Conjetura de Aaronson-Ambainis

escrito por

RAMON REBULL CAMARASA

Tutor: Antonio Pérez Hernández

Facultad de Ciencias
UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Trabajo presentado para la obtención del título de
Máster Universitario en Matemáticas Avanzadas de la UNED.
Especialidad de Matemáticas Aplicadas

JUNIO 2025

Abstract

Abstract en español:

A las puertas de lo que podría ser la revolución cuántica, una cuestión fundamental es determinar qué problemas pueden beneficiarse de la potencia del paralelismo cuántico y cuáles permanecen fuera de su alcance. En este contexto, han surgido dos conjeturas claves: la Conjetura Cuántica, que plantea una posible caracterización de los problemas que admiten una aceleración cuántica significativa, y la Conjetura Aaronson-Ambainis, que traslada esta cuestión al análisis de funciones booleanas, una área central en el estudio de la complejidad computacional.

En este trabajo exploramos ambas conjeturas, sus fundamentos teóricos y sus implicaciones para la computación cuántica y clásica. Analizamos la relación entre ellas y revisamos algunos de los resultados más relevantes obtenidos en los últimos años, destacando posibles direcciones futuras para abordar las cuestiones que siguen abiertas.

Abstract in English:

On the verge of what could be the quantum revolution, a fundamental question is to determine which problems can benefit from the power of quantum parallelism and which remain beyond its reach. In this context, two key conjectures have emerged: the Quantum Conjecture, which proposes a possible characterization of problems that admit significant quantum speedup, and the Aaronson-Ambainis Conjecture, which extends this question to the analysis of Boolean functions, a central area in computational complexity.

In this work, we explore both conjectures, their theoretical foundations, and their implications for both quantum and classical computation. We examine the relationship between them and review some of the most relevant results obtained in recent years, highlighting potential future directions to address these open questions.

Keywords: Funciones booleanas, Información cuántica, Conjetura Aaronson-Ambainis

Dedicatoria y agradecimientos

A mis padres, por haber puesto siempre provocaciones a mi alcance para estimular mi interés por las matemáticas.

A Martí, por echarme un cable en algunas correcciones en la redacción, y a Inma, por su apoyo y por simular interés en mis explicaciones.

También a David, por las conversaciones que hemos tenido sobre aspectos diversos del trabajo. Por escucharme, por corregirme, y por sus ideas. Tanto las que he aplicado, como las que no.

Y, por supuesto, a Antonio. Por haber orientado este trabajo a un rigor que en un principio no tenía. Por todo el tiempo dedicado en las revisiones, por las correcciones y por las múltiples ideas valiosas sin las cuales no me habría sido posible sacarlo adelante. Por las preguntas que me ha lanzado, y por ayudarme a encontrar las respuestas. Y por la paciencia.

ÍNDICE GENERAL

CAPÍTULO 1

INTRODUCCIÓN

1.1. Historia

La información y la computación cuántica son disciplinas muy jóvenes que surgen como consecuencia del interés en aplicar a la informática algunas de las propiedades de la mecánica cuántica descubiertas durante el siglo XX.

Una de las primeras motivaciones para desarrollar esta area de la teoría de la computación se planteó en 1981, cuando Richard Feynman sugirió que los ordenadores clásicos tienen dificultades para simular procesos cuánticos y propuso la idea de utilizar ordenadores cuánticos para esta tarea [20]. Un año más tarde se demostró la imposibilidad de clonar un estado cuántico [43], hecho fundamental en la comprensión de las bases de lo que más adelante sería la información cuántica.

En 1985, David Deutsch formalizó la idea de Feynman e introdujo el concepto de ordenador cuántico universal [16], ampliando la noción de máquina de Turing para incluir procesos cuánticos. En ese mismo trabajo propuso el que es conocido como algoritmo de Deutsch, que identifica si una función booleana de un bit de entrada y uno de salida es o no constante, ejecutando para ello una única invocación a la función (en vez de las dos necesarias en un ordenador clásico). Este algoritmo es la primera demostración de las posibilidades que tiene la computación cuántica en algunos tipos de problemas.

En la década de los 90 se desarrollaron algunos algoritmos cuánticos de especial importancia:

En 1992, Deutsch y Jozsa generalizaron el algoritmo de Deutsch a funciones de n bits. No se le ha encontrado ninguna aplicación práctica, pero se trata del primer ejemplo de algoritmo cuántico de más de un bit con mejora exponencial respecto a la complejidad de su equivalente clásico (estudiaremos este algoritmo en la Sección 3.3.1).

Dos años después, Peter Shor diseñó el algoritmo de estimación de fase y lo adaptó al problema de factorizar un número natural, con una mejora también exponencial respecto a la factorización con computadores clásicos [40]. Además de su evidente importancia teórica, la posibilidad de factorizar números grandes de forma eficiente abre la puerta a romper las claves RSA en las que se basan la mayor parte de cifrados asimétricos actuales, lo que ha provocado una crisis en la fiabilidad de los protocolos de seguridad que la comunidad matemática e informática todavía hoy están trabajando en solucionar.

En 1995, Lov Grover presentó su algoritmo de búsqueda cuántica [23]. La ventaja de este algoritmo es cuádratica respecto al clásico, lo cual es una mejora mucho más discreta que los que hemos descrito antes. Sin embargo, se trata de un algoritmo de gran importancia por su propósito generalista y por haber servido de inspiración para otros algoritmos cuánticos, o híbridos cuántico-clásicos.

Daniel R. Simon propuso en 1998 el problema de colisión (diferenciar funciones uno a uno vs. dos a uno), y un algoritmo que lo resuelve mejorando de forma superpolinomial el mejor algoritmo clásico [41]. Lo estudiaremos en la Sección 3.3.2.

A finales del siglo XX y durante la primera década del XXI se inició un estudio más formalizado de la complejidad cuántica. En este contexto, se plantea una conjetura (que en este trabajo denominaremos Conjetura Cuántica y en diversas referencias denominan Conjetura Folklore) que afirma que existe una relación polinomial entre la complejidad de cualquier algoritmo cuántico y un algoritmo clásico que lo aproxime en casi todos los puntos. Esta conjetura tiene profundas implicaciones en la caracterización de los problemas en los que la computación cuántica podría obtener una ventaja superpolinomial respecto a la complejidad de sus algoritmos clásicos equivalentes, como veremos en la Sección 3.2, donde la estudiaremos con más detalle.

En 1998, Beals, Buhrman, Cleve, Mosca y de Wolf publicaron un artículo [10] en el que demuestran que la probabilidad de aceptación de todo algoritmo cuántico de caja negra se puede representar con un polinomio multilineal, y establecen una relación entre el grado de este polinomio y la complejidad del algoritmo. Este resultado abre la puerta a muchos otros que verifican cotas en la complejidad cuántica de diversos algoritmos. A modo de ejemplo, entre 2000 y 2003 Ambainis introdujo una técnica de polinomios simétricos para estudiar cotas inferiores en la complejidad de consulta cuántica [6, 7], y Aaronson en [1] y Shi en [5] describieron sendas cotas en la complejidad cuántica del problema de colisión.

El resultado descrito antes de Beals et al. desplazó el estudio de la complejidad cuántica a la teoría de funciones booleanas. En este ámbito, en 2005 O'Donnell, Saks, Schram y Servedio [35], y en 2006 Dinur y Friedgut [18], demostraron resultados que determinan la existencia de una variable con alta influencia en algunos supuestos que estudiaremos de forma más detallada en las Secciones 5.2 y 5.3.

Generalizando estos últimos resultados, Aaronson y Ambainis formularon en 2008 la que se conoce como Conjetura Aaronson-Ambainis (Conjectura AA) [4], que afirma que cualquier polinomio multilineal booleano de grado bajo tiene alguna variable con una influencia significativa (más adelante, en el Capítulo 4, describiremos de forma más precisa y estudiaremos esta conjetura). Además, demostraron que de esta conjetura se deriva la Conjetura Cuántica.

Este es el punto de partida del trabajo. En la próxima sección describiremos con más detalle su objetivo, el contenido que vamos a tratar y su estructura.

1.2. Contenido

El objetivo de este trabajo es estudiar la Conjetura Aaronson-Ambainis, su relación con la computación cuántica, los resultados que se conocen relacionados con ella, y el estado actual de las líneas de investigación orientadas a demostrarla.

El Capítulo 2, el siguiente tras esta introducción, es el de conocimientos previos. Empieza con una sección donde se introducen los principales modelos de computación que más adelante se aplicarán en el trabajo. Siguen dos secciones más donde se exponen las bases de la información cuántica y de los circuitos cuánticos.

Los principales algoritmos cuánticos tratados en este trabajo están basados en el modelo de caja negra. En el Capítulo 3 se expone con más detalle la información necesaria sobre este modelo, se introduce la Conjetura Cuántica y se analiza qué implicaciones tiene esta conjetura sobre la relación de complejidad entre algoritmos clásicos y cuánticos. Al final del capítulo se estudian estos conceptos en dos ejemplos de algoritmos: el algoritmo de Deutsch-Jozsa y el algoritmo de Simon.

En el siguiente capítulo damos un salto a la teoría de funciones booleanas. Introduciremos el concepto de polinomio multilineal, y demostraremos que la probabilidad de aceptación de todo algoritmo cuántico de caja negra puede representarse con este tipo de funciones. En la introducción histórica ya anticipábamos la importancia de este hecho, que nos permitirá conectar la teoría de computación cuántica con la de funciones booleanas. A continuación enunciaremos la Conjetura AA y demostraremos que esta implica la Conjetura Cuántica. Seguramente, este es el resultado más importante del trabajo, así que dedicaremos bastante espacio y esfuerzo en profundizar de forma detallada en esta demostración con la intención de completar algunos pasos poco detallados de la demostración original. También realizamos una aportación original al ampliar las conclusiones a oráculos de m bits de salida, lo que justifica que algunos resultados del trabajo difieran de los disponibles en toda la bibliografía referenciada. En la Sección 4.3.3 incorporamos un esquema de toda la relación de conjeturas, teoremas y corolarios.

Tras haber expuesto todo el contexto relativo a las Conjeturas AA y Cuántica, en el Capítulo 5 hacemos una revisión de varios resultados relacionados con éstas. Se trata de un capítulo esencialmente orientado al estudio de funciones booleanas, y no a circuitos cuánticos, por lo

cual empezamos con una primera sección en la que ampliamos los conceptos de este tipo de funciones que habíamos visto en secciones previas. A continuación exponemos y demostramos varios resultados:

- La Conjetura AA es válida para funciones con rango booleano. La demostración que aportamos se basa en las ideas del artículo referenciado, aunque ha sido necesario incorporar otros resultados para que las conclusiones de ese artículo se ajusten a los términos en que está expresada la conjetura.

- La Conjetura AA sería válida si sustituyéramos la referencia al grado del polinomio por el exponencial de su grado. Hemos combinado ideas de dos artículos distintos para su demostración. Un lema troncal para esta demostración es el Lema 5.1. Su demostración es bastante extensa, por lo que no la incluimos de forma completa, dejando referencia al artículo original para los puntos que no se desarrollan con detalle. A pesar de ello, esta es la demostración a la que hemos dedicado más espacio de todo el trabajo.

- La Conjetura AA es válida para polinómios simétricos. Igual que en secciones anteriores, la demostración está basada en la descrita en el artículo de referencia, pero hemos hecho diversas adaptaciones en su enfoque. El Lema 5.3 es original.

- La Conjetura AA es válida para funciones desacopladas de un bloque si y solo si lo es en el caso general. La demostración de este resultado es bastante más simple que la de los anteriores, aunque también aquí hemos tenido que aplicar diversas adaptaciones a lo descrito en las referencias para que todos los pasos quedaran adecuadamente detallados.

Por razones de espacio y de propósito de este trabajo no ha sido posible revisar todos los resultados que se han desarrollado en relación con la conjetura. Por esta razón hemos incluido una última sección donde hacemos una breve enumeración cronológica de los distintos artículos que se han publicado al respecto, con una idea superficial de su significado y de qué aportan en el propósito de demostrar la conjetura de forma global.

CAPÍTULO 2

CONOCIMIENTOS PREVIOS

2.1. Teoría de la computación

Todo estudio en teoría de la computación debe basarse en un modelo que describa el conjunto de estructuras y operaciones mediante las cuales computar el output de una función dado un input determinado. A partir de los fundamentos de estos modelos se definen conceptos como la complejidad, que son esenciales en el estudio de la algorítmica, tanto clásica como cuántica. Existen múltiples modelos de computación: circuitos lógicos, máquinas de estados finitos, máquinas de Turing, y muchos otros [39, pp.16–23]. Hemos decidido optar por presentar el modelo de circuitos, por ser el más habitual en la literatura, especialmente cuando se presenta la computación cuántica.

2.1.1. Circuitos clásicos

Un computador clásico determinista con N bits de entrada y M de salida es un dispositivo que evalúa una función $F : \{0,1\}^N \to \{0,1\}^M$, de forma que el valor de los M bits de salida viene determinado únicamente por el de los bits de entrada,

$$(y_1, ..., y_M) = F(x_1, ..., x_N).$$

En esta sección estudiaremos la forma de descomponer esta función como composición de diversas **funciones booleanas**, que constan de n bits de entrada y uno de salida [38, p. 3]:

$$f : \{0,1\}^n \to \{0,1\}. \tag{2.1}$$

Para ello, definimos una colección $\mathscr{A}$ de funciones booleanas que denominamos **base**, y que representa el catálogo de funciones disponibles en la descomposición de la función a computar.

Definimos también un conjunto de variables auxiliares $z_1, ..., z_r$ que usaremos para registrar la evaluación de cada una de estas funciones.

Un ejemplo de base sería $\mathscr{A} = \{\neg, \wedge, \vee\}$, compuesta por funciones que se definen de la siguiente forma para todo $x, y \in \{0,1\}$:

$$\neg : \{0,1\} \to \{0,1\}, \quad \neg(x) = 1 - x.$$

$$\wedge : \{0,1\}^2 \to \{0,1\}, \quad \wedge(x,y) = xy,$$

$$\vee : \{0,1\}^2 \to \{0,1\}, \quad \vee(x,y) = x + y - xy,$$

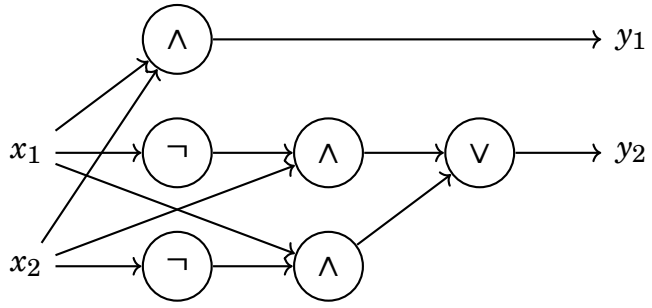
Estas funciones también se denominan NOT, AND y OR, respectivamente, y se suelen denotar como $\neg x := \neg(x), x \wedge y := \wedge(x,y), x \vee y := \vee(x,y)$.

Un **circuito** sobre la base $\mathscr{A}$ es una secuencia de asignaciones $z_j := f_j(u_1, ..., u_s)$, donde $f_j \in \mathscr{A}$ y cada u_i es una variable auxiliar previamente evaluada o una variable de entrada del algoritmo. Se puede representar como un grafo dirigido sin ciclos donde hay N vértices de entrada, uno para cada variable $(x_1, ..., x_N)$, y M de salida. El resto de vértices del grafo se corresponden con las funciones booleanas que se aplican sobre los resultados previos del circuito.

Por ejemplo, sea la siguiente función que evalúa la suma de dos bits

$$f_S : \{0,1\}^2 \to \{0,1\}^2, \qquad f_S(x_1, x_2) = (y_1, y_2) = (x_1 \wedge x_2, (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2)),$$

el siguiente grafo representa un circuito sobre $\mathscr{A} = \{\neg, \wedge, \vee\}$ de f_S:



Denominamos **puerta lógica** al dispositivo físico que evalúa una función de la base $\mathscr{A}$. Para evitar tener que construir muchas puertas lógicas distintas, nos interesará que el cardinal de $\mathscr{A}$ sea lo más reducido posible. Pero también será conveniente que toda función lógica pueda implementarse como circuito sobre esta base (las bases con esta característica se denominan bases completas).

La forma normal disyuntiva (DNF) de una función booleana es una forma estándar de implementarla a partir de la base $\{\neg, \wedge, \vee\}$ anterior. Consiste en encadenar operaciones $\vee$ sobre ítems construidos mediante puertas $\neg$ y $\wedge$. Por ejemplo, la función $f_P : \{0,1\}^3 \to \{0,1\}$ que evalúa

si el input es un número primo (es decir, $f_P(0) = f_P(1) = f_P(4) = f_P(6) = 0, f_P(2) = f_P(3) = f_P(5) = f_P(7) = 1$), tendría la siguiente implementación DNF:

$$f_P(x_1, x_2, x_3) = ((\neg x_1) \wedge x_2 \wedge (\neg x_3)) \vee ((\neg x_1) \wedge x_2 \wedge x_3) \vee (x_1 \wedge (\neg x_2) \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_3)$$

Toda función booleana puede ser implementada de esta forma (ver teorema 2.1 en [26, p. 19]), así que $\mathscr{A} = \{\neg, \wedge, \vee\}$ es una base completa. Sin embargo, la implementación en DNF de una función arbitraria podría requerir hasta 2^N ORs, $N2^{2N}$ ANDs y $N2^N$ NOTs [38, p. 5]. Un objetivo de la teoría de la computación es identificar otros procedimientos más óptimos que permitan reducir el número de puertas lógicas necesario para evaluar una función booleana determinada.

