ejercicio anterior tenemos una computadora dada y usamos programas específicos que interpretan el archivo como imagen y lo plasman en la pantalla, pero si vamos a buscar la mínima cantidad de información debemos de tomar en cuenta cualquier posible computadora y programa que pudiera usarse, por tanto debemos pensar en una máquina general para que nuestras conclusiones sean generales, aunque sabemos que en la realidad todas las máquinas tienen limitaciones de memoria, velocidad, etc.

Esta situación es análoga a la de la termodinámica, donde las máquinas de Carnot, o en general las máquinas reversibles, no pueden ser superadas en cuanto a eficiencia por ninguna otra máquina, así que muchas de sus propiedades se vuelven una barrera para las máquinas "reales".

En nuestro caso, el papel de la maquina de Carnot es un concepto matemático conocido como máquina de Turing desarrollado en la década de los 30's por Alan Turing, que se identifica intuitivamente con una computadora de memoria infinita y por tanto en cuanto almacenamiento es mayor que cualquier computadora realizable. Por otro lado, el teorema de Carnot tiene

su análogo en la existencia de una máquina universal de Turing y lo que tal vez sea el pilar de la teoría de la computación la Tesis Church-Turing.

1.1.1. Máquinas de Turing

Las máquinas de Turing [1] fueron introducidas en 1936 [Tur37] como una formalización alternativa [2] del concepto de "procedimiento efectivo" que estaba ya establecido en la época. Turing en su artículo original da dentro de los primeros párrafos una ligera motivación a esta definición: hacer un modelo de un hombre que calcula, este es el camino que usamos en lo siguiente.

Para empezar el hombre que calcula debe tener un papel donde escribir, algo para escribir y algo para borrar. Como no podemos poner una cota a priori al papel que va a usar vamos a permitir una cantidad infinita de papel, con la restricción de que el papel debe estar en blanco al principio del cálculo o en su defecto, sólo tener una parte finita escrita, esto debido a que en todo momento pedimos que los recursos sean finitos, y por otro lado en cualquier cálculo "real" se tendrá sólo una parte finita escrita.

Respecto al borrado y la escritura, el hombre usará un alfabeto, es decir, un conjunto de símbolos que es finito [3] y usará las combinaciones de estos símbolos para llevar a cabo su cálculo, además sólo podrá escribir, leer y borrar los símbolos de uno en uno.

En todos los sistemas de escritura la posición que cada símbolo toma respecto al todo influye en el significado del todo, generalmente se tiene un sistema de cuadros donde el sentido de la escritura puede ir de izquierda a derecha o de arriba hacia abajo etc, tomemos como ejemplo una operación matemática sencilla como la suma

$$\begin{array}{cccc} 1 & 6 & 8 & 6 \\ + & 2 & 5 & 0 \\ \hline 1 & 9 & 3 & 6 \end{array} \tag{1.1}$$

[1] En esta parte introducimos las máquinas de Turing más no su formalismo matemático si se desea consultarlo puede ir a [Dav53] donde se encuentra una excelente exposición con todo el rigor matemático debido.

[2] La primera formalización bien establecida se debe a Church un año antes, esta es equivalente a la de Turing

[3] A veces se puede pensar que esto es un poco contradictorio con que los números son infinitos o los enunciados sean infinitos, pero hay que recordar que en cualquier base que se representen los números naturales sólo se usa un finito de ellos por ejemplo en el sistema decimal sólo se usa 0,1,2,3,4,5,6,7,8,9 para representarlos a todos.

aquí vemos como la posición de cada dígito influye en su significado, por esto dividiremos el papel en casillas donde cada una puede tener un y sólo un símbolo(tomamos el espacio en blanco como símbolo también) como se muestra en la figura 1.1. Finalmente reducimos esta cuadrícula a una sola dimensión, es decir una sola hilera de cuadros, debido a que la relación vertical entre los símbolos se puede traducir a una horizontal, por ejemplo se puede hacer la suma con los términos escritos en el mismo renglón.

Resumiendo tenemos una cinta con casillas y un hombre que puede leer, escribir y borrar en ella, hay sólo un número finito de símbolos que puede escribir que llamamos el alfabeto, significa que la cinta sólo tendrá en todo momento una sección finita "escrita" que puede incluir símbolos blancos y que esta sección tendrá a su derecha e izquierda sólo símbolos blancos. A esta sección escrita antes de que el hombre (o la máquina) empiece a escribir en ella la llamamos la entrada de la máquina y sí la denotamos por α el número de símbolos que tiene esa entrada (incluyendo espacios en blanco) lo denotamos por $|\alpha|$. A esta misma sección en el caso en que el hombre (o la máquina) "acaban de escribir" se denota por $M(\alpha)$ donde M es una máquina de Turing. [4]

El hombre por otro lado estará restringido sólo a calcular, es decir no puede hacer ningún proceso "creativo", por ello se hará una abstracción y limitación de lo que su cerebro puede hacer; para esto se usa el concepto de estado mental, su cerebro sólo podrá tener una cantidad finita de estados mentales. [5] El término de estado mental puede ser confuso, para darle mayor significado pongamos el caso de una máquina que calcula usando engranes, aquí cada estado mental se identifica con cada configuración interna de la máquina, es decir la configuración describiría el estado de cada engrane en la máquina y análogamente con cualquiera sea su construcción.

Finalmente el procedimiento de cálculo será leer una casilla y dependiendo del estado del hombre(o la máquina) se escribirá el nuevo símbolo, se llevará la máquina al siguiente estado y finalmente se moverá a la casilla que esté a la derecha o a la izquierda. Además agregaremos un estado extra, que normalmente se denota como H. Este estado significa que el cálculo ha

[4] Las máquinas no siempre paran, por ejemplo una máquina que siempre se mueva a la derecha sin importar que símbolo lea. Como la cinta es infinita esa máquina nunca parará. En estos casos se suele definir $M(\alpha) = \text{ínf}$

[5] Esto es para mantener la finitud en todos los aspectos de la máquina, incluso la cinta si bien es infinita al sólo estar escrita en un finito de casillas sería más fácil considerarla un infinito "en potencia"

terminado y al llegar a él la máquina (o el hombre) se detiene.

Al conjunto del alfabeto, la cinta, los estados posibles y la función que determina los símbolos a imprimir, nuevo estado y movimiento en base a los anteriores se le conoce como Máquina de Turing y, como se mencionó en un principio, es un modelo formal de las capacidades de cálculo de un hombre o máquina. Estas máquinas entran bien en el concepto intuitivo de lo que es un algoritmo, es decir, difícilmente alguien dirá que lo que una máquina hace no es algorítmico.

De lo anterior vemos que cada máquina de Turing sigue un algoritmo, pero si buscamos que estas máquinas sean la formalización de este concepto necesitamos que cada algoritmo tenga una máquina de Turing que lo pueda llevar a cabo, es decir que tengamos una relación uno a uno entra las máquinas y los algoritmos. Esto naturalmente no se puede probar debido a que de principio no tenemos una definición de algoritmo así que debe tomarse como un hecho que está respaldado por los resultados que se obtienen, esta es la llamada tesis de Church-Turing. [6]

Otro de los grandes resultados de la teoría de la computación es la existencia de máquinas universales, una máquina universal se refiere a aquella máquina que dándole una entrada adecuada puede "imitar" el resultado de cualquier otra máquina. Esto es de hecho natural si se piensa en la siguiente pregunta ¿Seguir un algoritmo es algorítmico? Una vez que se entiende el lenguaje en que está escrito el algoritmo y se tiene la capacidad de realizar cada una de las acciones básicas de él ya no se requiere " creatividad" alguna para llevarlo a cabo, sólo es necesario seguirlo. Intuitivamente podemos decir que seguir un algoritmo es algorítmico [7] por lo tanto si la tesis Church-Turing es cierta debe de haber una máquina que al darle como entrada la descripción de una máquina(algoritmo) en cierto lenguaje, lo siga y reproduzca sus resultados.

[6] En realidad el tema es mucho más complicado que lo anterior dicho, hay mucha investigación en lo que se suele llamar la hypercomputación, buscando formas de calcular, generalmente por medios físicos, cosas que son incomputables en el sentido de Turing, hasta ahora si bien hay muchas propuestas ninguna ha logrado librarse de fallas, uno de los intentos más impresionantes esta dado por Tadaki en [Kie03] donde usando estados de número y cómputo adiabático se propone solucionar una ecuación diofántica que debido al trabajo de Matijasevic, Davis, Putnam y Robinson [Mat93] se sabe que encontrar sus raíces es incomputable

[7] Mientras nos mantengamos en ciertas barreras, por ejemplo podemos ser capaces de contar hasta ocho pero no podemos hacerlo en la décima parte de un segundo o tal vez no lo podamos hacer un millón de veces seguidas.

De las máquinas universales mencionamos aquí una definición debido a que juegan un papel fundamental para la TAI.

Definición 1.1 *Sea M una máquina de Turing, decimos que M es universal si para toda máquina N con entrada α existe una entrada α' tal que $N(\alpha) = M(\alpha')$ y además $|\alpha'| \leq |\alpha| + c$ donde c es una constante que sólo depende de las máquinas*

En esta definición se piden las condiciones anteriores y además se pide que $|\alpha'|$ no sea mucho mayor que la entrada original debido a que en principio sólo se debe agregar la descripción de la máquina y esta no depende de la entrada α. [8]

1.1.2. El Mínimo Programa

Usando lo anterior podemos escribir las ideas del principio. Tomemos algo más sencillo que una imagen,

Definición 1.2 *Sea a una cadena de símbolos sobre un alfabeto A y pensemos en una máquina universal M también sobre A. Debido a que la máquina es universal hay un programa que al terminar deja en la cinta solamente la cadena a. [9] Definimos la información en la cadena a, $K_M(a)$, como la longitud de la entrada más pequeña cuya salida es la cadena a (según orden lexicográfico en caso de longitud igual). Esta entrada se denomina el Programa mínimo y lo denotamos como $a*$.*

Hay que notar que la definición anterior depende de la máquina elegida y por tanto la información de un objeto dependerá de la máquina que se use para describirlo. Más aún, una cadena que tiene mucha información para una máquina puede tener muy poca en otra ¿Cómo podemos defender esta definición ante lo anterior?

La principal defensa se basa en la siguiente idea: supongamos 2 máquinas universales respecto de las cuales se miden las complejidades, como la primera

[8] Nunca hay que perder de vista el alfabeto que maneja la máquina pues si bien la mayoría de las ocasiones se puede hacer una "traducción", esta traducción tiene que llevarla a cabo otra máquina que maneje ambos alfabetos y esto le quita un poco la idea de que puede llevar a cabo todos los algoritmos, estos detalles se omiten por simplicidad.

[9] Se puede pensar en el programa que sólamente imprime la cadena a y termina. Debido a que estamos pidiendo a M como universal hay una entrada para M que imita este mismo comportamiento.

es universal, hay una entrada que le permite "imitar a la segunda" de manera independiente a la entrada que recibe la segunda máquina, digamos que esta entrada tiene una longitud de N bits, si a es una cadena y tiene k bits de información según la máquina 2 en la máquina 1 el programa que imita a la máquina 2 con entrada 1 tiene una longitud de $N + k$ bits por lo tanto la información según la máquina 1 no puede ser mayor a $N + k$ donde N es una constante que sólo depende de las máquinas.

Como el argumento anterior puede repetirse intercambiando papeles, obtenemos el siguiente resultado:

Teorema 1.1 *Sean M y M' dos máquinas universales y sea "a" una cadena. Existe una constante $N_{M,M'}$ que sólo depende de las máquinas tal que:*

$$|K_M(a) - K_{M'}(a)| < N_{M,M'} \tag{1.2}$$

Este resultado garantiza que la información medida por 2 máquinas será distinta en general pero en cuanto midamos un objeto cuya información en comparación con $N_{M,M'}$ sea grande, las máquinas darán resultados parecidos.