2.1.2. Complejidad

Las funciones booleanas a computar suelen definirse con la intención de resolver un problema. Por ejemplo, supongamos que queremos idear un algoritmo que determine si un número natural dado es o no primo. La función f_P descrita antes cumpliría este propósito para números representables con 3 bits, pero para resolver este problema para el número 13 o 24 podríamos necesitar otras funciones $f_4 : \{0,1\}^4 \to \{0,1\}$ o $f_5 : \{0,1\}^5 \to \{0,1\}$, que requerirían una implementación de circuitos lógicos distinta. Dado que hay infinitos números naturales pero solamente podemos codificar un número finito de inputs, consideramos una familia de funciones $(f_n)_{n\in\mathbb{N}}$ [1] donde cada $f_n : \{0,1\}^n \to \{0,1\}$ asigna valor 1 a una cadena de n bits si el número asociado a dicha cadena es primo y el valor 0 si no lo es.

Sea $(C_n)_{n\in\mathbb{N}}$ la familia de circuitos que implementan cada función f_n, para que ésta se considere admisible debe satisfacer dos condiciones: ser consistente, lo que significa que $C_n(x) = C_m(x)$ si $n < m$ y x es un número de n bits o menos; y que tenga cierta regularidad, es decir, que la implementación de los C_n obedezca a una regla natural e identificable (más formalmente, se requiere que la descripción de estos circuitos pueda ser descrita como una máquina de Turing, aunque nosotros nos limitaremos a un nivel más intuitivo de regularidad). Estas familias se denominan **familias uniformes de circuitos**.

Definición 2.1. Una familia uniforme de circuitos $(C_n)_{n\in\mathbb{N}}$ es de **complejidad de orden** de una función $c : \mathbb{N} \to \mathbb{R}^+$ si existe una constante $k > 0$ tal que

$$|C_n| < kc(n), \qquad \text{para todo } n \in \mathbb{N},$$

donde $|C_n|$ se corresponde con el número de puertas lógicas que contiene el circuito C_n.

En este caso, denotamos que $(C_n)_{n\in\mathbb{N}}$ es de complejidad $O(c)$.

Como caso particular, una familia de circuitos $(C_n)_{n\in\mathbb{N}}$ es de **complejidad de orden polinomial** si

$$|C_n| < \text{poly}(n), \qquad \text{para todo } n \in \mathbb{N},$$

[1] En este trabajo asumiremos en todas las referencias al conjunto de los números naturales $\mathbb{N}$ que éste no incluye el elemento 0.

donde poly(n) es un polinomio de variable n.

Finalmente, denominamos **problemas de clase P** a aquellos que tienen una familia de circuitos de complejidad de orden polinomial.

2.1.3. Computación reversible

El modelo de computación reversible se basa en emplear funciones invertibles (i.e., biyectivas).

$$f : \{0,1\}^n \to \{0,1\}^n$$

Aunque pueda parecer un modelo más restringido que el modelo general, no es así. De hecho, toda función booleana $f : \{0,1\}^n \to \{0,1\}$ se puede implementar como una función invertible $\tilde{f}$ de la siguiente forma

$$\tilde{f} : \{0,1\}^{n+1} \to \{0,1\}^{n+1}, \qquad \tilde{f}(x,y) = (x, y \oplus f(x)), \tag{2.2}$$

donde $x \in \{0,1\}^n$, $y \in \{0,1\}$, y $\oplus$ es el OR exclusivo (XOR). Si asignamos $y = 0$, el último bit del retorno de $\tilde{f}$ devolverá el valor de $f(x)$.

De forma análoga, toda función $f : \{0,1\}^n \to \{0,1\}^m$ puede implementarse también como una función invertible

$$\tilde{f} : \{0,1\}^{n+m} \to \{0,1\}^{n+m}, \qquad \tilde{f}(x,y) = (x, y \oplus f(x)). \tag{2.3}$$

Esta identificación entre f y $\tilde{f}$ permite embeber el modelo de circuitos (no reversibles) de la sección anterior en forma de circuitos reversibles de manera natural, respetando su complejidad.

Veamos algunos ejemplos:

Puerta NOT

La puerta NOT que ya hemos visto antes implementa la única función invertible no trivial de un bit de entrada:

$$f : \{0,1\} \to \{0,1\}, \qquad f(x) = 1 - x.$$

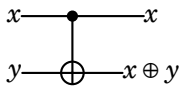
Observamos que efectivamente es invertible, pues $f(f(x)) = f(1-x) = 1-(1-x) = x$ para todo $x \in \{0,1\}$.

Puerta CNOT

La puerta CNOT o controlled-NOT es la puerta que implementa la función:

$$f : \{0,1\}^2 \to \{0,1\}^2, \qquad f(x,y) = (x, x \oplus y).$$

Se representa en un circuito de la siguiente forma:

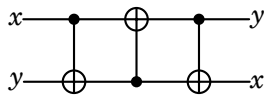


Puerta swap

La puerta swap es la puerta que implementa la función

$$f : \{0,1\}^2 \to \{0,1\}^2, \qquad f(x,y) = (y,x).$$

Se puede construir como concatenación de puertas CNOT de la siguiente forma,



circuito que genera esta secuencia de transformaciones sobre los datos de entrada:

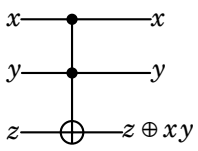
$$(x,y) \;\to\; (x, x\oplus y) \;\to\; (x\oplus(x\oplus y), x\oplus y) = (y, x\oplus y) \;\to\; (y, (x\oplus y)\oplus y) = (y,x)$$

Puerta de Toffoli

La puerta de Toffoli implementa la función

$$f : \{0,1\}^3 \to \{0,1\}^3, \qquad f(x,y,z) = (x,y,z\oplus xy),$$

y se representa en un circuito como



Esta puerta tiene la particularidad de actuar como puerta reversible universal, ya que a partir de ésta se pueden emular las puertas NOT, AND y OR: si $x = y = 1$, el tercer bit devuelve $\neg z$ (NOT); si $z = 0$, devuelve $x \wedge y$ (AND); y es posible también construir una puerta OR como composición de las funciones anteriores, ya que $x \vee y = \neg((\neg x) \wedge (\neg y))$.

2.1.4. Árboles de decisión

Sea una función booleana $f : \{0,1\}^n \to \{0,1\}$ que recibe un input $X = (x_1, ..., x_n)$, un **árbol de decisión determinista** es un árbol binario dirigido T, tal que [12, pp.24–25]:

- Cada hoja del árbol está etiquetada con un valor en $\{0,1\}$.
- Cada nodo (que no sea una hoja) está etiquetado con una variable x_i, $i = 1, ..., n$.
- Todos los nodos (distintos a una hoja) tienen dos arcos salientes. Uno está etiquetado con valor 0 y el otro con valor 1.

- Para evaluar $f(x_1,...,x_n)$ recorremos una rama eligiendo en cada nodo de etiqueta x_i el arco correspondiente al valor del input x_i. Es decir, si la raíz tiene etiqueta x_{i_1}, se recorrerá el arco 0 en caso que $x_{i_1} = 0$ y el arco 1 en caso contrario. Realizaremos la misma acción en el nodo destino del arco, y así sucesivamente hasta llegar a una hoja.

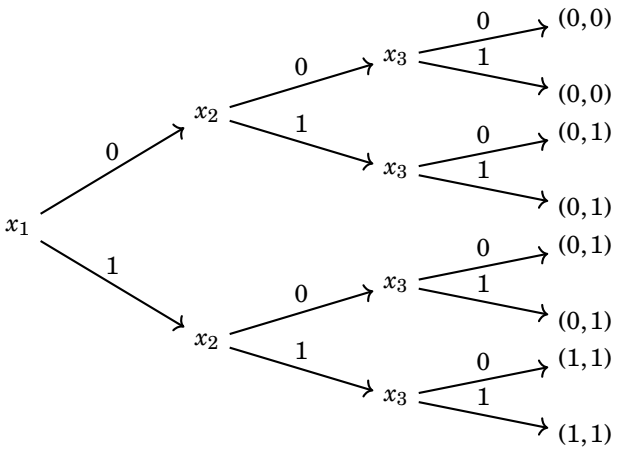
  El árbol debe construirse de tal forma que, para todo input $X = (x_1,...,x_n)$, la etiqueta de la hoja destino coincida con la evaluación $f(X)$.

Esta forma de computación es fácilmente generalizable a funciones $f : \{0,1\}^n \to \{0,1\}^m$, sustituyendo los etiquetados de las hojas por elementos de $\{0,1\}^m$.

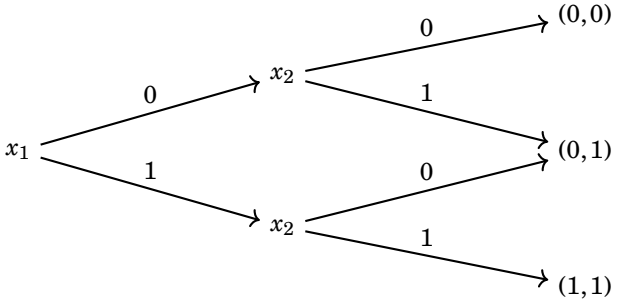
Veamos un ejemplo correspondiente a la siguiente función:

$$f : \{0,1\}^3 \to \{0,1\}^2 , \qquad f(x_1,x_2,x_3) = (x_1 \wedge x_2, x_1 \vee x_2).$$

Un posible árbol de decisión sería



pero habría muchas otras formas de construir árboles alternativos con el mismo resultado. Algunas de ellas involucrarían árboles de menor profundidad, como la siguiente:



Denominamos **árboles de decisión óptimos** a aquellos que tienen profundidad mínima (es decir, que la longitud de la rama más larga sea mínima). Redefinimos el concepto de **complejidad** de la Sección 2.1.2 para este caso como la profundidad del árbol de decisión óptimo que resuelve el problema.

2.1.5. Modelo de queries

Los modelos de computación estudiados hasta ahora definen el input del algoritmo como una cadena de bits. Estos modelos son especialmente adecuados para resolver problemas que consisten en transformar un conjunto de datos en otro (y que, por lo tanto, son directamente modelizables como una función numérica).

Vamos a plantear un problema de distinta naturaleza. Supongamos que tenemos un **oráculo o caja negra**: una función $f : \{0,1\}^n \to \{0,1\}$ de la que desconocemos algunas características, pero que tenemos la capacidad de evaluar (denominamos **query** a la acción de solicitar una evaluación de f). El problema consiste en averiguar si este oráculo cumple una determinada propiedad, y para ello podemos invocar las queries que sean necesarias. Interpretamos cada una de estas queries como "preguntas" que realizamos al oráculo, y la evaluación de f como la respuesta del oráculo a estas preguntas. La sucesión de preguntas (queries) distintas y la interpretación de sus respuestas (evaluaciones) tendría que ayudarnos a discernir si se cumple o no la propiedad evaluada.

El objetivo es diseñar un algoritmo que tome f como input y devuelva un valor en $\{0,1\}$ como output (1 si f cumple la propiedad que queremos discernir, o 0 en caso contrario). Veamos una idea que nos ayudará en este propósito: si definimos un vector que contiene la imagen por f de cada elemento del dominio, éste tiene dimensión $N = 2^n$ y se puede representar como $X = (f(x))_{x\in\{0,1\}^n} \in \{0,1\}^N$. Observamos que existe una correspondencia biunívoca entre el vector y la función, con lo que podemos tomar X como representante de f y usarlo como input del algoritmo que vamos a diseñar.

Veamos un ejemplo. Para $n = 2$ pretendemos identificar si una función $f : \{0,1\}^2 \to \{0,1\}$ actúa o no como una de las siguientes puertas lógicas: AND, OR o XOR. Observamos que los vectores asociados a las funciones que cumplen la propiedad son: $(0,0,0,1)$ el AND, $(0,1,1,1)$ el OR, $(0,1,1,0)$ el XOR, así que una posible representación funcional del algoritmo que realiza esta validación podría ser $F : \{0,1\}^4 \to \{0,1\}$, $F(x_1,x_2,x_3,x_4) = \neg x_1 \wedge ((\neg x_2 \wedge \neg x_3 \wedge x_4) \vee (x_2 \wedge x_3))$ que, si lo implementáramos como árbol de decisión, tendría profundidad mínima 4 (en caso de existir un árbol de decisión con profundidad menor, cada una de sus hojas representaría un número par de inputs; sin embargo, en este caso esto no es posible, ya que el cardinal de la preimagen de 1 es impar, $|F^{-1}(1)| = 3$).

Otro ejemplo podría consistir en evaluar si la función $f : \{0,1\}^2 \to \{0,1\}$ es no creciente ($x < y \Rightarrow f(x) \geq f(y)$, para todo $x, y \in \{0,1\}^2$) y no constante. Las representaciones vectoriales de las funciones que cumplen este criterio son: $(1,1,1,0),(1,1,0,0),(1,0,0,0)$, y con un argumento análogo al del problema anterior concluimos que el árbol de decisión correspondiente también tiene profundidad mínima 4.

A partir de los dos problemas anteriores estudiamos un tercer ejemplo. Supongamos la función $f : \{0,1\}^2 \to \{0,1\}$ que sabemos que está en uno de los dos supuestos previos (implementa una

puerta lógica AND, OR o XOR; o bien es no creciente y no constante), y planteamos el problema de identificar a cuál de los dos casos se corresponde. Nos interesaría un algoritmo que devuelva 1 si f implementa una puerta lógica, o 0 si es no creciente y no constante. Hay una cuestión interesante que diferencia este caso de los anteriores: el problema involucra la promesa de que, de entre todas las posibles funciones f definibles con el dominio e imagen anteriores, sólo las correspondientes a los vectores $S = \{(0,0,0,1),(0,1,1,1),(0,1,1,0),(1,1,1,0),(1,1,0,0),(1,0,0,0)\}$ son inputs del problema, y lo que esperamos del algoritmo es que discierna la caracterización sólo entre estos inputs. Observamos que, en este caso, la función $F : S \subset \{0,1\}^4 \to \{0,1\}$, $F(x_1,x_2,x_3,x_4) = \neg x_1$ resuelve el problema y tiene asociado un circuito mucho más simple que los de los ejemplos previos, con un árbol de decisión de profundidad 1. De forma análoga a este ejemplo, más adelante veremos que en algunas ocasiones esta promesa que restringe el conjunto de posibles inputs permite generar algoritmos más eficientes que sin la promesa.

Lo descrito hasta ahora admite una generalización que nos interesará revisar, ya que se corresponde con algunos de los problemas que estudiaremos más adelante. Ésta consiste en ampliar la definición del oráculo como una función $f : \{0,1\}^n \to \{0,1\}^m$ (antes lo habíamos definido para el caso $m = 1$). En esta ocasión, el vector que representa la función tendrá dimensión $N = m2^n$, puesto que en este caso cada input genera m bits de salida.

Formalizamos esta generalización modelizando el problema como la evaluación de la siguiente función

$$F : S \subseteq \{0,1\}^N \to \{0,1\}, \tag{2.4}$$

donde S representa el conjunto de inputs del problema y $N = m2^n$ el número de bits de este input. La evaluación de F para un vector X (que representa la función f) devolverá un valor booleano que identifica si la función f cumple o no la propiedad que estamos evaluando. En este sentido, definimos que el algoritmo "se acepta" para un oráculo f correspondiente a $X \in \{0,1\}^N$ si $F(X) = 1$.

Como veremos más adelante, es relevante saber qué cardinal tiene el conjunto S anterior. Si $S \neq \{0,1\}^N$, indicamos que la fracción de datos para los que está definido el problema es $|S|/2^N$ y que se trata de un **problema sobre inputs parciales**. En caso contrario, si $S = \{0,1\}^N$, la fracción de datos es 1 y lo denominamos **problema sobre inputs globales**.

Igual que en la sección anterior, también en este caso adaptaremos el concepto de **complejidad** detallado en la Sección 2.1.2, orientándolo al recuento de queries invocadas. Es decir, sustituyendo en la Definición 2,1 la referencia al cardinal del circuito por el número de queries invocadas:

Definición 2.2. Una familia de circuitos de caja negra $(Q_n)_{n\in\mathbb{N}}$ es de **complejidad de queries de orden** de una función $q : \mathbb{N} \to \mathbb{R}^+$ si existe una constante $k \in \mathbb{R}^+$ tal que

$$\mathrm{NQ}(Q_n) < kq(n), \qquad \text{para todo } n \in \mathbb{N}, \tag{2.5}$$

donde NQ representa el número de queries invocadas en el circuito.

En los casos en que la caja negra contenga un número acotado de puertas lógicas (que es lo habitual en los algoritmos conocidos), el orden de complejidad de queries será equivalente a la complejidad que definimos en el apartado de circuitos clásicos. Ello es debido a que la cota de puertas lógicas que implementan el oráculo puede ser absorbida por la constante multiplicativa k de (2.5). En adelante nos basaremos en esta equivalencia para poder comparar algoritmos clásicos con algoritmos de caja negra a partir de sus respectivas definiciones de complejidad. Si en algún caso la caja negra contuviera un número no acotado de puertas lógicas, habría que tener en cuenta este factor en la comparación de ambas complejidades.

2.1.6. Computabilidad

En [21, p. 20] se describe la clase de funciones computables como el conjunto de problemas eficientemente decidibles por un algoritmo de forma que todos sus outputs aproximen la función con una probabilidad superior a $1 - 1/n$, para algún $n \in \mathbb{N}$.

En la definición anterior suele elegirse 2/3 para esta cota de la probabilidad como convenio generalmente aceptado (caso $n = 3$), pero en realidad cualquier otra cota superior a 1/2 tendría el mismo efecto. Lo importante es que la probabilidad de acierto sea estrictamente superior a la de la elección aleatoria pura (1/2), de forma que se pueda amplificar mediante repetición y voto mayoritario. Si se dispusiera de un algoritmo con una tasa de acierto $1/2 < C < 2/3$, se podría mejorar la fiabilidad repitiendo diversas veces la ejecución y aplicando un esquema de decisión basado en la mayoría de resultados obtenidos. Esto es conocido como amplificación de probabilidades. En [26, Teorema 4.1 pp. 36–38] se demuestra que tras un número $k = O(1)$ de iteraciones, la probabilidad de error se puede reducir a 1/3 o a cualquier otra cota arbitrariamente pequeña contenida en $(0, 1/2)$.

En adelante, cuando nos refiramos a la computación de funciones en este trabajo, presupondremos que se trata de funciones bajo esta premisa de computabilidad, y que el algoritmo asociado cumple con la restricción de error máximo 1/3 descrita antes.

2.2. Información cuántica

2.2.1. El qubit

De la misma forma que la computación clásica está basada en el concepto de bit, el elemento de información elemental en computación cuántica es el quantum bit (o **qubit**). El qubit tiene ciertas analogías con el bit, pero también algunas características diferenciales que son la esencia del paradigma cuántico.

El estado de un bit clásico toma valor en $\{0, 1\}$. Para estudiar qué valores puede tomar el estado de un qubit, introducimos el primer postulado de la mecánica cuántica:

Postulado 1. *[32, p. 80] Todo sistema físico aislado tiene asociado un espacio vectorial complejo de dimensión finita dotado de un producto interno (por lo tanto, un espacio de Hilbert) que denominamos espacio de estados del sistema. El sistema se describe de forma completa por su vector de estado, que es un vector unidad en el espacio de estados del sistema.*

El espacio de estados correspondiente a un qubit es $\mathscr{H} \equiv \mathbb{C}^2$, que tiene estructura de espacio vectorial y además tiene definido el siguiente producto interno

$$x \cdot y = \sum_{i=1}^{2} \overline{x_i} y_i, \forall x, y \in \mathbb{C}^2, \tag{2.6}$$

donde $\overline{x_i}$ es el conjugado de la i-ésima coordenada de x, e y_i es la i-ésima coordenada de y. Así pues, se trata de un espacio de Hilbert. Este producto interno induce una norma $\|x\| = \sqrt{x \cdot x}$ para todo $x \in \mathbb{C}^2$, que coincide con el módulo definido en el cuerpo de los complejos, $|x|$.

Los estados de un qubit se encuentran restringidos al conjunto de vectores unidad dentro del espacio de Hilbert anterior:

$$S(\mathscr{H}) \equiv \{z \in \mathbb{C}^2 : |z| = 1\}. \tag{2.7}$$

El estado genérico ψ de un qubit suele representarse en notación de Dirac (o notación braket) como $|\psi\rangle$ (ket de ψ), y su dual respecto al producto interno (2.6) como $\langle\psi|$ (bra de ψ). El producto interno, en el caso de aplicarse entre dos qubits $|\psi\rangle, |\phi\rangle \in S(\mathscr{H})$, se representa como $\langle\psi|\phi\rangle$.

Hay dos estados especiales, que denotamos $|0\rangle$ y $|1\rangle$, que forman una base ortonormal en $\mathscr{H}$ denominada **base computacional** (en ocasiones nos referiremos a ellos también como estados básicos). A diferencia del bit, cuyo valor está restringido al binario $\{0,1\}$, el qubit puede tomar valores en combinación lineal compleja de los estados de la base computacional $|0\rangle, |1\rangle$ de la siguiente forma:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \qquad \alpha, \beta \in \mathbb{C}, \qquad |\alpha|^2 + |\beta|^2 = 1$$

A estos estados intermedios los llamamos estados en **superposición**, y los coeficientes complejos α, β se denominan **amplitudes** del estado respecto a $\{|0\rangle, |1\rangle\}$. La restricción en los valores que pueden tomar sus módulos es debida a la necesidad de que $\| |\psi\rangle \| = 1$ (más adelante veremos una justificación de por qué se requiere una norma unitaria).

En general, representaremos algebraicamente el estado de un qubit como un vector columna que contiene sus amplitudes respecto a $\{|0\rangle, |1\rangle\}$. Por ejemplo, los vectores que representan los estados $|\psi\rangle, |0\rangle$ y $|1\rangle$ descritos antes son:

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \qquad |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.8}$$

Esta representación puede ampliarse para el dual del estado y los productos entre estado y dual. Por ejemplo, si $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ y $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$,

$$\langle\psi_1| = \begin{bmatrix} \alpha_1 & \beta_1 \end{bmatrix}, \qquad \langle\psi_1|\psi_2\rangle = \begin{bmatrix} \alpha_1 & \beta_1 \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \beta_2 \end{bmatrix} = \alpha_1\alpha_2 + \beta_1\beta_2,$$

$$|\psi_1\rangle\langle\psi_2| = \begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix}\begin{bmatrix}\alpha_2 & \beta_2\end{bmatrix} = \begin{bmatrix}\alpha_1\alpha_2 & \alpha_1\beta_2\\ \beta_1\alpha_2 & \beta_1\beta_2\end{bmatrix}.$$

Otra base ortonormal de gran importancia en información cuántica es la *base de Hadamard*, formada por

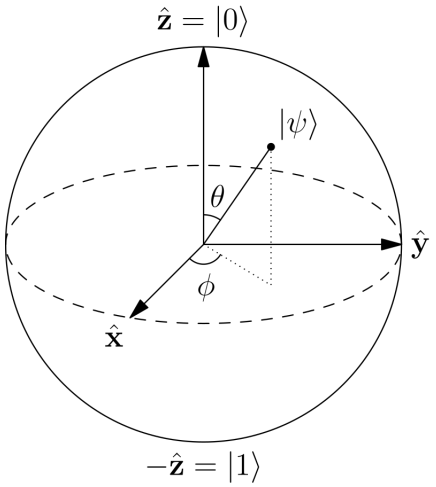
$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, \qquad |-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle. \tag{2.9}$$

Verificamos que efectivamente es ortonormal, ya que:

$$\langle+|+\rangle = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} = 1, \qquad \langle-|-\rangle = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} + \left(-\frac{1}{\sqrt{2}}\right)\left(-\frac{1}{\sqrt{2}}\right) = 1,$$
$$\langle+|-\rangle = \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}\left(-\frac{1}{\sqrt{2}}\right) = 0$$

A lo largo del trabajo haremos múltiples referencias a estos estados. Según lo requiera el contexto, usaremos la notación $|+\rangle, |-\rangle$, o expandiremos la expresión (2.9) como combinación lineal de la base computacional.

Para visualizar geométricamente el estado de un qubit, suele usarse la esfera de Bloch. Esta representación consiste en asociar el qubit a un punto de la superficie de una esfera de radio unitario tal como muestra el siguiente diagrama:



El punto correspondiente al estado $|0\rangle$ tiene coordenadas canónicas $(0,0,1)$, y el correspondiente a $|1\rangle$, $(0,0,-1)$. En general, la posición del punto en la esfera viene fijada por los dos ángulos del diagrama: $\theta \in [0,\pi]$, $\phi \in [0,2\pi)$. A partir de estos ángulos podemos determinar a qué estados se corresponde un punto de la esfera

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle\right), \qquad \gamma \in [0,2\pi). \tag{2.10}$$

El factor $e^{i\gamma}$ se denomina fase global del estado. Más adelante veremos que dos estados que difieren en esta fase son indistinguibles, por lo que se consideran el mismo estado. Esto es lo que justifica que este factor no sea tenido en cuenta en la representación geométrica de la esfera de Bloch.

2.2.2. Operadores y puertas lógicas

En computación clásica, las operaciones que se pueden realizar sobre un bit vienen limitadas por las únicas cuatro funciones lógicas de un bit de entrada $f : \{0,1\} \to \{0,1\}$ que existen: la asignación a 0, la asignación a 1, la identidad, y la negación. El NOT binario es una puerta lógica que implementa la negación, intercambiando los valores 0 y 1 del bit.

También en este aspecto la información cuántica nos aporta un abanico de posibilidades más amplio que la clásica. El hecho de que los qubits puedan tomar infinitos estados distintos da pie a definir operaciones entre ellos mucho más variadas, que quedan fundamentadas por el segundo postulado de la mecánica cuántica:

Postulado 2. *[32, p.81] La evolución de un sistema cuántico cerrado viene descrita por una transformación unitaria. Es decir, el estado $|\psi\rangle$ del sistema en el instante t_1 está relacionado con el estado $|\psi'\rangle$ del sistema en el instante t_2 por un* ***operador*** *unitario U que depende sólo de los momentos t_1 y t_2.*

$$|\psi'\rangle = U\,|\psi\rangle\,.$$

Por lo tanto, un operador es una aplicación $U : \mathscr{H} \to \mathscr{H}$ que transforma el estado de un qubit en otro estado válido, es decir: $U(S(\mathscr{H})) \subset S(\mathscr{H})$. Una consecuencia de este hecho es que U es unitario ($U^\dagger U = I$), y además es invertible (ya que $U^{-1} = U^\dagger$).

Algunos ejemplos de operadores unitarios son:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Especialmente importantes son el operador X, que intercambia las amplitudes de un qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

y se corresponde con la función invertible NOT revisada en la Sección 2.1.3; y el operador H, que transforma la base computacional en la base de Hadamard (y viceversa):

$$H\,|0\rangle = |+\rangle\,, \quad H\,|1\rangle = |-\rangle\,, \qquad\qquad H\,|+\rangle = |0\rangle\,, \quad H\,|-\rangle = |1\rangle\,.$$

Además, como $HH\,|0\rangle = |0\rangle\,, HH\,|1\rangle = |1\rangle$, entonces $H = H^{-1}$, y por lo tanto H es involutiva.

Los operadores anteriores se implementan con las puertas X, Y, Z (que también denominamos Pauli-X, Pauli-Y, Pauli-Z), y con la puerta de Hadamard, respectivamente. La representación de estas puertas en un circuito cuántico es la siguiente:

2.2.3. Medición

Tal como hemos visto antes, el estado de un qubit puede evolucionar como sistema cuántico cerrado a lo largo del tiempo. En algún momento podría interesarnos observar el estado de este qubit, tal como hacemos habitualmente con los bits en computación clásica. El siguiente postulado de la mecánica cuántica explica cómo funciona esta observación, que es una interacción con el sistema cuántico que deja de ser cerrado en ese instante.

Postulado 3. *[32, p.84] Las mediciones cuánticas vienen descritas por una colección de operadores de medición $\{M_m\}$, que actúan sobre el espacio de estado del sistema medido. El índice m se corresponde con los resultados que se pueden obtener en la observación. Si el estado del sistema cuántico es $|\psi\rangle$ en el instante previo a la medición, entonces la probabilidad de que produzca el resultado m es*

$$P(m) = \langle\psi| M_m^\dagger M_m |\psi\rangle,$$

y el estado del sistema después de la medición será

$$\frac{M_m |\psi\rangle}{\sqrt{\langle\psi| M_m^\dagger M_m |\psi\rangle}}.$$

Los operadores de medida satisfacen la ecuación de completitud

$$\sum_m M_m^\dagger M_m = I,$$

cosa que justifica que la suma de probabilidades de las distintas observaciones sea 1:

$$\sum_m P(m) = \sum_m \langle\psi| M_m^\dagger M_m |\psi\rangle = \langle\psi| \left(\sum_m M_m^\dagger M_m\right) |\psi\rangle = \langle\psi|\psi\rangle = 1$$

La observación del estado de un circuito suele realizarse respecto a la base computacional. En este caso, los posibles resultados de la medición son 0 y 1, y los operadores de medición asociados $M_0 = |0\rangle\langle 0|$ y $M_1 = |1\rangle\langle 1|$. Veamos qué sucede después de medir el qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

La probabilidad de observar 0 es

$$P(0) = \langle\psi| M_0^\dagger M_0 |\psi\rangle = |\alpha|^2.$$

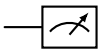
En este caso, el sistema evolucionará al siguiente estado:

$$\frac{M_0 |\psi\rangle}{\sqrt{\langle\psi| M_0^\dagger M_0 |\psi\rangle}} = \frac{\alpha |0\rangle}{|\alpha|} \tag{2.11}$$

En vista del Postulado 3, al medir dos estados que difieren en una fase global, los posibles resultados y sus probabilidades son idénticos. En efecto, si $|\psi'\rangle = e^{i\gamma}|\psi\rangle$, entonces la probabilidad de producir un resultado m es la misma en $|\psi\rangle$ y en $|\psi'\rangle$, puesto que $\langle\psi'| M_m^\dagger M_m |\psi'\rangle = \langle\psi| e^{-i\gamma} M_m^\dagger M_m e^{i\gamma} |\psi\rangle = \langle\psi| M_m^\dagger M_m |\psi\rangle$. Esto significa que ambos estados son indistinguibles (aunque como vectores sean distintos), por lo que se considera que los dos vectores representan el mismo estado [32, p.93]. En el caso de (2.11), el multiplicador $\alpha/|\alpha|$ es un complejo $e^{ki\pi}$, para algún $k \in \mathbb{Z}$. En consecuencia, el estado postmedición del resultado 0 es indistinguible de $|0\rangle$.

De forma análoga, obtenemos que la probabilidad de medir 1 es $P(1) = |\beta|^2$, y el estado después de realizar la medición $|1\rangle$.

Habitualmente, la medición es la última puerta aplicada en un algoritmo cuántico, y se representa de la siguiente forma en un circuito cuántico:



2.3. Sistemas múltiples

2.3.1. Múltiples qubits

Igual que sucedía en el caso de un qubit, los valores de los estados correspondientes a n qubits están contenidos en un espacio de Hilbert. Para describir más formalmente su estructura y la forma en que la modelizamos matemáticamente, definimos el producto tensorial [15, p.121].

Definición 2.3. Sean V, W dos $\mathbb{K}$-espacios vectoriales de dimensiones n y m respectivamente, el producto tensorial $V \otimes W$ es un espacio vectorial nm-dimensional generado por los elementos $v \otimes w$ (donde $v \in V, w \in W$) que satisfacen las siguientes propiedades:

- $(v_1 + v_2) \otimes w = (v_1 \otimes w) + (v_2 \otimes w)$, $\forall v_1, v_2 \in V$.
- $v \otimes (w_1 + w_2) = v \otimes w_1 + v \otimes w_2$, $\forall w_1, w_2 \in W$.
- $\alpha(v \otimes w) = (\alpha v) \otimes w = v \otimes (\alpha w)$, $\forall \alpha \in \mathbb{K}$.

Usaremos la notación $V^{\otimes p}, v^{\otimes p}$ para referirnos a la iteración de p productos tensoriales sobre el espacio V o sobre el elemento v.

A partir de los productos internos de los espacios V, W definimos un producto interno en $V \otimes W$ de la siguiente forma [32, p. 73]

$$\left\langle \sum_i |v_i\rangle \otimes |w_i\rangle \ , \ \sum_j |v'_j\rangle \otimes |w'_j\rangle \right\rangle = \sum_{ij} \langle v_i | v'_j \rangle \langle w_i | w'_j \rangle ,$$

lo que induce la norma $\|x\| = \sqrt{\langle x,x\rangle}$, para todo $x \in V \otimes W$.

Un sistema de n qubits tiene como espacio de estados $\mathscr{H}_n = \left(\mathbb{C}^2\right)^{\otimes n}$, y sus estados se encuentran restringidos al conjunto siguiente:

$$S(\mathscr{H}_n) \equiv \left\{z \in (\mathbb{C}^2)^{\otimes n} : \|z\| = 1\right\}.$$

Definimos en $S(\mathscr{H}_n)$ el conjunto $\{|0\rangle, |1\rangle\}^{\otimes n}$, formado por elementos de la forma $|k_1\rangle \otimes ... \otimes |k_n\rangle$, donde $k_i \in \{0,1\}$ para todo $i \in \{1,...,n\}$ (en adelante denotaremos $[n]$ al conjunto $\{1,...,n\}$, así que en este caso escribiríamos $i \in [n]$). Cuando no existan posibles ambigüedades y convenga según el contexto, usaremos la notación $|k_1,...,k_n\rangle$, o incluso en ocasiones omitiremos las comas.