1.2. Aleatoriedad e información

Si bien lo anterior parece una buena representación de varios conceptos intuitivos de información, ¿qué relación guarda con la aleatoriedad? La aleatoriedad o el azar son conceptos que, al igual que la información, son comunes en el día a día y sin embargo no tienen una definición precisa, si bien la teoría de la probabilidad y de los procesos estocásticos tienen gran éxito caracterizando los fenómenos que consideramos aleatorios, no dan un fundamento de la aleatoriedad.

En la teoría clásica de la información, el concepto de información se construye usando la siguiente ecuación:

$$I_{shannon} = log_2 \frac{1}{P} \tag{1.3}$$

donde $I_{shannon}$ es la información que provee un evento con probabilidad P de ocurrir y está medida en bits. Hay que notar que la definición anterior fundó el campo de la teoría de la información [10] Esta es de hecho una definición

[10] El articulo original de Shannon es la referencia [Sha48]

muy práctica que logró sentar un entorno para explicar muchos fenómenos de la comunicación. Desde otro punto de vista no tan pragmático y mucho más crítico vemos que usar la probabilidad como fundamento el concepto de información no es deseable debido al siguiente argumento: Supongamos una fuente de información que saca alternadamente unos y ceros

$$01010101010101...$$

entonces la probabilidad de encontrar un '0' o '1' es la misma entonces según la ecuación 1.3 la información estaría dada por

$$I_{shannon} = log_2 \frac{1}{\frac{1}{2}} = log_2 2 = 1$$

es decir en la cadena nos encontramos con un bit de información por cada símbolo otra con un patrón diferente

$$00011010010111...$$

tiene también igual probabilidad así que también tiene 1 bit por símbolo.

¿Estamos dispuestos a respaldar una teoría que ponga igual valor de información a las cadenas anteriores? [11]

En el caso de la teoría algorítmica la definición de aleatoriedad y de información es completamente distinta, pero increíblemente intuitiva, se basa en el hecho de que siempre que encontremos una cadena con "patrones" no nos resultará aleatoria, por ejemplo:

$$100101010010101001010..$$

tiene claramente un patrón, incluso podemos pensar en patrones no repetitivos: [12]

$$01011011101111$$

[11] Es muy importante notar que Shannon no trataba de dar un fundamento de la información sino de la comunicación, como se puede notar por el nombre del artículo[Sha48]. El nombre "teoría de la información" fue asociado después, esto lleva muchas veces a confusiones debido que el concepto intuitivo de información muchas veces no coincide con lo que la teoría intenta describir.

[12] No todos los patrones son repetitivos pero el estudio de ellos lleva a varias propiedades relacionadas con la aleatoriedad, todos estos se tocan en el análisis de la normalidad de Borel en relación con la aleatoriedad de las cadenas.

en general siempre que encontremos un patrón será más factible decir que la cadena no es aleatoria. Si buscamos cadenas que sean realmente aleatorias las cadenas no deben tener patrones y de ello sale una pregunta muy natural ¿Qué es un patrón? Para formalizar la idea de patrón haremos uso de la teoría de la computación e identificaremos patrón con algoritmo. Un problema de lo anterior es que cualquier cadena finita es algorítmica, por lo que identificar patrón con algoritmo necesita algunas precisiones. En general no pedimos solamente que haya un algoritmo que la pueda reproducir, sino que este se pueda escribir con menos símbolos que la cadena original. Esto se identifica con el proceso intuitivo de compresión de información y proponemos que si la cadena no se puede comprimir más, debe ser aleatoria.

Definición 1.3 *Sea M una máquina y c un número natural, decimos que una cadena a es c-aleatoria si se cumple que*

$$K_M(a) > |a| - c \tag{1.4}$$

y en caso de ser 0*-aleatoria decimos simplemente que es aleatoria.*

La definición de aleatoriedad que da la teoría algorítmica de la información es uno de sus puntos más fuertes. Si bien una teoría generalmente se mide por los resultados que obtiene, esta definición da un fundamento de la aleatoriedad no en un sentido práctico sino descriptivo, y realmente muy apegado a la visión intuitiva que se tiene de este concepto.

Como se mencionó respecto a la cantidad de información, esta depende de la máquina, por lo tanto la aleatoriedad de una cadena depende de la máquina que se use de referencia, sin embargo entre más sea la aleatoriedad de la cadena menos importante será la constante, finalmente en el límite cuando la aleatoriedad de la cadena es infinita obtenemos un excelente resultado: si una cadena infinita es aleatoria entonces es aleatoria para todas las máquinas.

Las cadenas de aleatoriedad infinita no son tan comunes, de hecho para construir alguna se debe recurrir a una formulación más avanzada de la teoría algorítmica de la información que se conoce como libre de prefijos, fue presentada en los 70's principalmente por Gregory Chaitin.

Las cadenas aleatorias por otro lado son muy comunes por un argumento muy simple: pensemos en el alfabeto binario, las cadenas $1, 0$ son aleatorias pues no hay cadena más corta que ellas, de las 4 cadenas de longitud 2 $(00, 01, 10, 11)$ sólo 2 pueden comprimirse pues sólo hay 2 de longitud menor, por tanto hay al menos 4 cadenas aleatorias 2 de longitud 1 y 2 de longitud

2. Generalizando este argumento para longitud N vemos que la cantidad de cadenas aleatorias es siempre mayor.

Los hechos anteriores son muy extraños, si las cadenas aleatorias son de las más comunes cómo puede haber tan pocas conocidas en la matemática, en los siguientes capítulos nos dedicaremos a buscar cadenas aleatorias, primero desarrollando una técnica que nos permita establecer su aleatoriedad y analizando secuencias de una de las pocas fuentes que parece tener una buena aleatoriedad: los fenómenos cuánticos.

1.3. Aleatoriedad e incomputabilidad

Todas las ideas anteriores parecen una descripción increíblemente precisa de un concepto de aleatoriedad que no estaba descrito en las matemáticas y que se encontraba olvidado debajo de mucha probabilidad, sin embargo una gran sorpresa nos espera al intentar llevarlo a la práctica aún en los modelos más sencillos que podamos imaginar. Para terminar este capítulo damos una prueba intuitiva e informal del resultado central en la Teoría Algorítmica de la Información

Teorema 1.2 *No existe un algoritmo general que nos permita saber si una cadena es aleatoria.*

Procedemos por contradicción, supongamos que existe tal algoritmo R que dice si una cadena es aleatoria y tomemos M una máquina universal, propongamos una máquina que haga lo siguiente:
$A(n)$:

1. Recibe un número natural n (en cierta base o codificación).

2. Genera las cadenas n (una a una).

3. Usando el algoritmo R busca las cadenas que sean aleatorias(según M).

4. Si son aleatorias las imprime.

Es decir la máquina recibe un número y empieza a ejecutar las cadenas que sean programa y aleatorias con longitud mayor al número que recibió; llamemos a este algoritmo A.

Como M es universal podemos hacer una descripción del algoritmo A en M, que equivale a escribir el código de A en la máquina M, como resultado tendremos una cadena que llamamos a.

Para obtener la contradicción hacemos lo siguiente: $|a|$ es la longitud de el código de A, así que aplicamos $A(|a|+1)$ entonces el programa llegará a una cadena c con longitud $|c| \geq |a|$ lo ejecutará y dará su salida. Pero eso significa que la información de la cadena es a lo más $|a|$, es decir $|c| \leq |a|$, con esto tenemos la contradicción.

Este resultado limita la aplicación de esta definición de información y aleatoriedad pues no hay forma (en general) de calcularla, de hecho como se ve en la prueba obtenemos una contradicción para todas las cadenas aleatorias con longitud mayor que $|a|$.

Este resultado no es el fin de las aplicaciones de esta teoría, por muy difícil que parezca hay aún algunos efectos donde se puede buscar la aleatoriedad que se tocarán en lo siguiente.

Capítulo 2

Normalidad de Borel

La normalidad de Borel como se define en lo siguiente trabaja como un buen método para probar la aleatoriedad de una cadena, debido a que nos permite ver si la cadena tiene peso hacia algún símbolo o subcadena en particular. Sin embargo sus relaciones con la aleatoriedad según la TAI no son tan transparentes. Aquí damos cuenta de estas relaciones y definiciones formales necesarias para entender a la normalidad de Borel como prueba de aleatoriedad.

2.1. Normalidad de Borel

Una manera muy natural de clasificar la aleatoriedad de una cadena se encuentra al contar la cantidad de veces que una letra aparece en esta, por ejemplo:

$$c_1 = 11111111111111111111111111111111$$

no parece una cadena muy aleatoria y aún

$$c_2 = 01101111111111111111111111111111$$

si bien parece más aleatoria que la anterior no parece tan aleatoria como

$$c_3 = 01000101110001001011011011110001$$

El hecho de que en una cadena una letra tenga un mayor número de apariciones que los demás hace parecer a una secuencia menos aleatoria, por tanto un

criterio de aleatoriedad aplicable es: " cada símbolo debe aparecer el mismo número de veces en la cadena".

El criterio anterior parece necesario para definir aleatoriedad pero no suficiente, un clásico ejemplo de esto es la siguiente cadena:

$$c_4 = 01010101010101010101010101010$$

donde cada símbolo aparece igual número de veces, sin embargo es la repetición de 01 lo cual no parece aleatorio. Supongamos ahora que un lector rápido percibiera los símbolos por pares. Luego la longitud de la cadena para él sería la mitad de la original; al leer un símbolo podría leer cuatro posibilidades $00, 01, 10, 11$ y desde su punto de vista el alfabeto está compuesto por cuatro elementos y no dos, así que el criterio que pedimos antes ya no se satisface para el último ejemplo pues 01 predomina por encima de los demás. Por tanto para este lector rápido el criterio de aleatoriedad se escribiría de la siguiente forma:

"Cada pareja de 2 símbolos debe aparecer el mismo número de veces en la cadena"

Si tuviéramos lectores más y más rápidos tendríamos que considerar que todas las cadenas de longitud 3 aparecieran el mismo número de veces, al igual con 4,5 etc. Para formalizar estas ideas proponemos las siguientes definiciones:

Definición 2.1 *Un alfabeto es una colección finita de símbolos.*

Definición 2.2 *Sea A un alfabeto, una cadena es una sucesión (finita o infinita) de símbolos. Sean c, d cadenas definimos su concatenación cd como la sucesión de símbolos de c seguida de la de d. Sea c una cadena, decimos que una cadena b es una subcadena de c si existen cadenas a y a' tales que. $c = aba'$*

Definición 2.3 *Sea A un alfabeto definimos $A*$ como la colección de todas las cadenas finitas de A*

Definición 2.4 *Sea a cadena en A alfabeto y n natural no nulo, definimos $||_n : A^* \times \mathbb{N}^+ \to \mathbb{N}$ como $|c|_n = [\frac{|c|}{n}]$*

Luego $|c|_n$ es la longitud según un lector rápido que percibe de n en n símbolos y además $|| = ||_1$.

Definición 2.5 *Sean a,c cadenas sobre un alfabeto A que cumplen*$|a| < |c|$*, definimos* $N : A^* \times A^* \to \mathbb{N}$ *como* $N(c, a)$ *igual al número de veces que la subcadena a aparece en la cadena c.*

Entonces si c_i corresponde a nuestros ejemplos anteriores tenemos que $N(c_1, 0) = 0, N(c_2, 0) = 2, N(c_3, 0) = 16 y N(c_4, 10) = 15$. [1]

Usando las definiciones anteriores el criterio que tomamos tiene una forma más regular

$$N(c, 0) = N(c, 1) = \frac{|c|}{2}$$

Para el caso del lector rápido cada par es un sólo símbolo entonces

$$N(c, 00) = N(c, 01) = N(c, 10) = N(c, 11) = \frac{|c|_2}{4}$$

.
.
.