Veamos que al ser $\{|0\rangle, |1\rangle\}$ una base ortonormal en $\mathscr{H}$, el conjunto $\{|0\rangle, |1\rangle\}^{\otimes n}$ también lo es en $\mathscr{H}_n$: en efecto, el producto interno entre dos elementos distintos de este conjunto $|k_1,...,k_n\rangle$ y $|k'_1,...,k'_n\rangle$ es nulo, ya que habrá algún índice $i \in [n]$ tal que $k_i \neq k'_i$, con lo que $\langle k_i, k'_i\rangle = 0$, y

$$\left\langle |k_1\rangle \otimes ... \otimes |k_n\rangle \ , \ |k'_1\rangle \otimes ... \otimes |k'_n\rangle \right\rangle = \prod_{i\in[n]} \langle k_i, k'_i\rangle = 0;$$

así mismo, de la ortonormalidad de $\{|0\rangle, |1\rangle\}$ se deriva que el producto interno de dos elementos iguales de $\{|0\rangle, |1\rangle\}^{\otimes n}$ es unitario (ya que $\langle k_i, k_i\rangle = 1$ para todo $i \in [n]$) y

$$\langle |k_1\rangle \otimes ... \otimes |k_n\rangle \ , \ |k_1\rangle \otimes ... \otimes |k_n\rangle \rangle = \prod_{i\in[n]} \langle k_i, k_i\rangle = 1.$$

Denominamos al conjunto $\{|0\rangle, |1\rangle\}^{\otimes n}$ base computacional de un sistema de n qubits. En general, todo estado $|\psi\rangle$ formado por n qubits se puede representar como combinación lineal compleja de los elementos de su base computacional:

$$|\psi\rangle = \sum_{k\in\{0,1\}^n} \alpha_k |k\rangle, \ \alpha_k \in \mathbb{C}.$$

Por ejemplo, la concatenación de dos qubits en estados no básicos $|\psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ y $|\psi_2\rangle = \alpha_2|0\rangle + \beta_2|1\rangle$ genera el siguiente estado:

$$|\psi_1\rangle \otimes |\psi_2\rangle = \left(\alpha_1|0\rangle + \beta_1|1\rangle\right) \otimes \left(\alpha_2|0\rangle + \beta_2|1\rangle\right) = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle. \tag{2.12}$$

Además, como era de esperar, el estado resultante es unitario:

$$|\alpha_1|^2+|\beta_1|^2 = 1, \quad |\alpha_2|^2+|\beta_2|^2 = 1 \Rightarrow \quad (|\alpha_1|^2+|\beta_1|^2)(|\alpha_2|^2+|\beta_2|^2) = |\alpha_1\alpha_2|^2+|\alpha_1\beta_2|^2+|\beta_1\alpha_2|^2+|\beta_1\beta_2|^2 = 1$$

La base computacional $\{|0\rangle, |1\rangle\}^{\otimes n}$ tiene 2^n elementos, por lo que podemos definir una biyección entre ésta y su correspondencia binaria $\{0,1,...,2^{n-1}\}$. En adelante usaremos esta asociación para referirnos indistintamente a unos u otros elementos según convenga en el contexto. En este caso,

justifica una representación vectorial de los estados de $S(\mathcal{H}_n)$ como vector columna que contiene las amplitudes respecto a cada elemento de la base computacional. Por ejemplo, la representación algebraica del estado $|\psi_1\rangle \otimes |\psi_2\rangle$ anterior sería:

$$|\psi_1\rangle \otimes |\psi_2\rangle = \begin{bmatrix} \alpha_1\alpha_2 \\ \alpha_1\beta_2 \\ \beta_1\alpha_2 \\ \beta_1\beta_2 \end{bmatrix}.$$

Así como normalmente usamos la variable $|\psi\rangle$ para referirnos al estado genérico de un sistema cuántico, cuando en adelante queramos referenciar un elemento de la base computacional habitualmente usaremos la notación $|\chi\rangle$ (o χ para referirnos a su valor binario). En ocasiones será conveniente usar otras nomenclaturas, en este caso se declararán previamente.

2.3.1.1. Entrelazamiento

En la ecuación (2.12) hemos visto un sistema de dos qubits que puede obtenerse como concatenación de qubits aislados

$$|\psi_1\rangle \otimes |\psi_2\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle.$$

Sin embargo, si estudiamos el siguiente ejemplo

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}},$$

que es un estado válido, ya que

$$\left\| \frac{|00\rangle + |11\rangle}{\sqrt{2}} \right\| = \sqrt{\left(\frac{1}{\sqrt{2}}\right)^2 + \left(\frac{1}{\sqrt{2}}\right)^2} = 1,$$

observamos que en este caso las amplitudes de $|01\rangle$ y $|10\rangle$ son nulas. Así pues, revisando los coeficientes de la ecuación (2.12) concluiríamos que α_1 o β_2 toman valor cero, por lo que las amplitudes de $|00\rangle$ y de $|11\rangle$ no podrían ser no nulas simultáneamente.

Esto nos lleva a la conclusión de que $|\psi\rangle$ no puede descomponerse en producto tensorial de estados más simples, lo que implica que existe una relación entre los qubits que lo componen que de alguna forma condiciona mutuamente su estado: la modificación de uno de los qubits implicará afectar al estado del otro, y viceversa. Denominamos a esta propiedad **entrelazamiento** entre qubits y, conjuntamente con la superposición, es responsable de la gran potencia de la computación cuántica y de lo que denominamos paralelismo cuántico.

2.3.2. Operadores sobre múltiples qubits

2.3.2.1. Puertas multiqubit

Una vez más, podemos buscar analogías con la computación clásica, en la cual hay implementadas puertas lógicas que actúan sobre más de un bit, por ejemplo las puertas AND y OR. También en computación cuántica hay operadores que actúan sobre varios qubits.

Antes hemos visto que los operadores sobre un qubit se correspondían con funciones reversibles. Este hecho es consecuencia del segundo postulado de la mecánica cuántica, que también aplica al caso de múltiples qubits. Por lo tanto, también los operadores sobre varios qubits son funciones reversibles.

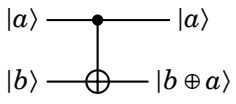
Concretamente, todas las funciones descritas en la Sección 2.1.3 se pueden generar con puertas cuánticas:

Puerta CNOT

La puerta **CNOT** (o controlled-NOT) implementa un operador $U_{CN} : \left(\mathbb{C}^2\right)^{\otimes 2} \to \left(\mathbb{C}^2\right)^{\otimes 2}$ que actúa sobre dos qubits tal como se describía en la Sección 2.1.3: al aplicarse sobre un estado $|a,b\rangle \in (\mathbb{C}^2)^{\otimes 2}$ de la base computacional, $a,b \in \{0,1\}$, tiene el efecto de una puerta X sobre el segundo qubit si el primero tiene estado $|1\rangle$, y en caso contrario aplica el operador identidad. Esto se puede formular como:

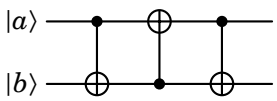
$$U_{CN}\,|a,b\rangle = |a, b \oplus a\rangle\,.$$

En un circuito cuántico esta puerta se representa como:



Puerta swap

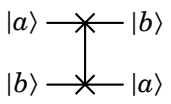
Usando diversas puertas CNOT podríamos emular otros operadores, como el operador **swap** $U_{SW} : \left(\mathbb{C}^2\right)^{\otimes 2} \to \left(\mathbb{C}^2\right)^{\otimes 2}$, que intercambia el estado de dos qubits. Una forma de implementar este operador sería mediante el siguiente circuito, siendo $|a,b\rangle \in (\mathbb{C}^2)^{\otimes 2}$



lo que genera el siguiente cambio sobre el estado inicial:

$$|a,b\rangle \;\to\; |a, a\oplus b\rangle \;\to\; |a\oplus a\oplus b, a\oplus b\rangle = |b, a\oplus b\rangle \;\to\; |b, a\oplus b\oplus b\rangle = |b,a\rangle\,.$$

Esta operación se representa de esta forma en circuitos cuánticos:



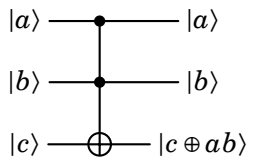
Puerta de Toffoli

La puerta de **Toffoli** implementa un operador $U_T : \left(\mathbb{C}^2\right)^{\otimes 3} \to \left(\mathbb{C}^2\right)^{\otimes 3}$ que actúa sobre tres qubits, tal como se describía en la Sección 2.1.3. Al aplicarse sobre un estado de la base computacional actúa como puerta X sobre el tercer qubit si los dos primeros tienen estado $|11\rangle$, y en caso contrario no aplica ningún cambio.

Su efecto sobre un estado $|a,b,c\rangle \in (\mathbb{C}^2)^{\otimes 3}$ de la base computacional, $a,b,c \in \{0,1\}$, sería el siguiente:

$$U_T\,|a,b,c\rangle = |a,b,c \oplus ab\rangle\,.$$

La representación de esta puerta en un circuito cuántico es



2.3.2.2. Aplicación de múltiples puertas

En esta sección estudiaremos el resultado de aplicar en paralelo diversas puertas de un qubit en un circuito cuántico y las matemáticas involucradas en su modelización. Para ello, completamos la definición de producto tensorial de la sección anterior al caso de operadores:

Definición 2.4. Sean U_1, U_2 dos operadores lineales definidos en V, W respectivamente, el operador lineal $U_1 \otimes U_2$ sobre $v \otimes w \in V \otimes W$ se define como

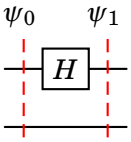
$$(U_1 \otimes U_2)(v \otimes w) = U_1 v \otimes U_2 w$$

Igual que en los casos anteriores, usaremos la notación exponencial $U^{\otimes m}$ para referirnos a la iteración de m productos tensoriales sobre el operador U.

Veamos un ejemplo para ver cómo aplica el producto tensorial entre determinados operadores. Sea un sistema de dos qubits con estado

$$|\psi_0\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \in (\mathbb{C}^2)^{\otimes 2},$$

si aplicamos una puerta de Hadamard sobre el primer qubit

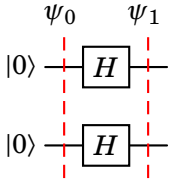


esto equivale a aplicar la puerta $H \otimes I$ sobre $|\psi_0\rangle$, lo que produce el estado:

$$|\psi_1\rangle = (H \otimes I)|\psi_0\rangle = (H \otimes I)\left(\frac{|01\rangle + |10\rangle}{\sqrt{2}}\right) = \frac{1}{\sqrt{2}}((H \otimes I)|01\rangle + (H \otimes I)|10\rangle) = \frac{1}{\sqrt{2}}(|+\rangle \otimes |1\rangle + |-\rangle \otimes |0\rangle)$$

$$= \frac{1}{\sqrt{2}}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |1\rangle + \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes |0\rangle\right) = \frac{|00\rangle + |01\rangle - |10\rangle + |11\rangle}{2}$$

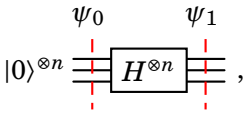
Un recurso especialmente útil en muchos circuitos cuánticos es la aplicación de varias puertas de Hadamard en paralelo sobre un estado de la base computacional. Estudiémoslo en el caso de dos qubits:



El estado inicial del circuito es $|\psi_0\rangle = |00\rangle$. Al aplicar las puertas de Hadamard sobre éste, obtenemos lo siguiente:

$$|\psi_1\rangle = H^{\otimes 2}|00\rangle = (H|0\rangle)^{\otimes 2} = |+\rangle^{\otimes 2} = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes 2} = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} = \sum_{k \in \{0,1\}^2} \frac{|k\rangle}{2}$$

Es decir, hemos generado un nuevo estado superposición de todos los elementos de la base computacional de dos qubits. Supongamos que este resultado se diera también al aplicar puertas de Hadamard sobre un estado de n qubits $|0\rangle^{\otimes n}$



,

y verifiquemos si, en este caso, esto seguiría sucediendo para $n+1$:

$$H^{\otimes n+1}|0\rangle^{\otimes n+1} = \left(H^{\otimes n}|0\rangle^{\otimes n}\right) \otimes (H|0\rangle) = \sum_{k \in \{0,1\}^n} \frac{|k\rangle}{\sqrt{2^n}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \sum_{k \in \{0,1\}^n} \frac{|k\rangle \otimes |0\rangle}{\sqrt{2^{n+1}}} + \sum_{k \in \{0,1\}^n} \frac{|k\rangle \otimes |1\rangle}{\sqrt{2^{n+1}}}$$

$$= \sum_{k \in \{0,1\}^{n+1}} \frac{|k\rangle}{\sqrt{2^{n+1}}}.$$

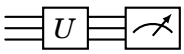
Esto demuestra (por inducción) la validez de la siguiente igualdad, que más adelante será muy útil en el estudio de algunos algoritmos cuánticos:

$$H^{\otimes n}|0\rangle^{\otimes n} = \sum_{k \in \{0,1\}^n} \frac{|k\rangle}{\sqrt{2^n}}. \tag{2.13}$$

2.3.3. Medición sobre múltiples qubits

2.3.3.1. Medición del sistema global

Con frecuencia, al finalizar un algoritmo cuántico realizaremos una medición sobre un número de qubits mayor a 1. Estudiaremos en primer lugar el caso en el que se aplica una medición sobre la totalidad de qubits del sistema:



Los posibles resultados de esta observación serán $\{k\}_{k\in\{0,1\}^n}$, y el operador de medición asociado a cada k es $M_k = |k\rangle\langle k|$. Partiendo del Postulado 3 de la mecánica cuántica, la probabilidad de observar k sobre un estado $|\psi\rangle = \sum_{i\in\{0,1\}^n} \alpha_i |i\rangle$, es

$$P(k) = \langle\psi| M_k^\dagger M_k |\psi\rangle = \left(\sum_{i\in\{0,1\}^n} \overline{\alpha_i}\langle i|\right) M_k^\dagger M_k \left(\sum_{i\in\{0,1\}^n} \alpha_i |i\rangle\right). \tag{2.14}$$

Los términos de los sumatorios anteriores se anularán para todos los índices $i \neq k$ como consecuencia de su producto con $M_k^\dagger M_k$. Esto justifica que la expresión correspondiente a esta probabilidad se pueda simplificar de la siguiente forma:

$$P(k) = \overline{\alpha_k}\langle k| M_k^\dagger M_k \alpha_k |k\rangle = \overline{\alpha_k}\alpha_k = |\alpha_k|^2. \tag{2.15}$$

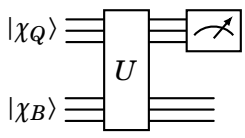
Finalmente, el estado postmedición será

$$\frac{M_k |\psi\rangle}{\sqrt{\langle\psi| M_k^\dagger M_k |\psi\rangle}} = \frac{\alpha_k |k\rangle}{|\alpha_k|}$$

que, tal como ya argumentamos para el caso de 1 qubit, es indistinguible del estado $|k\rangle$.

2.3.3.2. Medición de un subsistema

Veamos ahora el caso en que se aplica el operador medición sobre parte de los qubits del sistema. Supongamos un circuito que consta de dos subsistemas, Q y B, que contienen n y m qubits respectivamente, con espacios de estados $\mathscr{H}_Q \equiv (\mathbb{C}^2)^{\otimes n}$, $\mathscr{H}_B \equiv (\mathbb{C}^2)^{\otimes m}$ y estado inicial $|\chi_Q\rangle \otimes |\chi_B\rangle \in \mathscr{H}_Q \otimes \mathscr{H}_B$. Tras realizar determinadas operaciones unitarias, nos interesará medir el subsistema Q. Representaremos la operación descrita en un diagrama de circuito cuántico de la siguiente forma:



Los posibles resultados de la observación sobre el bloque Q serán $\{q\}_{q\in\{0,1\}^n}$, y los operadores de medición asociados $M_q = |q\rangle\langle q|$, para todo $q \in \{0,1\}^n$.

Para estudiar esta operación desde el punto de vista del Postulado 3, interpretaremos la medición anterior de otra forma. Realizamos una observación sobre todo el sistema, y asociamos la probabilidad de obtener un resultado q (en el supuesto anterior de mediciones sobre Q) a la de obtener un resultado en el sistema global que esté contenido en el conjunto $\{q\}\times\{0,1\}^m$. Los operadores de medición correspondientes a cada q sobre este sistema global son $\tilde{M}_q = |q\rangle\langle q| \otimes I_B$, siendo I_B la matriz identidad de orden 2^m [32, p.107].

Aplicando los resultados del Postulado 3 a este modelo, desarrollamos la probabilidad de observar q en el subsistema Q tras la medición del estado global $|\psi\rangle = \sum_{i\in\{0,1\}^n}\sum_{j\in\{0,1\}^m}\alpha_{ij}|i,j\rangle$, como

$$\begin{aligned}
P(q) &= \langle\psi|\tilde{M}_q^\dagger\tilde{M}_q|\psi\rangle \\
&= \left(\sum_{i\in\{0,1\}^n}\sum_{j\in\{0,1\}^m}\overline{\alpha_{ij}}\langle i,j|\right)\left(M_q\otimes I_B\right)^\dagger\left(M_q\otimes I_B\right)\left(\sum_{i\in\{0,1\}^n}\sum_{j\in\{0,1\}^m}\alpha_{ij}|i,j\rangle\right) \\
&= \left(\sum_{i\in\{0,1\}^n}\sum_{j\in\{0,1\}^m}\overline{\alpha_{ij}}\langle i,j|\right)\left(\sum_{i\in\{0,1\}^n}\sum_{j\in\{0,1\}^m}\left(M_q\otimes I_B\right)^\dagger\left(M_q\otimes I_B\right)\alpha_{ij}|i,j\rangle\right) \\
&= \left(\sum_{i\in\{0,1\}^n}\sum_{j\in\{0,1\}^m}\overline{\alpha_{ij}}\langle i,j|\right)\left(\sum_{j\in\{0,1\}^m}\alpha_{qj}|q,j\rangle\right)
\end{aligned}$$

Por la ortonormalidad de los estados de la base computacional, al multiplicar cada uno de los términos de los sumatorios anteriores obtendremos $\langle i,j|q,j'\rangle = 1$ cuando $i = q$ y $j = j'$, o $\langle i,j|q,j'\rangle = 0$ en caso contrario. Esto justifica que podamos reescribir la expresión anterior como:

$$P(q) = \sum_{j\in\{0,1\}^m}\overline{\alpha_{qj}}\alpha_{qj} = \sum_{j\in\{0,1\}^m}|\alpha_{qj}|^2, \tag{2.16}$$

donde α_{qj} es la amplitud de $|\psi\rangle$ respecto al estado de la base computacional $|q,j\rangle$.