De las ecuaciones anteriores podemos extraer la siguiente ecuación general para todos los lectores rápidos.

$$\frac{N(c, a)}{|c|_{|a|}} = \frac{1}{2^{|a|}} \tag{2.1}$$

Este es un enunciado matemático del criterio que especificamos al principio. Sin embargo no es muy útil debido a que las cadenas no necesariamente serán múltiplos de la longitud de la subcadena, luego la división de lado izquierdo no será exacta y en pocos casos tendremos la igualdad deseada. Para salvar este problema relajaremos la condición de igualdad y sólo pediremos que las cantidades sean muy parecidas de la siguiente manera:

Definición 2.6 [2] *Una cadena c sobre una alfabeto A es* ϵ*,m-limiting si para todas las cadenas a tales que* $|d| = m$ *entonces*

$$\left| \frac{N(c, a)}{|c|_{|a|}} - \frac{1}{2^{|a|}} \right| < \epsilon \tag{2.2}$$

[1] la definición anterior se encuentra en los textos como $N_i^m(c)$ donde m es la longitud de la subcadena, i es el número de la subcadena en el orden lexicográfico y c es la cadena a analizar. Cambiamos de notación por claridad

[2] Estas definiciones se deben a Calude [Chr94]

Definición 2.7 *Una cadena c es ϵ,m normal si es ϵ,n-limiting para n desde 0 hasta m.*

Hasta ahora logramos una buena definición del criterio con el que empezamos, pero aún nos encontramos frente a dos grandes problemas pues no tenemos definido que tan pequeña será ϵ y esto será crucial para discriminar algunas de las cadenas. El otro problema es relacionado con la m, es claro que m tiene que ser más pequeña que la cadena para que el conteo de apariciones pueda llevarse a cabo y aún más, esta debe ser mucho más pequeña que la cadena para que el análisis tenga sentido, por ejemplo si esperamos que todas las cadenas de longitud m aparezcan igual número de veces debemos permitir que cada una aparezca al menos una vez, como hay 2^m cadenas de longitud m significa que si n es la longitud de la cadena que vamos a analizar debemos tener:

$$n > 2^m m \Rightarrow \log n > m + \log m. \tag{2.3}$$

Este es un argumento sencillo que debe mejorarse pues si las cadenas aparecen sólo 1 vez no tendremos una estadística muy buena.

2.2. La relación con la aleatoriedad

La aleatoriedad en el sentido de la TAI tiene una relación muy estrecha con la normalidad como la definimos anteriormente. Esto fue encontrado por Calude en [Chr94] al principio de la década de 1990. Esta relación es natural pensando en la aleatoriedad como incompresibilidad, pues si uno de los símbolos apareciera mucho más que los demás sería fácil comprimirlo poniendo el número de veces que aparece sucesivamente o usando cualquier otro método.

Esta relación con la aleatoriedad se reflejará en los parámetros m y ϵ de la sección pasada poniendo los siguientes valores

$$m = \log\log|c| \tag{2.4}$$

$$\epsilon = \sqrt{\frac{\log|x|}{|x|}} \tag{2.5}$$

Definición 2.8 *Una cadena que es ϵ,m normal con los valores anteriores se llama Borel normal.* [3]

La relación exacta entre aleatoriedad y normalidad está dada por el teorema siguiente y es debido a esto que los parámetros se eligieron de esa manera como se menciona en [Chr94]

Teorema 2.1 *Para todo natural $t \geq 0$ podemos calcular un número natural M(t) tal que cualquier cadena t-aleatoria de longitud mayor a M(t) es Borel normal.*

El teorema anterior nos da un lugar donde buscar la aleatoriedad, ya que a partir de un valor todas las cadenas aleatorias serán Borel normales y tenemos un algoritmo para revisar esto en cada cadena a diferencia de la aleatoriedad para la cual estamos restringidos por el teorema 1.2.

Aunque en su mayoría son difíciles existen ejemplos de cadenas normales de Borel que tienen patrones muy simples.

Realmente la definición introducida aquí no es la definición original de normal de Borel, esta definición se hizo para números reales refiriendo a su desarrollo decimal

Definición 2.9 *Un número real x es normal de Borel si su desarrollo en cualquier base cumple con que el número de apariciones de cualquier dígito es igual para todos los dígitos cuando el número de cifras en el desarrollo tiende a infinito.*

sin embargo la modificación que se hizo es para tratar con cadenas finitas y no intenta rescatar en su totalidad el espíritu de la definición de Borel que originalmente estaba más ligada a los números reales, en lugar de esto se intenta acercar a la definición de la TAI.

[3] De hecho estos parámetros pueden seleccionarse de manera que el teorema se siga cumpliendo y permita a más cadenas entrar en la definición, esto es debido a que $\log\log|x|$ durante la prueba se toma para ser asintóticamente mayor que log sólo para lo siguiente se cumpla

$$\frac{(\log\log M)^3}{\log M} \leq \frac{1}{\sqrt{\log M}}$$

de manera que podemos relajar la condición ligeramente.

Capítulo 3

Generación Electrónica de Números Aleatorios

En este capítulo intentamos hacer contacto entre el formalismo matemático de los capítulos anteriores y la generación real de los números aleatorios tratando un caso en particular.

3.1. Distintas formas para generar números aleatorios

Como ya se mencionó anteriormente la necesidad de generar números aleatorios ha crecido con la evolución tecnológica y para solventarla se han buscado distintas opciones:

- Generadores de números pseudo-aleatorios.

  Existen varios algoritmos para producir números aleatorios, simplemente se toma un algoritmo cuya salida parezca suficientemente aleatoria. Varios problemas comunes surgen de este tipo de generadores como correlaciones. El azar que necesitan los dispositivos como PC's y celulares, viene de este tipo de generadores. Si bien existen muchos métodos de generación de números pseudo-aleatorios, la mayoría tiene problemas de período, es decir, después de generar una cierta cantidad de números al azar repite la secuencia, este defecto es solventado contemplando la cantidad de números que es necesaria para una cierta aplicación y

diseñando el algoritmo de manera tal que este período de repetición sea mucho más grande.

- Generadores con semilla.

  Este tipo de generación es muy similar al anterior, tiene la ventaja de usar un número, que se denomina semilla, a partir del cual se hace la generación. Si se consigue una semilla bastante aleatoria este tipo de generación es muy útil pues a partir de pocos números aleatorios usados como semilla podemos generar, dentro de ciertos límites, una cantidad mayor. Algunas semillas muy usadas son el tiempo, el teclado etc.

- Generación de números aleatorios por hardware.

  A diferencia de las opciones anteriores, en este caso se usa un dispositivo físico con una propiedad que parezca no tener estructura, generalmente ruido, al medir dicha propiedad podemos usar los resultados como números al azar. Este tipo de generación es bastante efectiva, siempre que se elija un buen dispositivo físico, algunos ejemplos son el ruido atmosférico que es usado con gran éxito por el servidor random.org (http://www.random.org/), el ruido térmico, el ruido de avalancha etc. generalmente se usan dispositivos de naturaleza eléctrica o electrónica debido a que la tecnología actual es de este tipo, hay otros dispositivos relacionados con fluidos por ejemplo pero la interfaz entre estos y la computadora es más difícil de realizar.

- Generadores cuánticos de números aleatorios.

  Este tipo de generación está contemplada en el inciso anterior es decir que el dispositivo físico sea de naturaleza cuántica, esto tiene la ventaja de que la teoría lo respalda como no determinista. En el siguiente capítulo se discute más ampliamente este tipo de generación.

3.2. Dispositivo físico

Para ilustrar las ideas anteriores tratamos el caso de un generador electrónico de números al azar, este generador se basa en el ruido de una unión p-n inversamente polarizada, esto puede ser un diodo inversamente polarizado o un transistor. El diodo es un dispositivo electrónico que sólo permite el paso de corriente en una dirección usando una unión p-n, en esta unión hay una

zona llamada la zona de agotamiento que cuando se polariza inversamente, es decir hay una diferencia de potencial negativa del lado p al n, se ensancha produciendo una barrera de potencial a los portadores de carga prohibiendo el paso de corriente.

El procedimiento anterior no es perfecto pues hay electrones con suficiente energía que logran cruzar la barrera de potencial y esto produce una pequeña cantidad de corriente, la señal producida por este fenómeno se suele llamar ruido avalancha. Uno de los procedimientos más comunes para generar números es justo esta señal, debido a su comodidad y su fácil implementación, por ejemplo algunos procesadores Intel ya presentan esta característica[Mec14].

En nuestro caso usaremos un circuito diseñado por Giogio Vazzana obtenido de http://holdenc.altervista.org/avalanche/, en esta página se ofrece un circuito basado en las ideas anteriores, es decir una unión p-n inversamente polarizada, una etapa de amplificación y una etapa de acoplamiento hacia un convertidor analógico-digital. El autor también prueba su circuito con algunas pruebas de aleatoriedad.

En la primera etapa del circuito tenemos un transistor 2N3904 que se polariza inversamente entre la base y el emisor, el autor usa este transistor en particular debido a que presenta más ruido que otros modelos similares, este ruido es conectado a la base de otro transistor del mismo modelo correctamente polarizado en configuración de emisor común logrando una señal de ruido.

La siguiente etapa consiste en un seguidor de voltaje para disminuir la impedancia y un amplificador usando un para de amplificadores operacionales TL084, y después de esta etapa ya se tiene disponible un ruido analógico.

La etapa final consiste en una fuente de corriente y un comparador que permite discretizar la señal de ruido en ceros y unos para obtener ruido digital. El diagrama de esto se puede ver en la figura 3.2.

Finalmente se midió la salida digital del circuito con un PIC18F2550 que manda directamente los datos a la computadora a través del puerto USB, se tomó una cadena de 1,000,000 de bits para analizar.

3.3. Análisis de los datos.

Para buscar aleatoriedad en los datos se usará el criterio de Borel expuesto anteriormente, que está relacionado con el concepto de aleatoriedad del primer capítulo a través del teorema de Calude. recordemos que la idea de

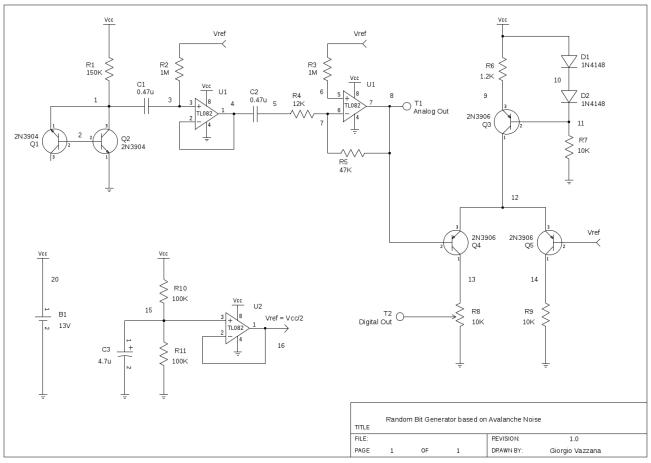

Figura 3.1: Diagrama del generador

esta prueba es separar los bits obtenidos de 2 en 2, 3 en 3 etc. y contar el número de veces que cada una de estas subcadenas aparece, si se encuentra que alguna aparece un número de veces mayor lo interpretamos como una señal de estructura y por tanto la señal no es aleatoria.

En la figura 3.2 vemos graficados los resultados para el análisis de Borel considerando las subcadenas '00','01','10' y '11', este gráfico está normalizado. Las líneas rojas superior e inferior representan el máximo y mínimo valor permitido por la cota de Borel respectivamente. La linea roja central representa el valor promedio esperado, en este caso como tenemos 4 cadenas el valor promedio que esperaríamos sería $\frac{1}{4} = 0{,}25$.

El caso de cadenas de longitud 3 se puede ver en la figura 3.3 aquí también tenemos la gráfica normalizada y las líneas representan lo mismo que en el caso anterior con la diferencia que el valor esperado para este caso es $\frac{1}{8} = 0{,}125$.