CAPÍTULO 3

ALGORITMOS DE CAJA NEGRA

3.1. Conceptos

3.1.1. El oráculo

Muchos de los algoritmos conocidos en computación cuántica se basan en el modelo de queries descrito en la Sección 2.1.5. Un elemento central en este modelo es el **oráculo**, un operador que se corresponde con una función $f : \{0,1\}^n \to \{0,1\}^m$ representable mediante un vector $X \in \{0,1\}^N$, donde $N = m2^n$. Con frecuencia, el oráculo se define con un único bit de salida ($m = 1$) [1], aunque en este trabajo consideraremos el caso general por corresponderse con algunos de los algoritmos cuánticos más conocidos.

Un operador cuántico es unitario e invertible, pero la función f no necesariamente lo es. Sin embargo, tal como ya vimos en (2.3), la función f puede implementarse como función invertible $\tilde{f}$ de la siguiente forma:

$$\tilde{f} : \{0,1\}^{n+m} \to \{0,1\}^{n+m}, \quad \tilde{f}(x,y) = (x, y \oplus f(x))$$

A partir de esta idea, veamos la implementación cuántica del oráculo que evalúa f. Su estructura de información está formada por dos sistemas Q,B, que contienen n y m qubits respectivamente. Denotamos los espacios de estados correspondientes a estos sistemas como $\mathcal{H}_Q \equiv \left(C^2\right)^{\otimes n}$, $\mathcal{H}_B \equiv \left(C^2\right)^{\otimes m}$. Sea $|\chi_Q\rangle \otimes |\chi_B\rangle \in \mathcal{H}_Q \otimes \mathcal{H}_B$ un elemento de la base computacional del sistema global, donde $\chi_Q \in \{0,1\}^n, \chi_B \in \{0,1\}^m$, el operador correspondiente al oráculo se definirá de forma que toma la información $|\chi_Q\rangle$ del sistema Q como datos de entrada y devuelve la

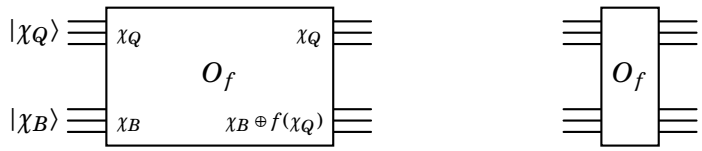
[1] De hecho, toda la bibliografía y referencias consultadas asocian el oráculo a una función $f : \{0,1\}^n \to \{0,1\}$. Tal como demostraremos en este trabajo, los resultados obtenidos en ese supuesto son generalizables al caso $m > 1$, pero en algunos casos este hecho no es evidente o implica algunas limitaciones, por lo que ha sido necesario detallar definiciones, teoremas y demostraciones adaptadas para este caso general.

evaluación $|\chi_B \oplus f(\chi_Q)\rangle$ en la salida del sistema B. Más concretamente, definimos este operador como la aplicación $O_f : \mathscr{H}_Q \otimes \mathscr{H}_B \to \mathscr{H}_Q \otimes \mathscr{H}_B$ tal que transforma los elementos $|\chi_Q\rangle \otimes |\chi_B\rangle$ de la base computacional de la siguiente forma:

$$O_f(|\chi_Q\rangle \otimes |\chi_B\rangle) = |\chi_Q\rangle \otimes |\chi_B \oplus f(\chi_Q)\rangle\,.$$

Observamos que O_f transforma la base computacional de $\mathscr{H}_Q \otimes \mathscr{H}_B$ en si misma, ya que los qubits de $|\chi_Q\rangle$ permanecen inalterados y la operación $\chi_B \oplus f(\chi_Q)$ preserva o invierte el valor de los qubits de χ_B en función del valor de $f(\chi_Q)$. En consecuencia, O_f es un operador unitario.

El oráculo se representa de la siguiente forma en un circuito cuántico (según el contexto usaremos un diagrama detallado, como el de la izquierda, o la representación simplificada de la derecha):



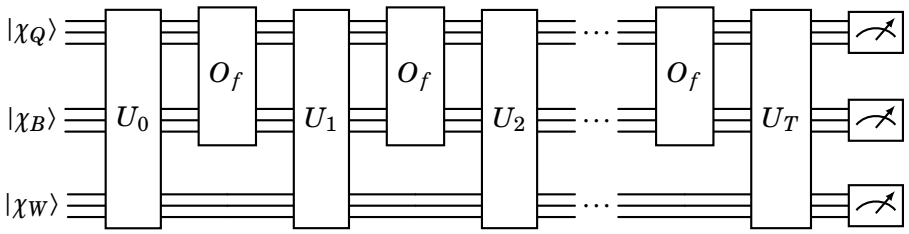
3.1.2. Algoritmos de caja negra

Detallemos la estructura habitual de un **algoritmo de caja negra de T queries**. El espacio de estados del algoritmo está formado por tres sistemas Q, B, W: los dos primeros tienen la estructura descrita antes, y el tercero contiene estados de l qubits, $\mathscr{H}_W \equiv \left(\mathbb{C}^2\right)^{\otimes l}$.

El algoritmo aplica una sucesión de puertas unitarias $U_0, O_f, U_1, O_f, ..., O_f, U_T$, donde O_f es el operador correspondiente al oráculo y U_j son operadores unitarios, con $j = 0, ..., T$. Sea $|\chi_Q\rangle \otimes |\chi_B\rangle \otimes |\chi_W\rangle \in \mathscr{H}_Q \otimes \mathscr{H}_B \otimes \mathscr{H}_W$ un elemento de la base computacional del sistema, las puertas O_f actúan sobre él de la siguiente forma

$$O_f(|\chi_Q\rangle \otimes |\chi_B\rangle \otimes |\chi_W\rangle) = |\chi_Q\rangle \otimes |\chi_B \oplus f(\chi_Q)\rangle \otimes |\chi_W\rangle \tag{3.1}$$

En el siguiente diagrama se representa la estructura del circuito global:



El último paso del algoritmo consiste en ejecutar la medición del sistema y obtener un resultado. La medición de un estado involucra un proceso aleatorio, así que el resultado del algoritmo es no determinista. En este punto aplicaremos lo descrito en la Sección 2.3.3.2: asociaremos la

aceptación del algoritmo al hecho de que su output esté contenido en un conjunto $D \subset \{0,1\}^{n+m+l}$ de elementos que consideramos válidos.

Siguiendo lo descrito en la ecuación (2.4), el problema asociado al algoritmo anterior puede formalizarse mediante la siguiente función

$$F : S \subset \{0,1\}^N \to \{0,1\},$$

donde $N = m2^n$.

Sintetizando lo anterior, denotaremos **aceptación de un algoritmo** de caja negra a la validación de que su resultado está en un conjunto arbitrario $D \subset \{0,1\}^{n+m+l}$ (que denominamos conjunto de aceptación), y **probabilidad de aceptación** del algoritmo para un input $X \in \{0,1\}^N$ a la probabilidad de que éste genere un output contenido en D.

3.1.3. Oráculo de fase

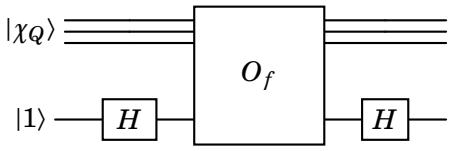
Un caso particular de aplicación del modelo de oráculo se produce cuando el sistema B tiene un sólo bit ($m = 1$) y su estado se inicializa con $|\chi_B\rangle = |-\rangle$. En este caso, si $f(\chi_Q) = 0$ el estado se mantiene inalterado, y si $f(\chi_Q) = 1$ el efecto sobre el estado es el siguiente:

$$O_f\left(|\chi_Q\rangle \otimes |-\rangle\right) = O_f\left(|\chi_Q\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = |\chi_Q\rangle \otimes \frac{|0\oplus 1\rangle - |1\oplus 1\rangle}{\sqrt{2}} = |\chi_Q\rangle \otimes \frac{|1\rangle - |0\rangle}{\sqrt{2}} = -|\chi_Q\rangle \otimes |-\rangle$$

Es decir, en cualquier caso B conserva el estado tras la ejecución del oráculo, pero en función del resultado de $f(\chi_Q)$ se modifica la fase del sistema de la siguiente forma:

$$O_f(|\chi_Q\rangle \otimes |-\rangle) = (-1)^{f(\chi_Q)}|\chi_Q\rangle \otimes |-\rangle \tag{3.2}$$

El estado inicial $|-\rangle$ puede generarse aplicando una puerta de Hadamard sobre un qubit en estado $|1\rangle$, así que una posible implementación de lo descrito podría ser la siguiente:



Más adelante veremos algún ejemplo de esta aplicación del oráculo.

3.1.4. Complejidad

En esta sección estudiaremos cómo comparar la complejidad de circuitos cuánticos de caja negra con la de sus circuitos clásicos equivalentes. Para ello usaremos los conceptos de complejidad descritos en la Definición 2.1 (circuitos clásicos) y en la Definición 2.2 (circuitos cuánticos de caja negra).

Sea $(F_n)_{n\in\mathbb{N}}$ una familia de funciones correspondiente a un problema, y sean $(C_n)_{n\in\mathbb{N}}$, $(Q_n)_{n\in\mathbb{N}}$ las familias de circuitos clásicos y cuánticos asociados con complejidades $c(n)$ y $q(n)$, donde $c, q : \mathbb{N} \to \mathbb{R}^+$, decimos que:

- El circuito clásico tiene un **sobrecoste polinomial** para este problema si

$$\frac{c(n)}{q(n)} \leq \text{poly}(n), \forall n \in \mathbb{N}.$$

- El circuito clásico tiene un **sobrecoste superpolinomial** en caso contrario.

Por supuesto, el objetivo será encontrar algoritmos cuánticos que reduzcan la complejidad en un orden superpolinomial respecto al mejor algoritmo clásico asociado. Es decir, problemas no resolubles a la práctica en ordenadores clásicos y que sí lo serían con tecnología cuántica. Esta situación, que se conoce habitualmente como **supremacía cuántica**, es la que da sentido a todo el esfuerzo por desarrollar la teoría y los dispositivos de computación cuántica.

Un ejemplo de algoritmo cuántico más eficiente que el equivalente clásico es el algoritmo de Shor, que resuelve la factorización de un número de n bits con una complejidad $O(\log^3 n)$, mientras que el mejor algoritmo clásico conocido requiere un número de operaciones muy superior, $O(\exp(1{,}9(\log n)^{1/3}(\log\log n)^{2/3}))$ [40, p. 15]. De esta relación de complejidades concluimos que la mejora cuántica en este caso es superpolinomial.

Otro ejemplo conocido de algoritmo cuántico con mejor rendimiento que el clásico es el algoritmo de Grover, que describe cómo localizar un elemento que cumpla una propiedad arbitraria dentro de una secuencia no ordenada de n datos. En este caso la complejidad de un algoritmo clásico sería $O(n)$ (ya que requiere ir revisando secuencialmente todos los elementos, hasta encontrar la coincidencia), mientras que Grover detalla la forma de encontrarlo en $O(n^{1/2})$ en un computador cuántico [32, p.248]. Además, se ha podido demostrar que no es posible resolver el problema cuánticamente con un orden menor de operaciones, así que la mejora cuántica es cuadrática, y por lo tanto de orden polinomial.

Ante estos resultados, surge una pregunta inevitable: ¿En qué tipos de problemas será posible diseñar algoritmos cuánticos superpolinomialmente más eficientes que sus equivalentes clásicos?

3.2. Conjetura sobre aceleración cuántica

La siguiente conjetura (a la que en adelante nos referiremos como *Conjetura Cuántica*) propone una respuesta a la pregunta anterior:

Conjetura 3.1. *(Conjetura Cuántica). Sea Q un algoritmo cuántico que invoca T queries sobre un input booleano $X \in \{0,1\}^N$, y sean $\varepsilon, \delta \in (0,1]$, entonces existe un algoritmo clásico determinista C que invoca* $\text{poly}(T, 1/\varepsilon, 1/\delta)$ *queries sobre X, y que aproxima la probabilidad de aceptación de Q en un error aditivo máximo ε sobre una fracción $1-\delta$ de los inputs.*[*4*, *pp*, 136 – 138][*19*, *p*, 2]

La probabilidad de aceptación referenciada en la conjetura es un concepto importante desde el punto de vista de la comprensión del resultado del algoritmo Q, pero el objetivo último de la computación es la obtención de este resultado. Por esta razón, nos interesará enunciar una conjetura alternativa que describa un algoritmo clásico cuyo resultado no sea la probabilidad de aceptación de Q sino la computación de su output.

Conjetura 3.2. *(Computabilidad) Sean $\alpha, c \in (0,1]$, y sea Q un algoritmo cuántico que computa, invocando T queries, una función $F : S \subseteq \{0,1\}^N \to \{0,1\}$ tal que $|S| \geq c2^N$, entonces existe un algoritmo clásico determinista que realiza* $\text{poly}(T, 1/\alpha, 1/c)$ *queries y que computa $F(X)$ en una fracción por lo menos $1-\alpha$ de inputs $X \in S$ [4, pp.161].*

Teorema 3.1. *La Conjetura 3.1 (Conjetura Cuántica) implica la Conjetura 3.2 (Computabilidad) para algoritmos de caja negra [4, p. 161].*

Demostración. Por definición de computabilidad (ver Sección 2.1.6), Q evalúa F con probabilidad de error inferior o igual a 1/3 usando T queries. Además, si $p(X)$ es la probabilidad de aceptación calculada por este algoritmo para cada input $X \in \{0,1\}^N$, por la Conjetura 3.1 existe un algoritmo clásico C que aproxima $p(X)$ con un error aditivo ε en una fracción $1-\delta$ de los inputs $\{0,1\}^N$ ejecutando $\text{poly}(T, 1/\varepsilon, 1/\delta)$ queries.

Si denotamos $\widetilde{p}(X)$ a la probabilidad de aceptación calculada por el algoritmo C y $S_0 \subset \{0,1\}^N$ el conjunto de inputs correspondiente a la fracción $1-\delta$ anterior, podemos sintetizar lo descrito en las dos desigualdades siguientes:

$$|F(X) - p(X)| \leq \frac{1}{3} \qquad \text{para todo } X \in \{0,1\}^N$$
$$|p(X) - \widetilde{p}(X)| \leq \varepsilon \qquad \text{para todo } X \in S_0 \subset \{0,1\}^N$$

Tomaremos $\varepsilon := 1/7, \delta := \alpha c$. Combinando las dos ecuaciones anteriores, en el caso de que para un $X \in S_0$, $F(X) = 1$, se cumplirá que

$$\widetilde{p}(X) \geq p(X) - \varepsilon \geq \left|1 - \frac{1}{3}\right| - \frac{1}{7} = \frac{2}{3} - \frac{1}{7} = \frac{11}{21} > \frac{1}{2}$$

Y en el supuesto de que $X \in S_0$, $F(X) = 0$ obtenemos

$$\widetilde{p}(X) \leq p(X) + \varepsilon \leq \left|0 - \frac{1}{3}\right| + \frac{1}{7} = \frac{1}{3} + \frac{1}{7} = \frac{10}{21} < \frac{1}{2}$$

Concluyendo que el resultado del algoritmo C permite discernir el valor de $F(X)$ para todo $X \in S_0$: si $\widetilde{p}(X) < 1/2$ tomaremos evaluación $F(X) = 0$; y si $\widetilde{p}(X) > 1/2$ tomaremos evaluación $F(X) = 1$). Y como $S_0/2^N \geq 1-\delta$, esto sucederá en por lo menos una fracción $1-\delta = 1-\alpha c \geq 1-\alpha$.

Finalmente, aplicando los criterios de la Conjetura 3.1 concluimos que el número de queries que requerirá el algoritmo clásico será $\text{poly}(T, 1/\alpha c, 1/7) = \text{poly}(T, 1/\alpha, 1/c)$. □

En [4, p. 161] se demuestra también la implicación recíproca del teorema anterior, de forma que las Conjeturas 3.1 y 3.2 son equivalentes. Sin embargo, como el resultado que determina la computabilidad si aceptamos la Conjetura Cuántica es suficiente para nuestro propósito, no hemos incluido este otro sentido de la implicación en nuestro teorema ni en su demostración.

3.2.1. Interpretación de la Conjetura Cuántica

Veamos cómo interpretar la Conjetura 3.1 desde el punto de vista de la caracterización de un problema en el que se consigue una situación de supremacía cuántica.