Finalmente el otro caso es análogo con un valor promedio de $\frac{1}{16} = 0{,}625$

En las siguientes tablas mostramos númericamente los resultados obtenidos para cada caso.

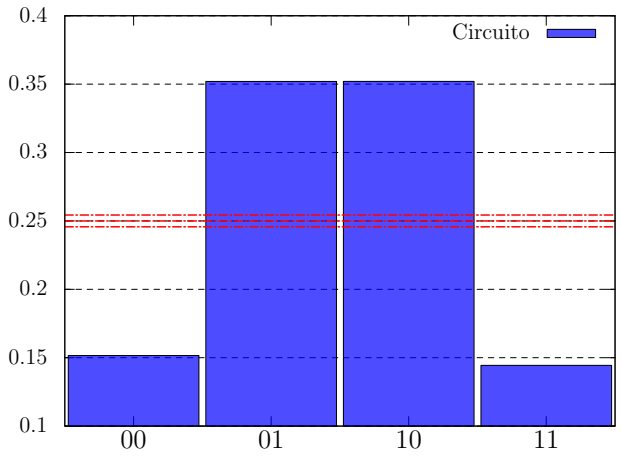

Figura 3.2: Análisis de Borel nivel 2 para el circuito

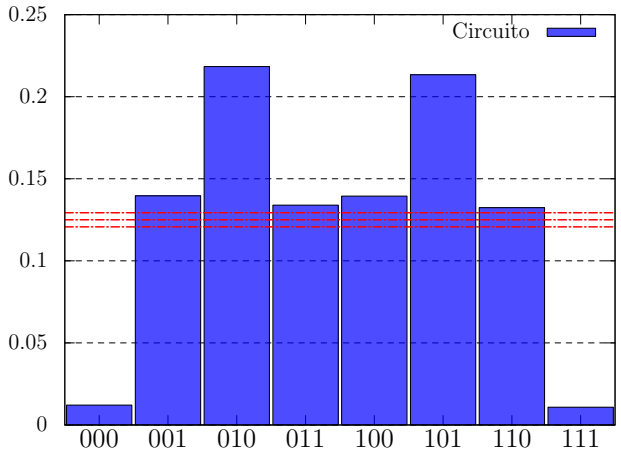

Figura 3.3: Análisis de Borel nivel 3 para el circuito

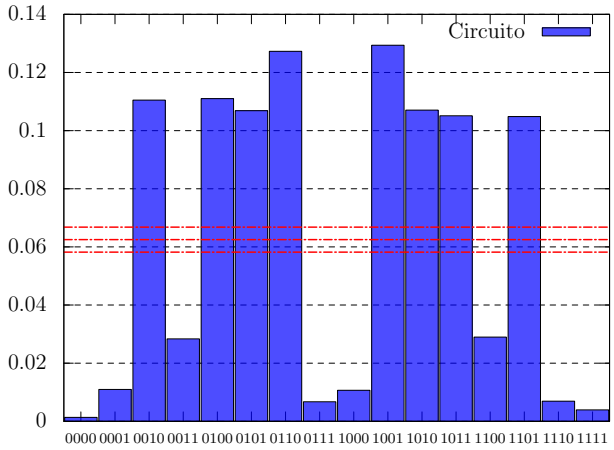

Figura 3.4: Análisis de Borel nivel 4 para el circuito

Resultados Borel	
nivel 1	
subcadena	porcentaje %
0	50.01
1	49.99

Resultados Borel	
nivel 2	
subcadena	porcentaje %
00	19.05
01	35.23
10	35.25
11	10.44

Resultados Borel	
nivel 3	
subcadena	porcentaje %
000	02.03
001	17.01
010	25.25
011	10.01
100	16.97
101	18.34
110	09.96
111	00.39

Como vemos para el nivel 1, es decir cuando sólo se cuenta el número de ceros y unos tenemos aproximadamente el mismo número de ambos, como se ve en la tabla del nivel 1 esto se cumple bien. En el caso del nivel 2 se ve la fuerza de la prueba que estamos usando, las cadenas de 2 símbolos no aparecen de manera igual, las de la forma 10 y 01 aparecen 2 o 3 veces más que las cadenas 00 y 11 y en las cadenas de 3 símbolos tenemos una distribución más desigual.

Recordemos que el criterio de Borel está dado por la desigualdad:

$$\left| \frac{N(x,a)}{|x|_{|a|}} - \frac{1}{2^{|a|}} \right| < \sqrt{\frac{\log|x|}{|x|}}. \tag{3.1}$$

De lado derecho tenemos algo del orden de 4×10^{-3} es decir 0,4 % de diferencia respecto de la media, pero esta es una condición que los datos anteriores violan ampliamente, por ejemplo tomemos en el nivel 2 la subcadena "01"

$$\left| \frac{N(c,01)}{|c|_{|01|}} - \frac{1}{2^{|01|}} \right| = \left| 0{,}3523 - \frac{1}{2^2} \right| = \left| 0{,}1023 \right| < \sqrt{\frac{\log|10^6|}{|10^6|}} = 4 \times 10^{-3} \tag{3.2}$$

El siguiente análisis que realizamos es el de autocorrelación, recordemos que la autocorrelación para una señal discreta de n datos se calcula de la siguiente manera:

$$R(k) = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} (X_t - < X >)(X_{t+k} - < X >) \tag{3.3}$$

Donde $< X >$ es el promedio y n,k son números naturales que cumplen $k << n$ pues generalmente n es un número muy grande. Es decir, multiplicamos el valor de la señal por el valor k lugares adelante y tomamos el promedio a lo largo de toda la serie de valores, por último la división por la varianza nos permite sólo valores entre -1 y 1.

La intención de esta prueba es encontrar patrones "periodicos", una cadena que tuviera este tipo de patrones, o los tuviera al menos de forma aproximada, sería más compresible pues en vez de especificar la cadena completa sólo se tendría que especificar uno de sus periodos y la cantidad de periodos a repetir.

Para obtener un poco de intuición sobre esta prueba, analicemos algunos casos sencillos. Para los casos en los que las cadenas están sólo compuestas de ceros o unos la autocorrelación puede tomarse como 1(aunque estrictamente la varianza ahí es cero), otro caso simple pero más interesante que tiene el mismo número de ceros y unos es el de la cadena con la forma

$$01010101010101\cdots$$

en este caso tenemos que la autocorrelación es 1 si el desplazamiento es par y -1 si es impar. Finalmente un caso un poco más complicado es el siguiente

$$0100100100100100100\cdots$$

aquí la autocorrelación vale 1 si el desplazamiento es divisible entre 3, y -1/3 en otro caso.

En la figura 3.3 podemos ver la autocorrelación de la muestra obtenida del circuito calculada para distintos valores, ahí vemos como para valores pequeños del desplazamiento la autocorrelación alcanza valores grandes; por ejemplo para un desplazamiento de 1 tenemos un valor de -0.4 este valor tan grande es consistente con el hecho de que en el análisis de las cadenas encontramos la subcadenas "01" y "10" un número de veces mayor, pues estas dos cadenas originan autocorrelaciones negativas en desplazamientos impares.

Estos dos análisis revelan que la cadena que se obtuvo usando este circuito tiene estructura o patrones, esto es justamente lo que busca al intentar demostrar que una cadena no es aleatoria. Hay que recordar del capítulo uno que no es posible demostrar en general la aleatoriedad de una cadena, pero al encontrar estas regularidades podemos asegurar que la cadena tiende a ser compresible, justo lo contrario de nuestra definición de aleatoriedad.

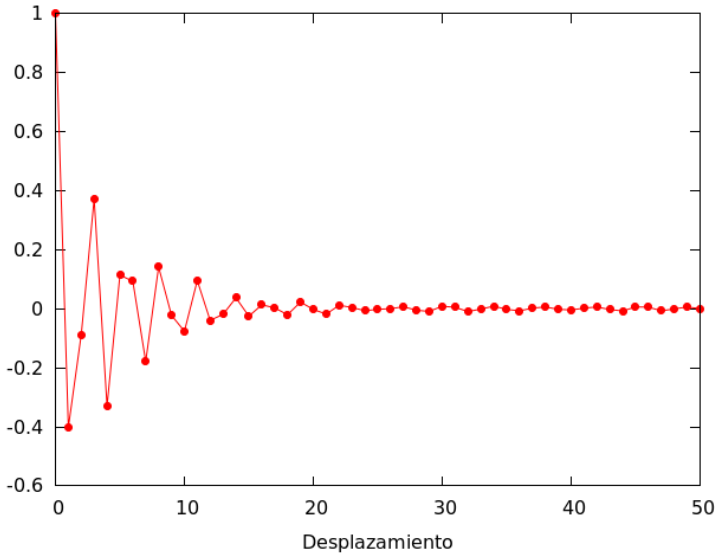

Figura 3.5: Autocorrelación

Para finalizar esta sección hay un punto importante que mencionar respecto a este tipo de aleatoriedad, se mencionó anteriormente el ruido de avalancha es usando en distintas aplicaciones que necesitan la aleatoriedad, entonces ¿cómo es posible que encontremos tantas regularidades? en realidad aquí no realizamos ningún procesamiento sobre la señal obtenida, salvo la discretización, pero en las aplicaciones reales es normal emplear varios tipos de procesamiento que generalmente se hacen buscando pasar las pruebas de aleatoreidad más comunes. Esto "mejora" la aleatoriedad de la cadena, un ejemplo de esto se encuentra cuando no tenemos el mismo número de ceros y unos, un procesamiento común es cambiar al azar ceros por unos o viceversa para igualar el número de ambos símbolos. Debido a que nuestra finalidad es analizar la aleatoriedad del fenómeno en sí no aplicamos este tipo de procedimientos a la señal lo que nos llevó a encontrar fácilmente estructura en ella.

Capítulo 4

Aleatoriedad en Mecánica Cuántica

4.1. El postulado de Born

El año de 1926 fue crucial para la mecánica cuántica y para la introducción del azar en la física, en Julio de ese año Schroedinger, que recientemente había propuesto su ecuación de onda bien recibida por mucha gente en contraposición a la mecánica matricial de Heisenberg, publica una nota [Erw26] haciendo notar su posición respecto a la oposición onda-partícula, prefiriendo el concepto de onda y mostrando para el caso del oscilador armónico que un paquete de ondas no se dispersa.[Abr82]

Por otro lado Max Born, en el artículo de Junio de ese mismo año [Bor26], sugiere por primera vez la idea de relacionar la probabilidad con el cuadrado de la función de onda [Abr82]. Sin embargo la diferencia entre las probabilidades clásicas, que se refieren a ignorancia del estado del sistema, y las nuevas no es clara. Born, en ese mismo artículo, toca el punto central de como el determinismo se pierde bajo este esquema "Desde el punto de vista de la mecánica cuántica no existe cantidad que en un caso individual determine el efecto de una colisión...Yo mismo tiendo a dejar el determinismo en el mundo atómico" [Bor26][Abr82]. Este último enunciado es de gran importancia en nuestra discusión, después de todo si no hay una forma de determinar el resultado de una interacción individual, ¿Significará esto que el conjunto de resultados no tiene ningún patrón?

Dentro de las formulaciones modernas de la mecánica cuántica el postu-

lado de Born es indispensable. Generalmente dice, aunque con lenguajes muy distintos, que la probabilidad de obtener un valor en una medición de una observable A es el módulo al cuadrado de su coeficiente, es decir si

$$|\Psi\rangle = \sum a_i \, |\psi_i\rangle$$

donde $\{|\psi_i\rangle\}_i$ es una base ortonormal dada por los vectores propios del operador A, la probabilidad de obtener el estado ψ_i esta dada por $|a_i|^2$.

Muchas exposiciones no hablan más de cuál será el valor obtenido en una medición individual, algunas suelen simplemente agregar que el valor que se obtiene en una medición es un eigenvalor (de la matriz asociada a la observable) al azar, o que no se puede saber de antemano que valor se va a obtener.

Sumando lo anterior podemos ver que este es otro concepto de azar distinto del que hemos manejado antes, pero no están directamente ligados.

Propongamos un sistema cuántico y hacemos una medición en él obteniendo un resultado, si lo repetimos para muchos sistemas preparados de la misma manera, al final tendremos una lista de resultados, ¿Estos resultados tendrán estructura?

4.2. Azar obtenido de la física

Una de las opciones que se ha explorado más en los últimos tiempos es tomar fenómenos de la naturaleza que sean muy "complejos" y medir alguna propiedad aprovechándola para obtener números aleatorios.

Debido a la naturaleza de la mecánica cuántica podemos usar el azar que aparece, según el postulado de Born, para generar números aleatorios con la ventaja que hay un postulado de la teoría que respalda su aleatoriedad.

El sistema más simple que podemos pensar para esto es uno que tenga únicamente dos estados. Sean A y B estados de un sistema, representados por $|a\rangle$ y $|b\rangle$ respectivamente, que corresponden a los valores a y b de una observable. Si preparamos un estado de la forma:

$$|s\rangle = \frac{|a\rangle + e^{i\alpha} \, |b\rangle}{\sqrt{2}}, \tag{4.1}$$

según la regla de Born es igual de probable obtener en una medición, de la observable en cuestión en la base dada por $|a\rangle$ y $|b\rangle$, el valor a o el

valor b. Usando el hecho anterior podemos obtener un bit aleatorio de este procedimiento simplemente tomando los resultados de varias mediciones y asociándolos a 0 o 1, por ejemplo:

ababbbaaababaabbab...

010111000101001101...

Estos números tienen propiedades deseables, debido a la forma en que los obtenemos.

Lo anterior lleva una cuestión importante de fondo, el concepto de aleatoriedad que se propone en la mecánica cuántica es de naturaleza estadística, ¿hasta que punto el azar cuántico coincide con el azar algorítmico? Esta pregunta es tratada en uno de los artículos centrales para este trabajo[Chr09].

Comparar estos conceptos no es fácil, de hecho parece prácticamente imposible debido a que como se mostró en el primer capítulo es imposible obtener por métodos algorítmicos la cantidad de información de una cadena, de manera que se limita bastante el análisis. Lo anterior es realmente una barrera muy fuerte que si bien no puede saltarse[1] nos podemos acercar mucho a ella.

4.3. Esfuerzos hacia la incomputabilidad

Como se señaló anteriormente el problema de la detención es incomputable de una máquina de Turing, luego no hay algoritmo que permita saber si una máquina se parará o no, así que si queremos obtener esa respuesta en general tendremos que correr la máquina y esperar a que pare. Si para, sabremos la respuesta, pero si no tendremos que seguir esperando.

De la misma manera, para saber si una cadena es aleatoria, podemos hacer este tipo de esfuerzos parciales. Si tenemos una cadena aleatoria de n bits hay un procedimiento que bien nos puede llevar a saber si es aleatoria, en la máquina que se use para determinar la información primero se corren todos los programas de 1 bit, luego todos los de 2 bits, luego los de 3, y continuamos hasta los programas de 29 bits; si alguno de los programas anteriores tuvo como salida la cadena que estamos analizando entonces no es aleatoria, y en caso contrario tenemos una cadena aleatoria.

[1] No puede saltarse en el sentido algorítmico los métodos de la hipercomputación , si los hubiere podrían violar esta barrera

El procedimiento anterior parece un método para poder saber si una cadena es aleatoria aún sobre el teorema 1.2, incluso no parece difícil implementarlo realmente en un lenguaje de programación específico pero debemos mirarlo más de cerca. Para poder revisar si la salida de cada uno de los programas es o no la cadena que analizamos debemos esperar a que el programa termine, ¿Cómo vamos a saber si el programa va a terminar? Es exactamente aquí donde el procedimiento falla, como se señaló antes este es el problema de la detención, que es incomputable, si esperamos a la salida de cada programa puede que nunca termine y que no obtengamos la respuesta que esperamos.

Debido a todo lo anterior, pareciera que los esfuerzos por estudiar la relación entre la aleatoriedad cuántica y la aleatoriedad algorítmica son en vano, pero si bien no podemos obtener la información de las cadenas puede ser que de alguna manera sepamos si algunas son más aleatorias que otras, es aquí donde se vuelve importante el concepto de normalidad de Borel estudiado anteriormente. El teorema 2.1 relaciona el concepto de normalidad de Borel y el de aleatoriedad algorítmica con una gran ventaja: Saber si una cadena es normal de Borel sí es computable, sí hay un algoritmo que me permite saber si una cadena es normal de Borel o no.[2]

Las cadenas que cumplan con normalidad no necesariamente serán aleatorias, pero todas aquellas que no sean normales podremos descartarlas como cadenas aleatorias. De la misma manera que la normalidad, hay otro conjunto de "pruebas" que podemos hacer a las cadenas que funcionan de la misma manera, no son condiciones suficientes pero siempre necesarias para el azar.

[2]Calude es un personaje central en esta área, acercarse a los objetos incomputables es algo por lo que el es conocido, en [CDS02] Calude calcula los primeros 64 bits de un número que no es algorítmico : una Ω de Chaitin

Capítulo 5

Experimento

En este capítulo volvemos a analizar una fuente de números aleatorios, en este caso una fuente cuántica usando fotones individuales, con el objetivo de comparar en la medida de lo posible la aleatoriedad de las cadenas obtenidas con la aleatoriedad algortítmica.

5.1. Dispositivo físico

El sistema de cuántico de 2 estados mencionado en anteriormente puede realizarse usando fotones y un divisor de haz, se disparan fotones hacia el divisor y se mide en las salidas del divisor si se encuentra o no un fotón, repetir este procedimiento nos genera 2 cadenas de resultados uno para cada detector usando estas podemos generar un cadena binaria que señale en que detector se llevo cada detección.

Para fijar las ideas anteriores nombremos a las entradas del divisor de haz como vertical y horizontal, debido a que disparamos sólo un fotón en sus entradas, escribimos: $|1\rangle_h |0\rangle_v$ si el fotón está en al dirección horizontal y $|0\rangle_h |1\rangle_v$ si el fotón está en la dirección vertical.

Si tomamos esos dos estados como base, el operador que nos da la evolución del estado al pasar por un divisor de haz 50:50 es:

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{5.1}$$

de manera que el estado que obtenemos al disparar un fotón hacia sus entra-

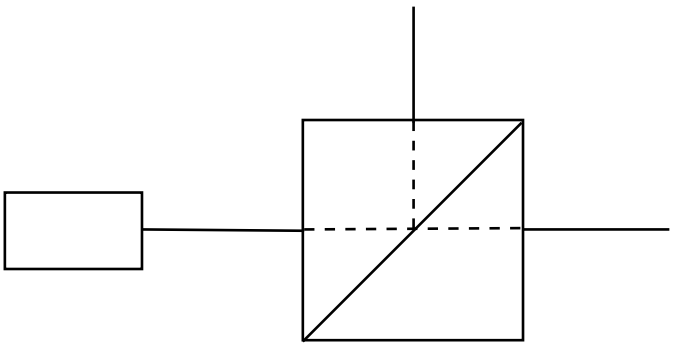

Figura 5.1: Esquema experimental

das es:

$$|1\rangle_h |0\rangle_v \rightarrow \frac{|1\rangle_h |0\rangle_v + |0\rangle_h |1\rangle_v}{\sqrt{2}} \tag{5.2}$$

este estado cumple el requisito de ser observado la mitad de las veces en el estado $|1\rangle_h |0\rangle_v$ y la mitad en el estado $|0\rangle_h |1\rangle_v$.

Otro acercamiento es el de usar la polarización de los fotones para producir un estado parecido al anterior:

$$\frac{|+\rangle + e^{i\alpha} |-\rangle}{\sqrt{2}} \tag{5.3}$$

donde $|+\rangle$ y $|-\rangle$ equivale a un estado con polarización vertical y horizontal respectivamente.

Aquí un esquema parecido a 5.1 se mantiene, simplemente pensando en el divisor de haz como uno polarizante, si en una de las entradas del divisor de haz se disparan fotones con el estado 5.3, el divisor de haz permitirá el paso a cierta polarización y reflejará el otro tipo, si ponemos detectores en las salidas de el divisor de haz tendremos el arreglo deseado. Esta propuesta está desarrollada en [Chr09].

Para este trabajo se utilizó un acercamiento diferente, en vez de trabajar con los estados de los fotones, se usaron los tiempos de llegada entre cada uno, es decir, disparamos fotones individuales hacia un detector, esperamos la llegada de un fotón y medimos el tiempo que tarda en llegar el siguiente, para ello se usa un diseño experimental como el de la figura 5.2

Naturalmente los detectores no son perfectos, ni el experimento aislado completamente de los alrededores, así que hay detecciones falsas tanto por

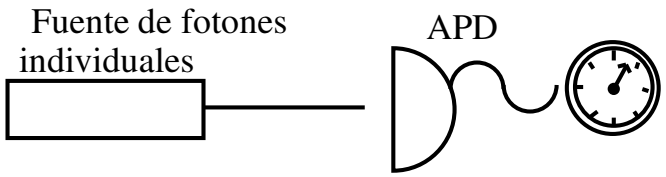

Figura 5.2: Esquema experimental

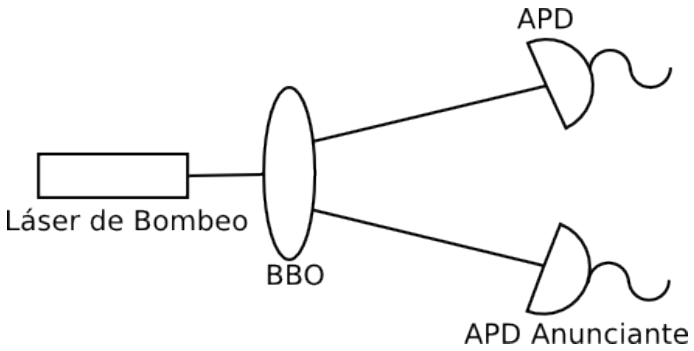

Figura 5.3: Esquema experimental

fotones provenientes del entorno como de cuentas obscuras debidas al ruido intrínseco del detector APD(fotodiodo de avalancha). Para asegurar que las detecciones obtenidas sean completamente "cuánticas" usamos como fuente de fotones individuales parejas de fotones producidas por un cristal BBO utilizando uno como el anunciante o testigo y otro para realizar la medición de tiempo. Así que pasamos al esquema de la figura 5.3.

La salida de los detectores es del tipo ttl y fue medida con un osciloscopio digital, cada detector ocupando un canal y muestreando cada 4ns; esto nos arrojó 2 señales de voltaje que fueron convertidas a cadenas de bits, donde '1' significa detección y '0' no detección, usando una discriminación en el voltaje (las detecciones consecutivas que provienen de considerar la misma deteccion 2 veces son descartadas). Las cadenas anteriores se analizaron buscando coincidencias temporales en las cadenas, esto es equivalente a realizar una operación '&' entre los bits, en caso de tener una coincidencia se interpreta una pareja que proviene del BBO.

Los detectores usados aquí tienen un tiempo muerto de 20ns de manera que el flujo de fotones debe mantenerse a menos de 50MHz, [1] de las medi-

[1] Usamos MHz para referirnos a millones de detecciones por segundo, también se suele

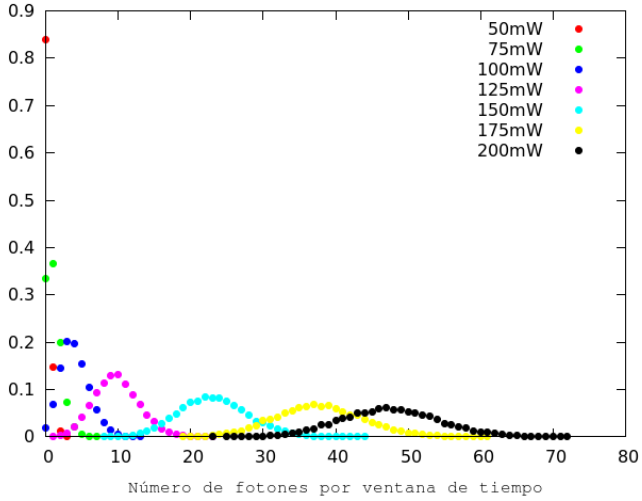

Figura 5.4: Distribución número de fotones

ciones tomadas ningún caso supero 10 MHz de manera que estamos en un régimen donde el tiempo muerto es de magnitud menor.