Supondremos primero un problema sobre **inputs totales** cuyo algoritmo cuántico es superpolinomialmente más eficiente que su mejor algoritmo clásico asociado. Es decir, disponemos de una familia de circuitos cuánticos $(Q_n)_{n\in\mathbb{N}}$ y otra de clásicos $(C_n)_{n\in\mathbb{N}}$ de complejidades $q(n)$ y $c(n)$ respectivamente, tales que cada par Q_n, C_n genera el mismo output en todos los inputs de $\{0,1\}^n$. Representamos el supuesto de supremacía cuántica en la siguiente relación:

$$c(n) > \text{poly}(q(n)) \tag{3.3}$$

Recordemos que en la Conjetura Cuántica intervienen dos parámetros ε, δ (el algoritmo clásico aproxima la probabilidad de aceptación con un error máximo ε en una fracción $1-\delta$ de los inputs). En la demostración del Teorema 3.1 asignábamos un valor $\varepsilon := 1/7$, y demostrábamos que este error máximo era suficientemente bajo para asegurar que ambos algoritmos generaran el mismo output en la fracción de inputs considerada. Fijamos también ahora este valor para ε, pero asumiremos que la fracción $1-\delta$ puede ser distinta para los distintos valores de n (por ello, denotaremos $\delta(n)$ al valor de δ para cada $n \in \mathbb{N}$).

Con esta notación, la conjetura indica que

$$c(n) = \text{poly}\left(q(n), \frac{1}{\varepsilon}, \frac{1}{\delta(n)}\right) = \text{poly}\left(q(n), \frac{1}{7}, \frac{1}{\delta(n)}\right) = \text{poly}\left(q(n), \frac{1}{\delta(n)}\right), \tag{3.4}$$

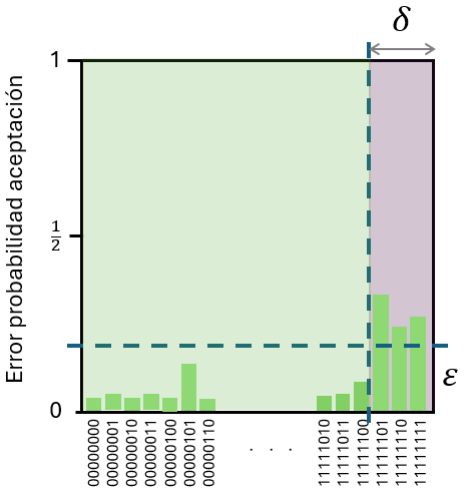
y de las ecuaciones (3.3) y (3.4) deducimos que

$$\frac{1}{\delta(n)} > \text{poly}(n) \tag{3.5}$$

Es importante remarcar que las complejidades $q(n), c(n)$ representan el número máximo de queries necesarias para procesar cualquiera de sus inputs, pero que no todos ellos requerirán este máximo de queries. Por lo tanto, podría haber inputs para los que la relación entre el número de queries requerido por los algoritmos cuántico y clásico sea distinta a la de las complejidades de estos algoritmos. Concretamente, la Conjetura Cuántica afirma que todos los inputs contenidos en la fracción $1-\delta(n)$ "escaparán" a la restricción descrita en (3.3), puesto que su relación con $q(n)$ es polinomial.

Veamos este hecho en el siguiente diagrama: en la parte izquierda representamos (en verde claro) el conjunto de datos que el algoritmo clásico computa en un número de queries polinomial

respecto al cuántico, y en la parte derecha (morado oscuro) el conjunto de inputs que podrían requerir más queries, y que por lo tanto podrían ser responsables de la supremacía cuántica [2].



La ecuación (3.5) indica que en el supuesto de supremacía cuántica la fracción $\delta(n)$ de inputs candidatos a requerir tal incremento de queries en el algoritmo clásico (conjunto morado) debería ser extremadamente reducida, y que seguirá reduciéndose al incrementar n de forma que el inverso de esta fracción sea superior a cualquier polinomio de n. Así pues, se trataría de problemas con una estructura tal que sólo una fracción extraordinariamente pequeña de inputs podrían justificar la ventaja cuántica.

Más allá de esta interpretación, en [10, p. 791] y [4, p. 135] se referencia un resultado más restrictivo que el descrito: el sobrecoste clásico respecto al cuántico para problemas sobre inputs totales es, como mucho, del orden de la sexta raíz: $c(n) = O(q(n)^6)$. No se conoce ningún problema en el que se obtenga esta mejora (de momento sólo el algoritmo de Grover ha conseguido una mejora cuadrática) [4, p. 135], así que existe la posibilidad de que esta cota consiga reducirse todavía más. Pero en cualquier caso, existe una relación polinomial entre ambas complejidades, por lo que quedan descartados escenarios de supremacía cuántica en problemas sobre inputs totales.

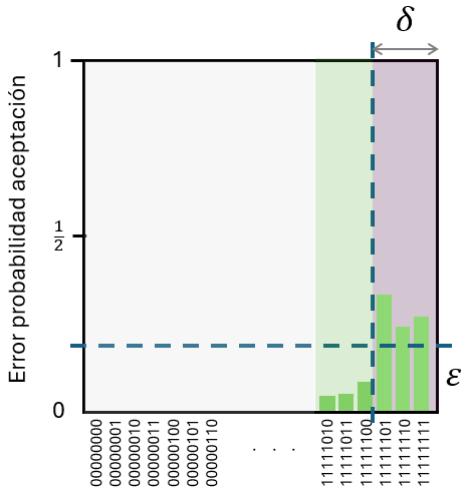
Para problemas sobre **inputs parciales** no existen resultados tan concluyentes. Se conocen algoritmos cuánticos con complejidades superpolinomialmente más reducidas que el mejor algoritmo clásico asociado (por ejemplo, los algoritmos de Deutsch-Jozsa y el de Simon que estudiaremos en las próximas secciones), pero no se ha podido demostrar de forma más precisa cuál es el límite en estas posibles mejoras cuánticas. Sin embargo, sí podremos hacer una interpretación similar a la anterior sobre el comportamiento del algoritmo para las distintas fracciones de inputs.

[2] Para simplificar la representación, hemos asumido en el diagrama que los inputs correspondientes a los números binarios más altos están en la fracción δ y los binarios más bajos en la fracción $1-\delta$. Por supuesto, esto no tiene por qué ser así, pero mediante una simple reordenación de los inputs el diagrama seguiría siendo válido.

Para ello, recordemos primero la representación de la evaluación de un algoritmo de queries que describíamos en (2.4):

$$F : S \subseteq \{0,1\}^n \to \{0,1\}^m \tag{3.6}$$

En un problema sobre inputs parciales, el conjunto S del dominio es distinto a $\{0,1\}^n$, así que habrá que tener en cuenta que algunos elementos de $\{0,1\}^n$ no son inputs del problema, con lo que no tiene sentido el cálculo de su probabilidad de aceptación. Por lo tanto, tendremos tres tipologías de datos: (i) los que no son inputs del problema (en gris claro en el siguiente diagrama), (ii) los inputs del problema incluidos en la fracción $1-\delta$ (en verde), para los que podemos asegurar relación polinomial en el número de queries necesarias para computarlos en su algoritmo cuántico vs. el clásico, y (iii) los inputs del problema que podrían escapar a esta relación polinomial (en morado), y que por lo tanto son candidatos a ser responsables de la supremacía cuántica.



Actualizando la interpretación anterior, en este caso el hecho de que la fracción $\delta(n)$ sea tan reducida puede ser debido a dos razones: (a) que la fracción de datos del problema $|S|/2^n$ es muy reducida, (b) que el número de inputs del problema responsables de la ventaja cuántica es muy restringido (la misma explicación que en el caso de problemas sobre inputs globales).

Aaronson y Ambainis [4, p. 135] plantean la posibilidad (a) anterior, indicando que la supremacía cuántica debería producirse en problemas con funciones f altamente estructuradas, cuyo dominio incluya sólo inputs que satisfacen una determinada promesa muy restringida. A esta idea habría que añadir la posibilidad teórica que hemos descrito en la posibilidad (b). En ambos supuestos se reafirma la idea de funciones f con una estructura muy definida: (i) bien sea por corresponderse con un conjunto de inputs muy reducido, o (ii) por la especificidad que permite que el algoritmo sólo obtenga la ventaja cuántica en un conjunto de inputs residual. Pero en general, cuando interpretemos la conjetura en problemas concretos, priorizaremos la revisión de la posibilidad (a), que además de ser la identificada en [4], es la más natural.

3.3. Ejemplos

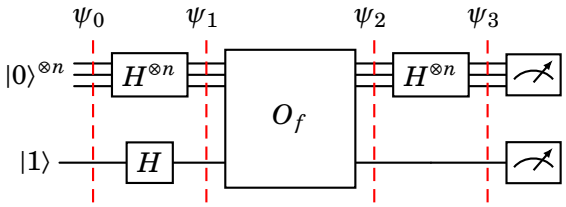
A continuación veremos un par de ejemplos de algoritmos de computación cuántica. En cada uno de ellos, además de exponer el problema y explicar el funcionamiento del algoritmo cuántico que lo resuelve, estudiaremos su complejidad y la interpretaremos en clave de la Conjetura Cuántica.

3.3.1. Algoritmo de Deutsch-Jozsa

Decimos que una función $f : \{0,1\}^n \to \{0,1\}$, es *constante* si $f(x) = C$ para todo $x \in \{0,1\}^n$ y algún $C \in \{0,1\}$. Por otro lado, f es *balanceada* si $|f^{-1}(1)| = |f^{-1}(0)| = 2^{n-1}$. Es decir, si la mitad de los elementos de $\{0,1\}^n$ tienen imagen 0 y la otra mitad 1. Estudiemos el problema que consiste en, dada una función f como la anterior de la que sabemos que es *constante* o *balanceada*, discernir de cuál de los dos tipos es [17].

En computación clásica este problema requiere ir evaluando la imagen de f para cada elemento de $\{0,1\}^n$ hasta encontrar un valor distinto a los previos (en este caso no es *constante*, luego es *balanceada*), o bien haber encontrado que la mitad más 1 de las imágenes son iguales (en este caso es *constante*, pues más de la mitad de los elementos tienen el mismo valor, con lo que no es *balanceada*). Por lo tanto, en el peor de los casos tendremos que evaluar $2^{n-1}+1$ veces la función, así que se trata de un algoritmo de complejidad exponencial $O(2^n)$.

A continuación describimos el algoritmo Deutsch-Jozsa, que resuelve este mismo problema en computación cuántica de una forma mucho más eficiente [32, pp. 34-36]. La información del algoritmo consta de dos sistemas Q, B, que contienen n y 1 qubits respectivamente, con espacios de estados $\mathscr{H}_Q \equiv \left(\mathbb{C}^2\right)^{\otimes n}$, $\mathscr{H}_B \equiv \mathbb{C}^2$. A continuación representamos el circuito que lo implementa:



Tal como indica el diagrama, el estado inicial es:

$$|\psi_0\rangle = |0\rangle^{\otimes n} \otimes |1\rangle .$$

Veamos cómo evoluciona. Al aplicar las puertas de Hadamard iniciales obtenemos:

$$|\psi_1\rangle = \left(H^{\otimes n+1}\right)|\psi_0\rangle = (H|0\rangle)^{\otimes n} \otimes (H|1\rangle) = |+\rangle^{\otimes n} \otimes |-\rangle .$$

Tal como vimos en (2.13), el sistema Q puede expresarse como superposición de su base computacional, así que reescribimos la ecuación anterior como:

$$|\psi_1\rangle = \sum_{k\in\{0,1\}^n} \frac{|k\rangle}{\sqrt{2^n}} \otimes |-\rangle .$$

Aplicamos el oráculo sobre este estado:

$$|\psi_2\rangle = O_f\,|\psi_1\rangle = O_f\left(\sum_{k\in\{0,1\}^n}\frac{|k\rangle}{\sqrt{2^n}}\otimes|-\rangle\right).$$

Como se trata de un oráculo de fase como el representado en la Sección 3.1.3, aplicando la ecuación (3.2) obtenemos

$$|\psi_2\rangle = \sum_{k\in\{0,1\}^n}\frac{(-1)^{f(k)}|k\rangle}{\sqrt{2^n}}\otimes|-\rangle\,.$$

Y el último estado previo a la medición es

$$|\psi_3\rangle = \left(H^{\otimes n}\otimes I\right)|\psi_2\rangle = H^{\otimes n}\left(\sum_{k\in\{0,1\}^n}\frac{(-1)^{f(k)}|k\rangle}{\sqrt{2^n}}\right)\otimes|-\rangle\,. \tag{3.7}$$

Veamos cuál es el efecto de aplicar puertas de Hadamard sobre el sistema Q. En primer lugar, observamos que para el caso particular en que $n=1$, si $k=0$, $H|k\rangle = |+\rangle = (|0\rangle+|1\rangle)/\sqrt{2}$; y si $k=1$, $H|k\rangle = |-\rangle = (|0\rangle-|1\rangle)/\sqrt{2}$. Por lo tanto, podemos escribir la acción de las puertas de Hadamard sobre un elemento de la base computacional como

$$H|k\rangle = \sum_{z\in\{0,1\}}\frac{(-1)^{kz}|z\rangle}{\sqrt{2}}.$$

En el caso general de n qubits ($k\in\{0,1\}^n$) el efecto anterior se aplica sobre cada uno de ellos. Es decir, el sumatorio iterará elementos $z\in\{0,1\}^n$, y cada uno de los qubits (de índice j) aportará $(-1)^{k_j z_j}$ al producto general (donde k_j y z_j son el j-ésimo bit de k y z respectivamente). Como la expresión del producto interno en $\{0,1\}^n$ es $kz = \sum_{j=1}^n k_j z_j$, expresamos el estado resultante de aplicar las n puertas de Hadamard como

$$H^{\otimes n}|k\rangle = \sum_{z\in\{0,1\}^n}\frac{(-1)^{kz}|z\rangle}{\sqrt{2^n}}. \tag{3.8}$$

Aplicando (3.8) y la linealidad de la puerta de Hadamard sobre el primer bloque de qubits, el desarrollo de la ecuación (3.7) quedaría:

$$\begin{aligned}
|\psi_3\rangle &= H^{\otimes n}\left(\sum_{k\in\{0,1\}^n}\frac{(-1)^{f(k)}|k\rangle}{\sqrt{2^n}}\right)\otimes|-\rangle \\
&= \left(\sum_{z\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}\sum_{k\in\{0,1\}^n}\frac{(-1)^{kz}(-1)^{f(k)}|z\rangle}{\sqrt{2^n}}\right)\otimes|-\rangle && \text{por (3.8)} \\
&= \left(\sum_{z\in\{0,1\}^n}\sum_{k\in\{0,1\}^n}\frac{(-1)^{kz+f(k)}|z\rangle}{2^n}\right)\otimes\frac{|0\rangle-|1\rangle}{\sqrt{2}} && \text{reordenando términos}
\end{aligned} \tag{3.9}$$

Al final del circuito realizamos la medición del sistema. Veamos cuál es la probabilidad de que el resultado de la medición en el sistema Q sea 0^n [3] . Para ello aplicamos la ecuación (2.16):

$$P(0^n) = |\alpha_{0^n 0}|^2 + |\alpha_{0^n 1}|^2,$$

[3] En este caso, 0^n no representa la operación aritmética de potencia del número 0 (que evidentemente tendría valor nulo) sino el único elemento del conjunto $\{0\}^n$. Es decir: la concatenación de n ceros. Seguiremos usando esta notación en lo sucesivo.

donde $\alpha_{0^n0}, \alpha_{0^n1}$ representan las amplitudes de $|\psi_3\rangle$ respecto a los estados de la base computacional $|0^n,0\rangle, |0^n,1\rangle$, respectivamente. Aplicando las amplitudes que se derivan de la ecuación (3.9), obtenemos

$$\begin{aligned} P(0^n) &= \left| \sum_{k\in\{0,1\}^n} \frac{(-1)^{k0+f(k)}}{2^n\sqrt{2}} \right|^2 + \left| \sum_{k\in\{0,1\}^n} -\frac{(-1)^{k0+f(k)}}{2^n\sqrt{2}} \right|^2 \\ &= \frac{1}{2}\left| \sum_{k\in\{0,1\}^n} \frac{(-1)^{f(k)}}{2^n} \right|^2 + \frac{1}{2}\left| -\sum_{k\in\{0,1\}^n} \frac{(-1)^{f(k)}}{2^n} \right|^2 \\ &= \left| \sum_{k\in\{0,1\}^n} \frac{(-1)^{f(k)}}{2^n} \right|^2 \end{aligned}$$

Estudiaremos esta probabilidad diferenciando los casos en que f es *constante* respecto a f *balanceada*.

- Supongamos f *constante*, $f(x) = C$. Entonces,

$$P(0^n \mid f \text{ constante}) = \left| \sum_{k\in\{0,1\}^n} \frac{(-1)^{f(k)}}{2^n} \right|^2 = \left| \sum_{k\in\{0,1\}^n} \frac{(-1)^{C}}{2^n} \right|^2 = \left| 2^n \frac{(-1)^{C}}{2^n} \right|^2 = 1.$$

- Supongamos ahora que f es *balanceada*. En este caso, el sumatorio de la probabilidad anterior tomará valor $+1$ en la mitad de sus términos y -1 en la otra mitad, con lo cual se anulará,

$$P(0^n \mid f \text{ balanceada}) = \left| \sum_{k\in\{0,1\}^n} \frac{(-1)^{f(k)}}{2^n} \right|^2 = \left| \frac{+1+1\ldots+1-1-1\ldots-1}{2^n} \right|^2 = 0.$$

De forma que, tras aplicar la medición final del algoritmo, podemos concluir que si el resultado del sistema Q es 0^n, la función es *constante*, y en caso contrario es *balanceada*.

Veamos algunas conclusiones sobre este algoritmo desde el punto de vista de lo estudiado hasta ahora. En primer lugar, observamos que se trata de un algoritmo que usa el oráculo de fase. Realiza una sola iteración sobre el oráculo, así que es de complejidad de queries $O(1)$. Como el algoritmo clásico equivalente es $O(2^n)$, tenemos un caso de sobrecoste clásico superpolinomial.

Por lo tanto, según las conclusiones de la interpretación de la Conjetura 3.1, sería de esperar que los datos de entrada del problema fueran muy reducidos respecto al total de posibles datos. Veamos si es así, contabilizando qué proporción de entre todas las posibles funciones $f : \{0,1\}^n \to \{0,1\}$ son *constantes* o *balanceadas*.

Si denotamos $N = 2^n$, hay en total 2^N funciones $f : \{0,1\}^n \to \{0,1\}$ distintas, de las cuales 2 son *constantes* y $\binom{N}{N/2}$ *balanceadas* (se corresponde con la selección de los $N/2$ elementos cuya imagen

es 0). Si denominamos $\phi(n)$ a la proporción de funciones *constantes* o *balanceadas* respecto al total, obtenemos.

$$\phi(n) = \frac{2 + \binom{2^n}{2^{n-1}}}{2^{2^n}} = 2^{1-2^n} + \frac{(2^n)!}{\left((2^{n-1})!\right)^2 2^{2^n}}$$

En este punto aplicamos la fórmula de Stirling, según la cual las siguientes expresiones convergen asintóticamente para valores elevados de n

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n, \tag{3.10}$$

de forma que:

$$\phi(n) \sim 2^{1-2^n} + \frac{\sqrt{2\pi 2^n}\left(\frac{2^n}{e}\right)^{2^n}}{\left(\sqrt{2\pi 2^{n-1}}\left(\frac{2^{n-1}}{e}\right)^{2^{n-1}}\right)^2 2^{2^n}} = 2^{1-2^n} + \frac{\sqrt{\pi}\ 2^{\frac{n+1}{2}}\ 2^{n2^n}\ e^{2^n}}{\pi\ 2^n\ 2^{2(n-1)2^{n-1}}\ e^{2^n}\ 2^{2^n}} = 2^{1-2^n} + \frac{1}{\sqrt{\pi}} 2^{\frac{1-n}{2}} \sim \frac{1}{\sqrt{\pi}} 2^{\frac{1-n}{2}}.$$

Así que

$$\phi(n) = O(2^n).$$

Es decir, esta razón $\phi(n)$ decrece asintóticamente como una exponencial, cosa que es coherente con la interpretación de la conjetura de la sección anterior.

3.3.2. Algoritmo de Simon

Una función $f : \{0,1\}^n \to \{0,1\}^n$ es invariante bajo una máscara XOR si existe $s \in \{0,1\}^n$ tal que para todo $x, y \in \{0,1\}^n$ distintos,

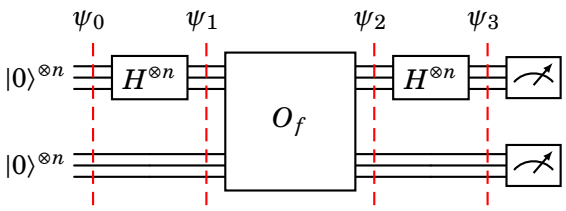
$$f(x) = f(y) \iff x \oplus y = s.$$

En este caso, decimos que f es *uno a uno* si $s = 0^n$ y *dos a uno* si $s \neq 0^n$. Si sabemos que f es invariante bajo una máscara XOR, el problema de Simon consiste en identificar en cuál de las dos situaciones anteriores está.

El estudio de este problema mediante computación clásica implicaría ir revisando secuencialmente los elementos de $\{0,1\}^n$ hasta encontrar una imagen repetida de f (en este caso es un problema *dos a uno*), o haber recorrido la mitad más uno de los elementos sin repetir imagen (en este caso es un problema *uno a uno*, puesto que ya podemos descartar la existencia de un s con las características anteriores). Por lo tanto, se trata de un problema de complejidad $O(2^n)$.

Para abordar este problema en computación cuántica optaremos por un procesamiento mixto. Primero ejecutaremos un algoritmo cuántico $O(n)$ veces y, a continuación, analizaremos el resultado mediante un algoritmo clásico que procesará las mediciones para finalmente dar una respuesta al problema. Veamos la estrategia (descrita en [41]) de forma más detallada.

El algoritmo consta de dos sistemas Q, B, que contienen n qubits cada uno, con espacios de estados $\mathscr{H}_Q \equiv \mathscr{H}_B \equiv \left(\mathbb{C}^2\right)^{\otimes n}$. El circuito que lo implementa es el siguiente:



El estado inicial del sistema es

$$|\psi_0\rangle = |0\rangle^{\otimes 2n},$$

y tras aplicar las puertas de Hadamard sobre el sistema Q obtenemos

$$|\psi_1\rangle = \left(H^{\otimes n} \otimes I^{\otimes n}\right)|0\rangle^{\otimes 2n} = (H|0\rangle)^{\otimes n} \otimes |0\rangle^{\otimes n},$$

que puede reescribirse de la siguiente forma aplicando (2.13):

$$|\psi_1\rangle = \sum_{k\in\{0,1\}^n} \frac{|k\rangle}{\sqrt{2^n}} \otimes |0\rangle^{\otimes n}.$$

A continuación, interviene el oráculo:

$$|\psi_2\rangle = O_f|\psi_1\rangle = O_f\left(\sum_{k\in\{0,1\}^n} \frac{|k\rangle}{\sqrt{2^n}} \otimes |0\rangle^{\otimes n}\right) = \frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle \otimes |f(k)\rangle.$$

Finalmente, volviendo a aplicar puertas de Hadamard sobre el sistema Q, el estado evoluciona de la siguiente forma

$$|\psi_3\rangle = \left(H^{\otimes n} \otimes I^{\otimes n}\right)|\psi_2\rangle = H^{\otimes n}\left(\frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle\right) \otimes |f(k)\rangle = \frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} H^{\otimes n}|k\rangle \otimes |f(k)\rangle$$

que, por la igualdad (3.8), podemos reescribir como

$$= \frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} \sum_{z\in\{0,1\}^n} \frac{(-1)^{kz}|z\rangle}{\sqrt{2^n}} \otimes |f(k)\rangle = \frac{1}{2^n} \sum_{k\in\{0,1\}^n} \sum_{z\in\{0,1\}^n} (-1)^{kz}|z\rangle \otimes |f(k)\rangle.$$

En este punto realizamos una medición del sistema. Analizaremos por separado el efecto de la medición dependiendo de si la función es *uno a uno* o *dos a uno*:

- Función *uno a uno*: En este caso, como f es biyectiva, el estado $|\psi_3\rangle$ es superposición de todos los elementos de la base computacional, y el módulo de las amplitudes es en todos los casos $1/2^n$. Aplicando (2.16) tenemos que, para cada imagen $w \in f(\{0,1\}^n)$,

$$P(w) = \sum_{j\in\{0,1\}^n} \left(\frac{1}{2^n}\right)^2 = 2^n \frac{1}{2^{2n}} = \frac{1}{2^n}. \tag{3.11}$$

- Función *dos a uno*: En este caso, para cada imagen $w \in f(\{0,1\}^n)$ hay dos elementos $k_1, k_2 \in \{0,1\}^n$ tales que $f(k_1) = f(k_2) = w$. Si denominamos $E \subset \{0,1\}^n \times \{0,1\}^n$ al conjunto formado por los pares de elementos que tienen la misma imagen, podremos reescribir la expresión del estado $|\psi_3\rangle$ de la siguiente forma:

$$\begin{aligned}
|\psi_3\rangle &= \frac{1}{2^n} \sum_{k\in\{0,1\}^n} \sum_{z\in\{0,1\}^n} (-1)^{kz} |z\rangle \otimes |f(k)\rangle \\
&= \frac{1}{2^n} \sum_{(k_1,k_2)\in E} \sum_{z\in\{0,1\}^n} \left((-1)^{k_1 z} + (-1)^{k_2 z}\right) |z\rangle \otimes |f(k_1)\rangle \\
&= \frac{1}{2^n} \sum_{(k_1,k_2)\in E} \sum_{z\in\{0,1\}^n} \left((-1)^{k_1 z} + (-1)^{(k_1 \otimes s) z}\right) |z\rangle \otimes |f(k_1)\rangle \\
&= \frac{1}{2^n} \sum_{(k_1,k_2)\in E} \sum_{z\in\{0,1\}^n} \left((-1)^{k_1 z}\,(1 + (-1)^{sz}\right) |z\rangle \otimes |f(k_1)\rangle\,.
\end{aligned}$$

  A partir de esta expresión, si calculamos la probabilidad de que el resultado de la medición sobre el sistema Q sea w aplicando (2.16), obtenemos

$$P(w) = \sum_{(k1,k_2)\in E} \left| \frac{1}{2^n} (-1)^{k_1 w} (1 + (-1)^{sw}) \right|^2 .$$

  En el caso de que $sw \equiv 1$ (mód 2), la expresión anterior generará una probabilidad $P(w) = 0$ (ya que todos los términos del sumatorio se anularán), y en el caso $sw \equiv 0$ (mód 2), la probabilidad de obtener w será

$$\begin{aligned}
P(w) &= \sum_{(k1,k_2)\in E} \left| \frac{1}{2^n} (-1)^{k_1 w} (1 + (-1)^0) \right|^2 = \sum_{(k1,k_2)\in E} \left| \frac{2(-1)^{k_1 w}}{2^n} \right|^2 = \sum_{(k1,k_2)\in E} \frac{1}{2^{2n-2}} = \frac{2^{n-1}}{2^{2n-2}} \\
&= \frac{1}{2^{n-1}}.
\end{aligned}$$

Es decir, el resultado w obtenido cumplirá en cualquier caso que $sw \equiv 0$ (mód 2) (si es *uno a uno*, porque $s = 0$; si es *dos a uno*, porque en caso contrario $P(w)$ tomaría valor nulo y no habría sido posible obtener el resultado w en la medición).

La parte clásica del algoritmo de Simon consiste en invocar iterativamente al algoritmo cuántico anterior e ir registrando las mediciones del primer bloque Q hasta que hayamos obtenido n soluciones distintas $w_1, ..., w_n$ linealmente independientes. La probabilidad de que esto suceda es superior a 1/2 (ver [37, pp.55-56]), con lo que la probabilidad de obtener un conjunto de soluciones válido después de k intentos es de $1 - 2^{-k}$, un valor que podemos forzar que sea arbitrariamente próximo a 1 variando el número de iteraciones. Así pues, este paso requiere $O(n)$ iteraciones con muy alta probabilidad.

Como sabemos que para cada $j \in [n]$, $w_j s \equiv 0$ (mód 2), planteamos y resolvemos el sistema de ecuaciones

$$\{w_j s \equiv 0 \text{ (mód 2)}\}_{j\in[n]}.$$

Este sistema tiene dos soluciones $s_0, s_1 \in \{0,1\}^n$: la trivial $s_0 = 0^n$ y la no trivial $s_1 \neq 0^n$. Evaluando $f(s_1)$ podemos concluir qué tipo de función es f:

- Si $f(s_1) = f(0^n)$, entonces $f(0^n) = f(0^n \oplus s_1) = f(s_1)$, así que $s = s_1$ y f es *dos a uno*.
- Si $f(s_1) \neq f(0^n)$, entonces $s = s_0 = 0^n$, con lo que f es *uno a uno*.

Y de esta forma obtenemos la solución del problema.

Veamos cómo encaja este algoritmo con lo descrito en la Sección 3.2.1. En primer lugar, se trata de un algoritmo basado en el modelo de caja negra. A nivel de complejidad cuántica, el circuito ejecuta $O(1)$ veces el oráculo, pero es invocado desde la parte clásica $O(n)$ veces. Posteriormente hay un post-proceso clásico que requiere $O(n^2)$ instrucciones (resolución del sistema de ecuaciones). Por lo tanto, la complejidad global del algoritmo es polinomial. Al inicio de esta sección hemos observado que el algoritmo clásico equivalente tiene complejidad $O(2^n)$, cosa que implica un sobrecoste superpolinomial.

Analizando el problema desde el punto de la vista de la interpretación que hicimos de la Conjetura Cuántica, el sobrecoste superpolinomial debería implicar que la fracción de datos de entrada del problema se reduce muy rápidamente al crecer el número de inputs n. Comprobemos que es así. Si denotamos $N = 2^n$, existen N^N funciones posibles, pero sólo $N!$ de ellas son *uno a uno* y $(N-1)N!/(N/2)!$ son *dos a uno* (obtenemos este número teniendo en cuenta que hay un total de $N-1$ máscaras posibles, que la elección de la máscara determina los pares de elementos que tienen la misma imagen, y que las posibles asignaciones de imágenes a cada par de elementos es $N!/(N/2)!$).

Debido a lo anterior, la expresión de la proporción de funciones *uno a uno* o *dos a uno* respecto al total (que denotamos $\phi(n)$) es

$$\phi(n) = \frac{(2^n)! + \frac{(2^n-1)(2^n)!}{(2^{n-1})!}}{(2^n)^{2^n}}.$$

Aplicando la fórmula de Stirling (3.10),

$$\phi(n) \sim \frac{\sqrt{2\pi 2^n}\left(\frac{2^n}{e}\right)^{2^n}}{(2^n)^{2^n}} + \frac{(2^n-1)\sqrt{2\pi 2^n}\left(\frac{2^n}{e}\right)^{2^n}}{\sqrt{2\pi 2^{n-1}}\left(\frac{2^{n-1}}{e}\right)^{2^{n-1}}(2^n)^{(2^n)}} = \sqrt{\pi}2^{\frac{n+1}{2}}e^{-2^n} + (2^n-1)2^{\frac{1}{2}+(2-n)2^{n-1}} \sim 2^{n+\frac{1}{2}+(2-n)2^{n-1}}$$

Es decir,

$$\phi(n) = O\left(2^{-n2^n}\right),$$

con lo que $\phi(n)$ decrece asintóticamente todavía más rápido que una exponencial (y, por lo tanto, más rápido que en el algoritmo de Deutsch-Jozsa). Concluimos que la evolución del conjunto de

inputs del problema cuando n incrementa es coherente con la interpretación de la Conjetura Cuántica.

CAPÍTULO 4

CONJETURA AARONSON-AMBAINIS

4.1. Funciones booleanas

A continuación introduciremos algunos conceptos que serán necesarios para las próximas secciones. Empezaremos describiendo el concepto de polinomio multilineal real de n variables, que es una aplicación $f : \{0,1\}^n \to \mathbb{R}$ de la forma

$$f(X) = \sum_{S \subset [n]} c(S) \prod_{i \in S} x_i, \tag{4.1}$$

donde $X = (x_1, ..., x_n) \in \{0,1\}^n$ y $c(S) \in \mathbb{R}$ es el coeficiente correspondiente a $\prod_{i \in S} x_i$.

Se trata de un polinomio de n variables booleanas que se agrupa en monomios producto de algunas de estas variables multiplicadas por un coeficiente real. El número de variables que intervienen en cada monomio se denomina grado del monomio, y el grado del polinomio es el grado máximo de sus monomios. Sintetizamos lo descrito en una única definición:

Definición 4.1. Un **polinomio multilineal real de N variables y grado** d es una aplicación $f : \{0,1\}^n \to \mathbb{R}$ de la forma

$$f(X) = \sum_{S \subset [n], |S| \leq d} c(S) \prod_{i \in S} x_i$$

donde $X = (x_1, ..., x_n) \in \{0,1\}^n$ y $c(S) \in \mathbb{R}$ es el coeficiente correspondiente a $\prod_{i \in S} x_i$.

El mismo concepto puede aplicarse sobre el cuerpo de los complejos si definimos la aplicación $f : \{0,1\}^n \to \mathbb{C}$ y los coeficientes $c(S)$ toman valores complejos. En este caso, denominaremos *polinomio multilineal complejo* a esta aplicación.

Teorema 4.1. *Toda función $f : \{0,1\}^n \to \mathbb{R}$ se puede expresar como polinomio multilineal,*

$$f(x) = \sum_{S \subset [n]} c(S) \prod_{i \in S} x_i,$$

donde $c(S) \in \mathbb{R}$.

Posponemos la demostración de este teorema a la Sección 5.1, donde argumentaremos su validez con las herramientas que en ese punto del trabajo ya habremos desarrollado.

En adelante, haremos uso de esta asociación entre función booleana y polinomio multilineal sin volver a justificarla. Concretamente, sobreentenderemos que el grado de una función es el grado de su polinomio multilineal asociado.