En primer lugar sacamos el distribución de número de fotones, para ello dividimos nuestra muestra total en intervalos de tiempo iguales y contamos la cantidad de fotones que hay en cada intervalo, después se graficó la cantidad de fotones contra el número de veces que apareció un intervalo con ese número de fotones y esta gráfica fue normalizada, es decir tenemos la distribución de probabilidad de la cantidad de fotones por intervalo de tiempo, el resultado para 100 μs se ve en la figura 5.4.

La distribución de este fenómeno es de tipo poissoniano [SZ97].

5.2. La distribución de tiempos

Para calcular los tiempos simplemente se tomaron las diferencias entre los tiempos de llegada de las detecciones y debido a que las coincidencias son obtenidas en forma de cadenas esto es lo mismo que contar el número de ceros entre unos.

Para caracterizar estas diferencias de tiempo se contó el número de apariciones de cada diferencia de tiempo para después normalizarse de manera que

usar las unidades Mcps.

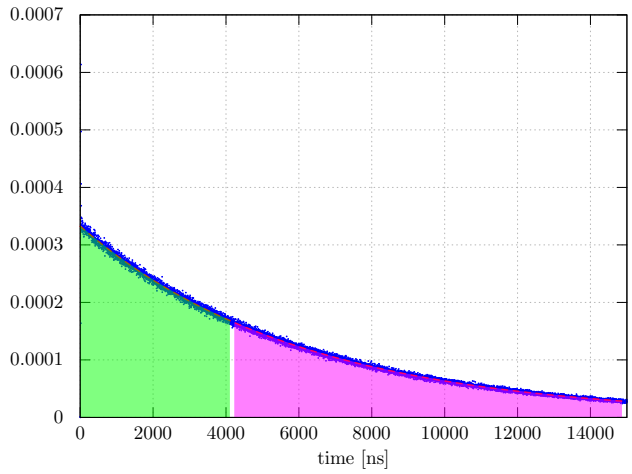

Figura 5.5: Distribución de tiempos entre fotones

obtenemos la distribución de probabilidad de la diferencia de tiempo entre fotones (fig.5.5).

La distribución de los tiempos de llegada sigue una ley exponencial, si se asume una probabilidad instantánea constante de que aparezca una pareja, es posible llegar fácilmente a la distribución exponencial. Si llamamos α a esta probabilidad tenemos que en un intervalo t la probabilidad de que no tengamos una pareja es $1 - t\alpha$, si vemos la primera mitad del intervalo y la segunda como eventos independientes tenemos que la probabilidad de que no haya una pareja en ambas mitades es:

$$(1 - \frac{t\alpha}{2})(1 - \frac{t\alpha}{2}) = (1 - \frac{t\alpha}{2})^2$$

y en general si dividimos el intervalo en n intervalos independientes obtenemos que la probabilidad es:

$$(1 - \frac{t\alpha}{n})^n$$

esta expresión en el límite de n a infinito corresponde a la función exponencial. Es de notar que en esta deducción se usaron 2 hipótesis, que α es constante y por otro lado que los eventos en cada subintervalo de tiempo son independientes, esto revela que el proceso de generación de parejas es independiente para cada evento.

En la figura 5.5 se muestra además de los datos de las diferencias de

tiempo la curva exponencial correspondiente al flujo promedio en los experimentos.

Es importante notar que el hecho de que la distribución de tiempo siga bien una distribución exponencial implica que los eventos son independientes unos de otros, esto es fundamental para garantizar, al menos en un primer acercamiento, que el sistema usado no tiene memoria.

Una vez establecida la distribución exponencial hay que tomar un criterio para obtener una distribución de números plana, aquí proponemos utilizar cada pareja para obtener un bit poniendo una discriminación en los tiempos, los que sean mayores al tiempo de referencia se tomarán como un estado 1 y aquellos con un tiempo menor como un estado 0. El tiempo de referencia debe cumplir con las siguientes propiedades:

$$\int_0^{\tau} \alpha e^{-\alpha t} dt = \int_{\tau}^{\infty} \alpha e^{-\alpha t} dt = \frac{1}{2}$$

de lo anterior resulta que τ tiene la misma forma que el tiempo de semidesintegración

$$\tau = \frac{ln2}{\alpha}$$

recordemos de lo anterior que α está dada por la probabilidad instantánea y para calcularla sólo necesitamos saber el flujo de parejas por unidad de tiempo entonces podemos obtener el tiempo de discriminación tan sólo haciendo un promedio del número de parejas que obtuvimos en la medición total y el tiempo que nos llevó hacerlas.

Este procedimiento se ilustra gráficamente en la figura 5.5, las zonas verde y morada corresponden a los dos intervalos en los que es dividido el espacio de posibles diferencias de tiempo, como se dijo estas zonas tienen igual área para garantizar la igualdad de ceros y unos.

Resumiendo, la asignación de bits para cada fotón se hace revisando su tiempo de llegada respecto al anterior y asociando '1' si el tiempo resulta encontrarse en la zona morada (figura 5.5) o '0' si se encuentra en la zona verde.

Para este estudio se obtuvieron datos suficientes para generar 10 cadenas de 1,000,000 de bits de datos cada una.

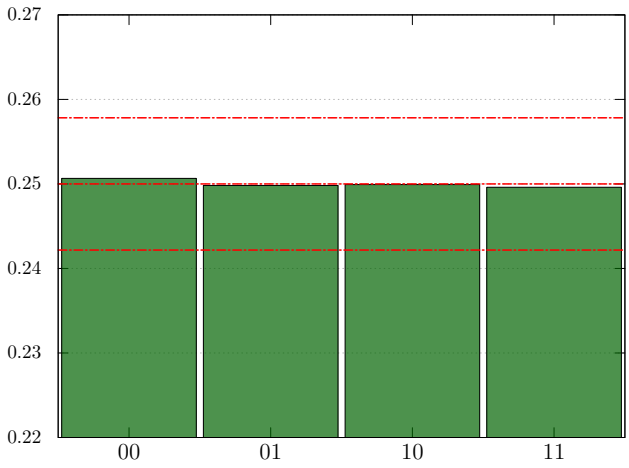

Figura 5.6: Análisis de Borel nivel 2

5.3. Analizando las cadenas obtenidas

5.3.1. Normalidad de Borel

Las cadenas obtenidas, debido al criterio de discriminación, están garantizadas a tener un número de ceros y unos muy cercano, sin embargo no tenemos garantía (salvo tal vez que estos eventos son independientes) de que estén libres de otros patrones, en términos de la normalidad de Borel tenemos garantizado que las cadenas binarias que se obtienen serán normal de Borel grado 1 pero debemos garantizar los otros grados.

Para el análisis de las cadenas se usó un simple programa de conteo, que registra el número de apariciones de una cierta subcadena, recordemos que para revisar Borel nivel 2 debemos contar el número de apariciones de las cadenas de dos símbolos '00','01','10' y '11'. En la figura 5.6 tenemos el histograma de apariciones para una de las cadenas generadas (las otras tienen características similares), la gráfica está normalizada. Las líneas superior e inferior en rojo representan los máximos valores de desviación permitidos por la cota de Borel. La línea roja central corresponde al valor promedio que se espera, es decir en este caso como tenemos 4 secuencias esperamos que cada una de ellas aparezca $\frac{1}{4} = 0{,}25$ de el número total de apariciones. En el gráfico podemos ver que las cadenas pasan la prueba con bastante holgura, y se puede ver un pequeño favorecimiento a la cadena '00'.

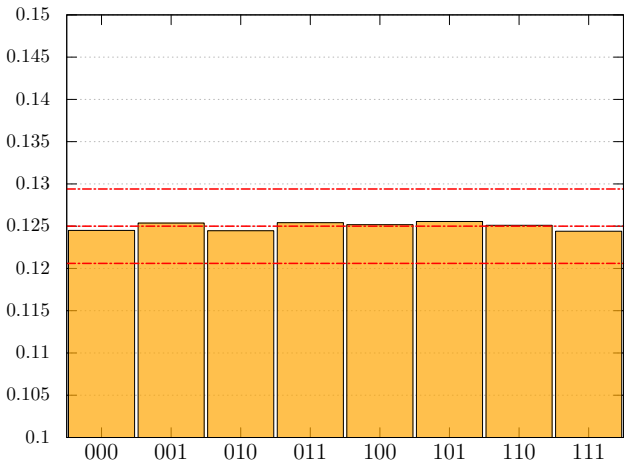

Figura 5.7: Análisis de Borel nivel 3

El análisis para Borel nivel 3 se puede ver en la figura 5.7 donde de nuevo observamos que la cadena pasa las pruebas cómodamente, en este caso hay ocho posibles cadenas por lo que tendremos un valor promedio de $\frac{1}{8} = 0{,}125$.

Finalmente tenemos el análisis para el nivel 4, recordemos que no podemos ir más allá pues el teorema de Calude impone una cota dependiendo del nivel, como en este caso trabajamos secuencias de un millón de datos la cota es :

$$loglog10^6 = 4{,}3$$

en este caso también encontramos que los datos pasan las pruebas fácilmente.

En el fondo sólo nos interesa que todas las cadenas aparezcan el mismo número de veces, así que no nos interesan los valores individuales de las cadenas sino que tanto se desvían de la media, nótese que la cota para las desviaciones es la misma sin importar el nivel así que podemos reportar todas estas juntas. En la figura 5.9 tenemos representadas todas las desviaciones respecto del promedio para cada cadena, de arriba hacia abajo cada marca representa el valor máximo, tres cuartas partes, la mediana, la cuarta parte y el valor mínimo de los datos, la línea roja representa la mayor desviación permitida por la cota de Borel.

En la figura 5.10 tenemos graficados los máximos y mínimos valores de desviación reportados en [CDDS10], donde la línea roja representa la máxima desviación permitida por la cota de Borel. Podemos ver claramente que sus datos tienen problemas para la fuente cuántica en su artículo referida

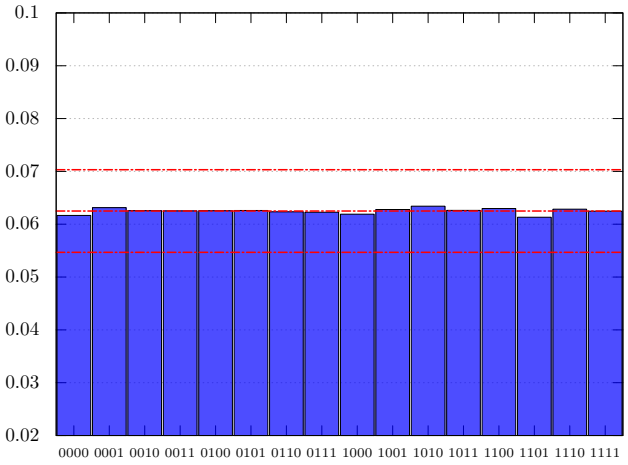

Figura 5.8: Análisis de Borel nivel 4

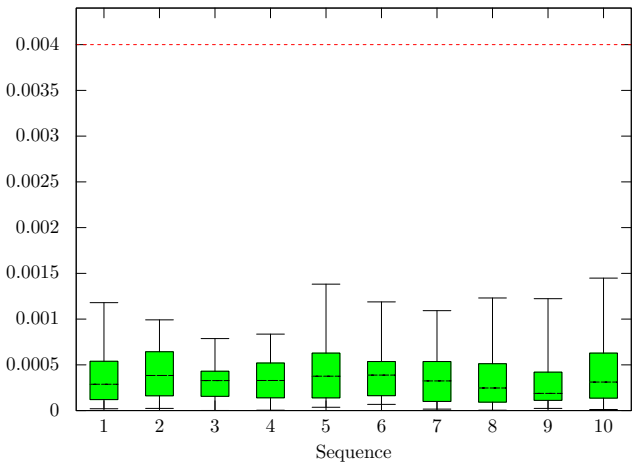

Figura 5.9: Desviaciones respecto del promedio

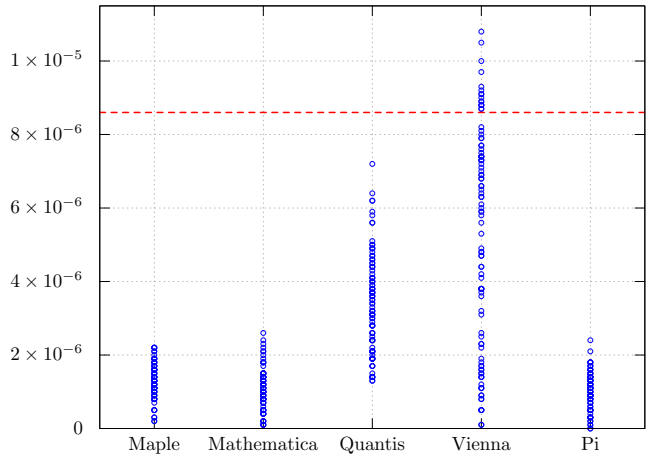

Figura 5.10: Análisis de Borel nivel 2

como "Viena" y ellos reportan esta pequeña desviación que nuestros datos no presentan.

Como vemos las desviaciones de nuestros datos están dentro de los límites de la prueba concluimos que estas cadenas cumplen con la normalidad de Borel.

5.3.2. Autocorrelación

Al igual que en el ejemplo del circuito electrónico también usaremos la autocorrelación para buscar patrones periódicos en las cadenas obtenidas. En la figura 5.11 vemos las correlaciones de los datos obtenidos a distintas potencias. Cada símbolo en la gráfica tiene asociado una de las 10 secuencias.

Como se ve en el gráfico, las autocorrelaciones para todas las cadenas son muy pequeñas, en la figura 5.12 tenemos un acercamiento a los valores de desplazamiento mayores a cero(recordemos que en desplazamiento 0 la autocorrelación es automáticamente 1) y como se puede ver todos están por debajo de 0.005

Lo más importante de estos valores es que a diferencia del caso del circuito, presentado capítulos atrás, aquí no tenemos un patrón claro en las correlaciones o siquiera llegan a valores que podríamos considerar significativos para el análisis. Por otro lado los valores no presentan ninguna estructura cualitativa para ningún caso.

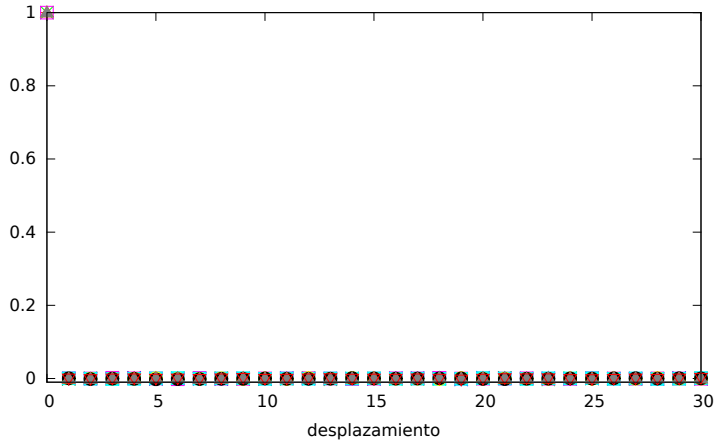

Figura 5.11: Autocorrelación

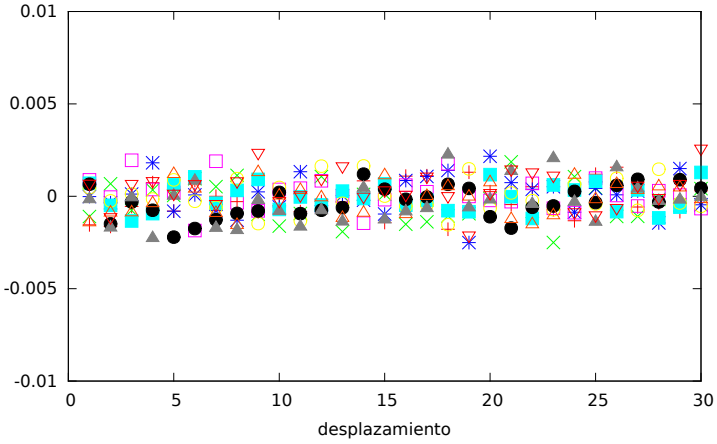

Figura 5.12: Autocorrelación

Capítulo 6

Generación usando Tiempos de Llegada

De los capítulos anteriores podemos ver el contraste entre la fuente de números electrónica y la fuente de números cuántica, además de que la teoría cuántica respalda el indeterminismo de los resultados lo que es una propiedad muy deseable en la generación de números al azar, por otro lado la generación de números usando tiempos de llegada tiene ventajas sobre otros tipos de generación usando sistemas cuánticos. Aquí revisamos distintos trabajos que se relacionan con este esquema de generación a lo largo de los años recientes, pues antes este esquema no era usado, en ellos se trata la caracterización de estos sistemas así como distintas dificultades técnicas de su realización práctica.

6.1. Random number generation based on the time arrival of single photons

Diciembre 2005, [HQM05]

Para la generación usando tiempos de llegada aquí se presenta una propuesta un poco distinta a lo usual, se utiliza un láser muy atenuado dirigido a un detector (actively quenched single-photon detector), por otro lado se usa una electrónica un poco complicada basada en un convertidor de tiempo a voltaje que permite la medición del tiempo de llegada.

El procedimiento de generación es el siguiente: Al tener una detección el convertidor tiempo a voltaje se reinicia para empezar a contar el tiempo, si

en un intervalo de $2\mu s$ no hubo una detección este evento se considera como cero, en caso de haberla se considera como uno.

Debido a que la distribución de valores no es igual para ceros y unos, pues no hay un ajuste de este valor con la distribución que depende del flujo promedio, es necesario realizar una etapa de procesamiento extra, en este caso una renormalización de von Neumann [1] la cual si bien permite igualar el número de ceros y unos disminuye la cantidad de estos.

Para el análisis de la aleatoriedad de salida se calcula la entropía de Shannon, la prueba Chi cuadrada, el promedio, se usan los números para calcular Pi usando Monte Carlo y se mide el coeficiente correlación. Las secuencias generadas pasan todas las pruebas anteriores.

6.2. Patente Pub. No.: US 2006/0010182 A1

Esta patente [AJK06] registrada el 12 de enero del 2006 con los nombres Joseph B. Altepeter, Urbana, IL (US); Evan Jeffrey, Champaign, IL (US); Paul G. Kwiat, Champaign, IL (US) describe un dispositivo que permite la generación de números al azar usando tiempos de llegada entre fotones. Esto presenta un avance respecto a los generadores cuánticos de números al azar debido a que hasta ese momento se generaba un bit por fotón, por ejemplo disparándolo hacia un divisor de haz y detectando en las salidas de este divisor, por tanto el flujo de números se ve limitado por el flujo que puede aceptar el detector y usando el tiempo como variable se puede superar esta barrera logrando generar más bits por fotón, por ejemplo en el experimento del capítulo anterior se dividió la distribución de tiempos en dos regiones para obtener un bit por detección, pero se pudo haber dividido en las partes necesarias 4,8 o incluso 3 para obtener más de un bit por detección.

Dentro de esta patente se menciona la posibilidad de obtener distintas distribuciones en la salida de los números aleatorios variando el flujo de la fuente de fotones a lo largo del tiempo, además de una etapa de salida donde se le quita la posible tendencia que tengan los números generados hacia cierto algún valor, pues esto es señal de una aleatoriedad pobre, esta última característica es de hecho muy común en todo tipo de generadores.

[1] Este procedimiento consiste en tomar una serie de bits de dos en dos y desechar todas las combinaciones '00' y '11', sólo dejando las '01' y '10' si cada bit es independiente la probabilidad de obtener '01' o '10' es la misma aunque '0' y '1' no tengan la misma probabilidad de aparecer. Finalmente se sustituye '01' por '0' y '10' por '1'.

6.3. Photon arrival time quantum random number generation

23 Enero 2009, [WJAK09]

Uno de los problemas que tiene la generación usando tiempos es el rendimiento, para muchas aplicaciones de números al azar se suele necesitar un flujo grande de ellos, mientras que los generadores por software pueden cubrir esta demanda este tipo de generadores no.

Esta es una de las primeras propuestas de usar tiempos de llegada para generar números al azar, es de notar que es relativamente reciente. En este caso la generación tiene el mismo esquema que el de la patente anterior pero se especifica la forma de asociar los números respecto a las detecciones. Se divide un cierto intervalo, donde casi con toda seguridad habrá una detección, y se divide en segmentos iguales asociando un número a la detección dependiendo del segmento donde apareció.

Debido a que los tiempos de llegada entre fotones tienen una distribución exponencial algunos números tendrán una presencia mayor que otros en la salida. El modelo de probabilidad que se propone es el siguiente:

$$P_i = \begin{cases} 0 & si \quad i < \delta \\ \frac{e^{1/\lambda}}{\lambda} e^{-i/\lambda} & si \quad i > \delta \end{cases} \tag{6.1}$$

donde δ es el tiempo muerto del detector. Debido a que la distribución de valores no es constante se debe usar un procesamiento les quite la tendencia.

Para la realización del aparato se usa un APD 100-MMF50-ULN con un tiempo muerto de 45ns y 22 Mcps como flujo máximo. Para el procesamiento indicado se usa el hash SHA-256, hay que notar que este hash a veces da a distintos valores la misma entrada por lo que 2 de los bits generados se usarán para retirar ese efecto de los números de salida. Para la medición de tiempos y la interfaz con la computadora se utiliza un FPGA(Field Programmable Gate Array) y el aparato en conjunto logra 11 Mhz con una resolución de 5ns y 5.5 bits por detección.

Las pruebas de aleatoriedad se realizaron con la suite del NIST, obteniendo en segmentos desde 10MB hasta 200MB de 10 en 10.

6.4. Low-bias high-speed quantum random number generator via shaped optical pulses

Abril 2010, [WK10]

Como continuación de la referencia anterior algunos de los autores de ese trabajo proponen un nuevo esquema de generación usando tiempos de llegada, debido al problema que presenta el post-procesamiento de los datos se busca obtener una distribución de valores de tiempo constante de manera que no sea necesaria la etapa final donde se aplica el hash.

Para ello consideran el mismo arreglo que en su trabajo anterior pero permitiendo que el láser tenga un flujo variable de fotones, de esta manera el número de detecciones esperado entre dos valores de tiempo es:

$$\lambda_{a,b} = \int_a^b \lambda(t)dt \tag{6.2}$$

esto cambia los parámetros de la distribución exponencial de tiempos dentro del intervalo a, b a:

$$\lambda(t)e^{-\int_a^b \lambda(t')dt} \tag{6.3}$$

debido a que deseamos una distribución constante la expresión anterior se debe igualar a una constante. Como se muestra en [WK10] un parámetro de la forma $1/(T-t)$ es una solución para esto donde T es un periodo definido por el usuario.

Como se menciona en [WK10] la modulación deseada se logra notando que la corriente es proporcional con el flujo de fotones así que se regula la corriente según $k/(T-t)$ donde k es una constante de proporcionalidad. Lo anterior es posible gracias a una implementación aproximada de la función usando una electrónica compuesta principalmente por un generador de diente de sierra, un convertidor logarítmico y un circuito diferenciador.

En lo que se refiere a la detección se usa el APD 100-MMF50-ULN y el tiempo medido usando un FPGA igual que en el arreglo anterior.

Con el arreglo anterior es posible llegar a un flujo neto de 126 Mbits por segundo que por razones de estabilidad en la distribución final de los datos se reduce a 119 Mbits por segundo.

Las pruebas de aleatoriedad realizadas es la suite de NIST, todas pasadas satisfactoriamente.

6.5. True random number generator based on discretized encoding of the time interval between photons

Enero 2013, [LWW+13]

Al igual que la propuesta anterior, esta intenta eliminar el procesamiento extra que recibe la salida del generador, pues como se mencionó antes disminuye el flujo de números y por tanto la eficiencia del generador.

Para la generación se usó un láser pulsado (20ps,1310nm) y un atenuador dirigido a un divisor de haz que tiene en sus salidas un par de detectores (id Quantique id 200). Uno de los detectores arranca un convertidor de tiempo a amplitud, el otro detector detiene el convertidor, si el tiempo entre fotones es menor que 2ms entonces se obtendrá un voltaje válido entre 0V y 10V, en otro caso el convertidor se reiniciará. Finalmente la salida es enviada a una computadora donde se discretiza y asignan los valores de salida.

El método de asignación de números es similar a la que usamos aquí para asignar valores a las detecciones, se divide la distribución de tiempos en las partes necesarias(generalmente potencias de 2) y se busca que la probabilidad de cada una sea igual, es decir que:

$$\int_0^a P(t)dt = \int_a^b P(t)dt = \cdots = \frac{1}{2^n} \tag{6.4}$$

donde n está limitado por la resolución temporal del arreglo. A cada uno de estos intervalos se le asigna un número que tendrá n bits y a cada detección se le asignará el número del intervalo al que pertenece.

Las pruebas realizadas en este caso para evaluar la aleatoriedad es la suite ENT que prueba la aleatoriedad con la entropía de Shannon, la prueba chí-cuadrada, el promedio aritmético, calcula π usando un método de Monte Carlo y el coeficiente de correlación serial de la muestra. En todos los casos se obtuvieron buenos resultados que confirman la aleatoriedad.

6.6. Practical and fast quantum random number generation based on photon arrival time relative to external reference

12 Enero 2014, [NZZ+14]

En [NZZ+14] nos encontramos con una propuesta de generación que también usa los tiempos de llegada pero en un esquema diferente. Con tiempo de llegada generalmente se refiere al tiempo que tarda un fotón en llegar desde que el anterior llegó, como se vio anteriormente estos tiempos tienen una distribución exponencial y si se requieren números que tengan una distribución constante, se debe hacer un procesamiento que puede disminuir el flujo de números al azar o incluso presentar una debilidad desde el punto de vista de la seguridad.

Para saltarnos el procesamiento y usar directamente los números medidos en [NZZ+14] se propone usar un periodo de tiempo establecido, del orden del tiempo promedio entre fotones, y dentro de ese periodo de tiempo medir en que momento llegó un fotón. De manera análoga con el procedimiento realizado normalmente con los tiempos de llegada el periodo de tiempo se divide en 2,4,8 o el número deseado, mientras esté de acuerdo con la resolución del aparato, y dependiendo del segmento donde llegó la detección se le asocia un valor. Como se muestra ahí, este esquema permite obtener una distribución de valores prácticamente constante, es decir cada fotón tiene la misma probabilidad de llegar en cualquiera de los segmentos.

Naturalmente esto tiene algunos problemas como que puede no haber detección alguna en el intervalo o que haya 2 pero sólo se cuantifique una, sin embargo el éxito de esta propuesta es la realización de un prototipo que implementa estas características y usa valores de manera que estos problemas no afecten significativamente el resultado logrando una distribución prácticamente plana sin necesidad de procesamiento alguno.

Para la realización del aparato se usa una diodo láser que es fuertemente atenuado usando un atenuador variable, para la detección se usa un APD id100-smf20-std que tiene un tiempo muerto de 45ns y un flujo máximo de 13.9 Mcps, la salida de este es conectada a un TDC(Time to Digital Converter) que permite la medición del tiempo y finalmente es pasado a un FPGA(Field Programmable Gate Array) , que permite una respuesta rápida, que se encarga de la asignación de valores y la interfaz de salida USB 2.0.

Esta realización usa como periodo 40.96 ns y tiene una resolución de 160

ps permite 8 bits por detección y un flujo de 13.6 Mcps, el procedimiento en total alcanza los 106 Mbps, además se usa un procesamiento extra que asegura la aleatoriedad bajando el flujo de bits a 96 Mbps.

Naturalmente las pruebas de aleatoriedad a las que se sometió esta realización es la suite del NIST, todas las pruebas de aleatoriedad fueron pasadas.

Capítulo 7

Conclusiones

Durante este trabajo se buscó explorar distintos aspectos del concepto de aleatoriedad y su relación con los sistemas físicos, con especial interés en los sistemas cuánticos.

El concepto mismo de aleatoriedad tiene distintas caras, se suele usar este término en relación a la ignorancia de información, la falta de estructura y el indeterminismo. La teoría algorítmica de la información(TAI) da un fundamento muy fuerte de la aleatoriedad como falta de estructura, y logra unos resultados intuitivos que respaldan el marco formal que propone.

Desde un punto de vista más práctico la TAI tiene varios resultados que nos confrontan con algunas dificultades conceptuales asociadas a la definición misma de azar y limitan las aplicaciones de la teoría en el sentido de la incomputabilidad. Es aquí donde el teorema de Calude toma fuerza, pues muestra al concepto de normalidad de Borel como necesario para la aleatoriedad dándonos un vínculo entre una prueba aplicable efectivamente y el concepto incomputable de aleatoriedad.

Por otro lado la aplicación práctica de estas ideas para la generación de números al azar de manera electrónica revela la existencia de estructura en señales de ruido electrónico, esto lleva a la necesidad de procesamiento para obtener la aleatoriedad deseada en este tipo de fuentes.

La aleatoriedad de la que se habla en mecánica cuántica suele ser de una naturaleza distinta, pues se refiere generalmente al indeterminismo de los resultados de las mediciones individuales, esto de primera mano no parece relacionado con la falta de estructura, después de todo para hablar de estructura necesitamos un conjunto de datos, sin embargo regularidades en una serie de mediciones sí estarían en contra de este indeterminismo.

De los datos obtenidos midiendo tiempos de llegada de fotones vemos que sin ningún tipo de procesamiento intermedio, salvo la discretización, esta fuente de aleatoriedad presenta buenas propiedades y prácticamente nada de autocorrelaciones lo que la califica como una excelente fuente de números al azar.

Uno de los resultados más importantes que obtuvimos es la normalidad de Borel para los datos obtenidos de los tiempos de llegada. Hay que notar que encontramos una discrepancia importante en comparación con los datos reportados en [CDDS10] donde la generación se hace usando un divisor de haz.

Desde esta perspectiva los generadores de números al azar usando tiempos de llegada de fotones tienen un futuro prometedor. Como se notó en la revisión de otros trabajos relacionados, hay gran interés en este tipo de generación debido a su fácil implementación. En los distintos trabajos revisados se usó un láser atenuado como fuente de fotones o como en este trabajo un cristal no lineal, lo anterior sumado a una electrónica que si bien es bastante compleja hoy día hay soluciones dedicadas a este tipo de problemas como los FPGA, logra una implementación tecnológica sencilla incluso en el sentido comercial.

Por otro lado la fuerza de esta propuesta viene con el flujo de aleatoriedad que puede presentar pues, como se mencionó antes, otras propuestas de generadores cuánticos generan uno o pocos bits por detección. Con la propuesta de dividir la distribución en regiones o usar modulaciones en el flujo de fotones pueden generarse muchos más bits por fotón e incluso lograr que los números generados sigan una distribución predefinida y finalmente con la propuesta de dividir la distribución en regiones de probabilidad igual se solventa la necesidad de una etapa de procesamiento que asegure la aleatoriedad de los números pero que tiene un costo computacional alto en muchos casos.

Si bien es cierto que este tipo de generación tiene aún mucho por desarrollar para convertirse en una solución ampliamente usada, parece que las barreras con las que se encuentra han sido superadas de manera relativamente fácil. Tal vez sea esta la forma cuántica de generación que permita a los dispositivos de uso común acceder a la calidad de azar más alta que se tiene hasta el momento.

Bibliografía

[Abr82] Pais Abraham. Max born and the statistical interpretation of quantum mechanics. *Science*, 218(1193-8), 1982.

[AJK06] J. Altepeter, E. Jeffrey, and P. Kwiat. Quantum random number generator, January 12 2006. US Patent App. 10/885,503.

[Bor26] Max Born. Zur quantenmechanik der stoßvorgänge. *Zeitschrift für Physik*, 37(12):863–867, 1926.

[Cal02] Cristian Calude. *Information and Randomness: An Algorithmic Perspective*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2nd edition, 2002.

[CDDS10] Cristian S. Calude, Michael J. Dinneen, Monica Dumitrescu, and Karl Svozil. Experimental evidence of quantum randomness incomputability. *Phys. Rev. A*, 82:022102, Aug 2010.

[CDS02] Cristian S. Calude, Michael J. Dinneen, and Chi-Kou Shu. Computing a glimpse of randomness. *Experimental Mathematics*, 11(3):361–370, 2002.

[Chr94] Calude Christian. Borel normality and algorithmic randomness. *Developments in Language Theory*, 1994.

[Chr09] Calude Christian. How quantum is quantum randomness? an experimental approach. 2009.

[Dav53] Martin Davis. *Computability and Unsolvability*. Dover, 1953.

[EB09] M. Emile Borel. Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo*, 27(1):247–271, 1909.

[Erw26] Schroedinger Erwin. Note. *Naturwiessenschaften*, 14(644), 1926.

[HQM05] Ling-An Wu Hai-Qiang Ma, Yuejian Xie. Random number generation based on the time of arrival of single photons. *Applied Optics*, 44(36):7760–3, 2005.

[Kie03] Tien D. Kieu. Quantum algorithm for hilbert's tenth problem. *42*, pages 1461–1478, Julio 2003.

[LV08] Ming Li and Paul M.B. Vitnyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 3 edition, 2008.

[LWW+13] Shen Li, Long Wang, Ling-An Wu, Hai-Qiang Ma, and Guang-Jie Zhai. True random number generator based on discretized encoding of the time interval between photons. *J. Opt. Soc. Am. A*, 30(1):124–7, Enero 2013.

[Mat93] Yuri Matiyasevich. *Hilbert's 10th Problem*. The MIT Press, 1993.

[Mec14] John Mechalas. Intel® digital random number generator (drng) software implementation guide, May 2014.

[NZZ+14] You-Qi Nie, Hong-Fei Zhang, Zhen Zhang, Jian Wang, Xiongfeng Ma, Jun Zhang, and Jian-Wei Pan. Practical and fast quantum random number generation based on photon arrival time relative to external reference. *Applied Physics Letters*, 104:051110, 2014.

[RSN+10] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, and San Vo. Statistical test suite for random and pseudorandom number generators for cryptographic applications. *NIST Special Publication*, Abril 2010.

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

[SZ97] Marlan O. Scully and M. Suhail Zubairy. *Quantum Optics*. Cambridge University Press, 1997. Cambridge Books Online.

[Tur37] Alan Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1937.

[WJAK09] Michael A. Wayne, Evan R. Jeffrey, Gleb M. Akselrod, and Paul G. Kwiat. Photon arrival time quantum random number generation. *Journal of Modern Optics*, 56(4):516–22, 2009.

[WK10] Michael A. Wayne and Paul G. Kwiat. Low-bias high-speed quantum random number generator via shaped optical pulses. *Optics Express*, 18(9):9351–57, 2010.