Veamos un ejemplo de cómo podría aplicarse este teorema a la siguiente aplicación:

$$f : \{0,1\}^2 \to \mathbb{R}, \qquad f(x_1, x_2) = 2^{1+2x_1-x2}$$

Evaluando la función en $\{0,1\}^2$ obtenemos estos valores

$$f(0,0) = 2, \qquad f(1,0) = 8, \qquad f(0,1) = 1, \qquad f(1,1) = 4,$$

a partir de los cuales asignamos los coeficientes

$$c(\emptyset) = 2, \qquad c(\{1\}) = 6, \qquad c(\{2\}) = -1, \qquad c(\{1,2\}) = -3$$

que definen el siguiente polinomio multilineal $p : \{0,1\}^2 \to \mathbb{R}$:

$$p(x_1, x_2) = 2 + 6x_1 - x_2 - 3x_1x_2.$$

Podemos comprobar que p coincide con f en todo su dominio, así que ambas son la misma función.

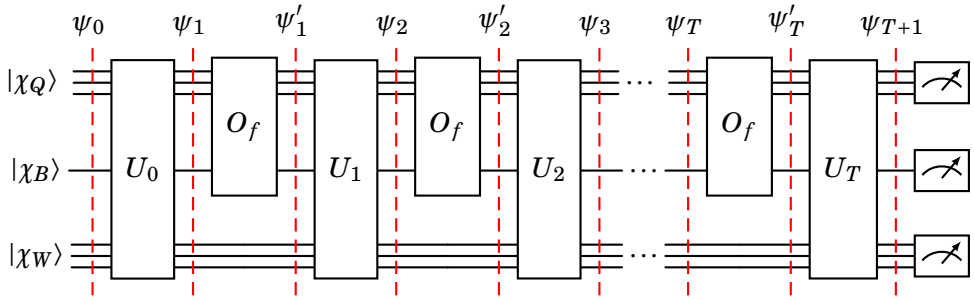
4.2. Probabilidad de aceptación como polinomio

En esta sección estudiaremos cómo podemos modelizar la probabilidad de aceptación de un algoritmo cuántico de caja negra mediante un polinomio multilineal real.

4.2.1. Oráculos de un qubit

Empezamos revisando un algoritmo como el detallado en la Sección 3.1.1, pero restringido al caso particular $m = 1$. Se trata de una simplificación del modelo de oráculo estudiado hasta ahora, pero hemos optado por revisar en primer lugar este caso por ser el analizado en [10] y en toda la bibliografía a la que hemos tenido acceso. Más adelante ampliaremos la conclusión al caso general.

La estructura de información del algoritmo está formada por tres sistemas Q, B, W, cuyos estados toman valor en $\mathscr{H}_Q \equiv \left(\mathbb{C}^2\right)^{\otimes n}$, $\mathscr{H}_B \equiv \mathbb{C}^2$, $\mathscr{H}_W \equiv \left(\mathbb{C}^2\right)^{\otimes l}$.



Al final del algoritmo efectuamos una medición del sistema, momento en el que se pueden producir dos situaciones: que el algoritmo se acepte (es decir, que su salida esté contenida en el conjunto $D \subset \{0,1\}^{n+1+l}$ correspondiente a los outputs que consideramos aceptados), o que no se acepte (en caso contrario).

En la Sección 3.1.2 introdujimos el concepto de probabilidad de aceptación del algoritmo, veamos cómo podría ser una función que devuelva su valor.

Teorema 4.2. *La probabilidad de aceptación de un algoritmo de T queries como el descrito se puede calcular mediante un polinomio multilineal real $p : \{0,1\}^N \to [0,1]$, con $N = 2^n$ variables y grado máximo $2T$ [4, pp. 153-154].*

Demostración. El algoritmo realiza T queries. Tal como se observa en el diagrama anterior, para cada $i \in [T]$ denotamos $|\psi_i\rangle$ al estado previo a la i-ésima invocación al oráculo, $|\psi'_i\rangle$ al posterior, y $|\psi_0\rangle, |\psi_{T+1}\rangle$ a los estados inicial y final del circuito (antes de la medición). El Oráculo O_f tiene asociada una función $f : \{0,1\}^n \to \{0,1\}$, que representaremos como vector $X = (f(k))_{\{0,1\}^n}$, según lo descrito en la Sección 3.1.1 (con la única particularidad de que aquí estamos asumiendo $m = 1$) .

El primer objetivo será estudiar cómo evoluciona el sistema a lo largo del algoritmo, y demostrar que para cada $i \in [T]$, los estados $|\psi_i\rangle, |\psi'_i\rangle$ se pueden representar como combinación compleja de la base computacional con amplitudes que se corresponden con polinomios multilineales complejos de variable X, y de grado $i-1$ e i respectivamente, concluyendo que las amplitudes de $|\psi_{T+1}\rangle$ son también un polinomio multilineal de grado máximo T.

La nomenclatura que usaremos para estos polinomios es la siguiente:

- $p_{0ik} : \{0,1\}^n \to \mathbb{C}$ representa la amplitud del estado del sistema $|\psi_i\rangle$, con $i \in \{0, \cdots, T+1\}$ respecto al estado básico $k \in \{0,1\}^{n+1+l}$
- $p_{1ik} : \{0,1\}^n \to \mathbb{C}$ representa la amplitud del estado del sistema $|\psi'_i\rangle$, con $i \in \{1, \cdots, T\}$, también respecto al estado básico $k \in \{0,1\}^{n+1+l}$.

Sea $|k\rangle$ un elemento de la base computacional, denotaremos su descomposición en los sistemas Q, B, W de la siguiente forma $|k\rangle := |q_k, b_k, w_k\rangle \in \mathcal{H}_Q \otimes \mathcal{H}_B \otimes \mathcal{H}_W$. Usaremos una notación análoga para el estado inicial $|\psi_0\rangle := |q, b, w\rangle \in \mathcal{H}_Q \otimes \mathcal{H}_B \otimes \mathcal{H}_W$.

Empezamos señalando que $|\psi_0\rangle$ puede descomponerse como combinación lineal compleja de elementos de esta base (uno de los coeficientes será 1 y el resto 0), así que podemos representar el estado como

$$|\psi_0\rangle = |q_0, b_0, w_0\rangle = \sum_{k\in\{0,1\}^{n+1+l}} p_{00k}(X)|k\rangle\,, \tag{4.2}$$

donde los polinomios $p_{00k} : \{0,1\}^n \to \mathbb{C}$ son de grado 0 (constantes) para todo $k \in \{0,1\}^{n+1+l}$.

El siguiente estado es $|\psi_1\rangle$, resultado de aplicar el operador U_0 sobre $|\psi_0\rangle$. Como la función f no interviene en este operador, no se incorporará ninguna variable de X en la expresión de las amplitudes del nuevo estado, así que la representación de $|\psi_1\rangle$ como combinación compleja de elementos de la base computacional tendrá coeficientes $p_{01k} : \{0,1\}^n \to \mathbb{C}$ que también serán constantes:

$$|\psi_1\rangle = \sum_{k\in\{0,1\}^{n+1+l}} p_{01k}(X)|k\rangle \tag{4.3}$$

Supongamos, como hipótesis de inducción, que para un $i \in [T]$ el estado $|\psi_i\rangle$ puede descomponerse como

$$|\psi_i\rangle = \sum_{k\in\{0,1\}^{n+1+l}} p_{0ik}(X)|k\rangle\,,$$

con polinomios multilineales $p_{0ik} : \{0,1\}^n \to \mathbb{C}$ de grado máximo $i-1$ para todo $k \in \{0,1\}^{n+1+l}$.

En este supuesto, el siguiente estado del algoritmo será

$$\begin{aligned}
|\psi_i'\rangle &= O_f\,|\psi_i\rangle \\
&= O_f\left(\sum_{k\in\{0,1\}^{n+1+l}} p_{0ik}(X)\ |k\rangle\right) \\
&= \sum_{k\in\{0,1\}^{n+1+l}} p_{0ik}(X)\, O_f\,|k\rangle \\
&= \sum_{k\in\{0,1\}^{n+1+l}} p_{0ik}(X)\, O_f\,|q_k, b_k, w_k\rangle \\
&= \sum_{k\in\{0,1\}^{n+1+l}} p_{0ik}(X)\ |q_k, b_k \oplus x_q, w_k\rangle \\
&= \sum_{k\in\{0,1\}^{n+1+l}} p_{0ik}(X)\ \left(x_q\,|q_k, b_k\oplus 1, w_k\rangle + (1-x_q)|q_k, b_k, w_k\rangle\right) \\
&= \sum_{k\in\{0,1\}^{n+1+l}} \left(p_{0ik}(X)\ x_q\,|q_k, b_k\oplus 1, w_k\rangle + p_{0ik}(X)\ (1-x_q)|q_k, b_k, w_k\rangle\right)
\end{aligned} \tag{4.4}$$

Por lo tanto, el nuevo estado seguirá siendo combinación compleja de estados de la base computacional con amplitudes expresables como polinomio multilineal complejo $p_{1ik} : \{0,1\}^n \to \mathbb{C}$ de grado máximo i (por la incorporación de la variable x_q a un polinomio de grado máximo $i-1$).

El siguiente estado $|\psi_{i+1}\rangle$ será el resultado de aplicar U_i sobre $|\psi'_i\rangle$. Como U_i no depende de la función f, los polinomios $p_{0(i+1)k} : \{0,1\}^n \to \mathbb{C}$ correspondientes a las amplitudes de este nuevo estado seguirán siendo de grado máximo i.

Teniendo en cuenta que en (4.3) habíamos verificado el caso $i = 1$, deducimos (por inducción) el resultado que proponíamos al inicio de la demostración: que el estado final $|\psi_{T+1}\rangle$ admite la siguiente descomposición

$$|\psi_{T+1}\rangle = \sum_{k\in\{0,1\}^{n+1+l}} p_{0(T+1)k}(X)|k\rangle\,, \tag{4.5}$$

donde $p_{0(T+1)k} : \{0,1\}^n \to \mathbb{C}$ son polinomios multilineales de grado máximo T.

Sea $D \subset \{0,1\}^{n+1+l}$ el conjunto de estados básicos asociado a la aceptación del algoritmo, aplicamos la ecuación (2.16) para calcular la probabilidad de que el resultado de la medición $r \in \{0,1\}^{n+1+l}$ esté en este conjunto:

$$P(r \in D) = \sum_{k\in D} \left|p_{0(T+1)k}(X)\right|^2.$$

Si descomponemos cada polinomio $p_{0(T+1)k}$ en su parte real e imaginaria, $p_{0(T+1)k}(X) = pr(X) + i \cdot pi(X)$, obtenemos dos polinomios multilineales reales $pr, pi : \{0,1\}^{n+1+l} \to \mathbb{R}$ de grado máximo T, y ambos con las mismas 2^n variables. Por lo tanto,

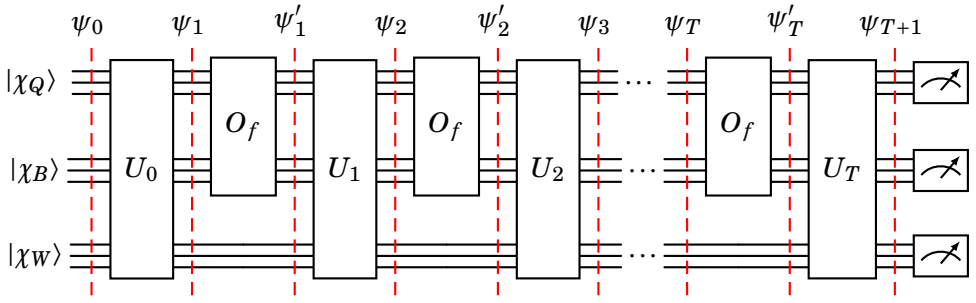
$$P(r \in D) = \sum_{k\in D} \left|p_{0(T+1)k}(X)\right|^2 = \sum_{k\in D} \left(pr(X)^2 + pi(X)^2\right)$$

Cada término del sumatorio anterior es un polinomio multilineal real de grado máximo $2T$, así que la probabilidad de aceptación del algoritmo también lo es, completando la demostración del teorema. □

4.2.2. Oráculos de m qubits

En la versión previa del teorema hemos considerado una simplificación del algoritmo al caso en que el sistema B contiene un sólo qubit. A continuación revisaremos la vigencia de sus conclusiones para el caso general de un sistema B con m qubits.

Partimos de un algoritmo como el de la Sección 3.1.2, formado por los sistemas Q, B, W, con estados que toman valor en $\mathcal{H}_Q \equiv \left(\mathbb{C}^2\right)^{\otimes n}$, $\mathcal{H}_B \equiv \left(\mathbb{C}^2\right)^{\otimes m}$, $\mathcal{H}_W \equiv \left(\mathbb{C}^2\right)^{\otimes l}$, tal como se representa en el siguiente diagrama:



En este caso, la función del oráculo tiene forma $f : \{0,1\}^n \to \{0,1\}^m$, así que las imágenes respecto a f de cada elemento de la base computacional del sistema Q son elementos de $\{0,1\}^m$. Denotamos el vector que representa al oráculo $X \in \{0,1\}^N$, con $N = m2^n$, y representamos sus $m2^n$ coordenadas de la siguiente forma

$$X = \left(x_{11}, x_{12}, \ldots, x_{1m}, x_{21}, \ldots, x_{2n}, \ldots, x_{(2^n)1}, x_{(2^n)2}, \ldots, x_{(2^n)m}\right),$$

El estudio de la evolución del estado del sistema que se detalla en la demostración del Teorema 4.2 seguiría siendo válido, aunque habrá que hacer algún ajuste debido al cambio de estructura del vector X. Concretamente, la ecuación (4.4) donde se describe la construcción del polinomio correspondiente a p_{1ik} a partir de las amplitudes previas en este caso recibe influencia de todos los estados de la base computacional de B.

Veamos un ejemplo. Supongamos $m = 3$. Ahora, la variable x_q consta de tres coordenadas binarias que denotamos $x_q := (x_{q1}, x_{q2}, x_{q3})$. Teniendo en cuenta esto, la expresión del estado $|\psi'_i\rangle$ descrita en (4.4), en este caso tomará la siguiente forma:

(4.6)

$$\begin{aligned}
|\psi'_i\rangle &= O_f\,|\psi_i\rangle \\
&= O_f\left(\sum_{k\in\{0,1\}^{n+m+l}} p_{0ik}(X)\ |k\rangle\right) \\
&= \sum_{k\in\{0,1\}^{n+m+l}} p_{0ik}(X)\, O_f\,|k\rangle \\
&= \sum_{k\in\{0,1\}^{n+m+l}} p_{0ik}(X)\, O_f\,|q_k, b_k, w_k\rangle \\
&= \sum_{k\in\{0,1\}^{n+m+l}} p_{0ik}(X)\ |q_k, b_k \oplus x_q, w_k\rangle \\
&= \sum_{k\in\{0,1\}^{n+m+l}} p_{0ik}(X)\,((1-x_{q1})(1-x_{q2})(1-x_{q3})\ |q_k, b_k, w_k\rangle + (1-x_{q1})(1-x_{q2})x_{q3}\ |q_k, b_k\oplus 1, w_k\rangle \\
&\qquad + (1-x_{q1})x_{q2}(1-x_{q3})\ |q_k, b_k\oplus 2, w_k\rangle \ +\ (1-x_{q1})x_{q2}x_{q3}\ |q_k, b_k\oplus 3, w_k\rangle \\
&\qquad + x_{q1}(1-x_{q2})(1-x_{q3})\ |q_k, b_k\oplus 4, w_k\rangle \ +\ x_{q1}(1-x_{q2})x_{q3}\ |q_k, b_k\oplus 5, w_k\rangle \\
&\qquad + x_{q1}x_{q2}(1-x_{q3})\ |q_k, b_k\oplus 6, w_k\rangle \ +\ x_{q1}x_{q2}x_{q3}\ |q_k, b_k\oplus 7, w_k\rangle)
\end{aligned}$$

Cada término del sumatorio incluye 8 sumandos, y cada uno de éstos contribuye a la amplitud de $|\psi'_i\rangle$ respecto a algún estado de la base computacional. En todos los casos, esta contribución es un polinomio de grado máximo el grado de p_{0ik} más 3. Aplicando esta idea sobre la demostración por inducción del Teorema 4.2, cada iteración de la inducción incrementará el grado del polinomio en un máximo de 3 unidades, con lo que el grado máximo de p_{0Tk} será $3T$.

Trasladando este ejemplo al caso genérico $m > 1$, la expansión de la ecuación correspondiente a (4.6) generará 2^m sumandos dentro del sumatorio general, y cada uno de ellos incrementará en un máximo de m el grado del polinomio previo, concluyendo de forma análoga a lo que hemos descrito que el grado del polinomio p_{0Tk} será como máximo mT.

Con esta única modificación podemos aplicar el mismo razonamiento del teorema previo, concluyendo que la probabilidad de aceptación se corresponde con un polinomio $p : \{0,1\}^{m2^n} \to \mathbb{R}$ de grado $2mT$ y $m2^n$ variables.

Resumimos la conclusión de lo descrito en el siguiente corolario:

Corolario 4.1. *La probabilidad de aceptación de un algoritmo de T queries como el descrito se puede calcular mediante un polinomio multilineal real $p : \{0,1\}^N \to [0,1]$, con $N = m2^n$ variables y grado máximo $2mT$.*

4.3. La Conjetura AA

4.3.1. Preliminares y enunciado

Sea $p : \{0,1\}^N \to \mathbb{R}$ un polinomio multilineal real para algún $N \in \mathbb{N}$, y $X \in \{0,1\}^N$ una variable aleatoria con distribución uniforme en $\{0,1\}^N$, definimos los siguientes conceptos:

La **varianza** de p es

$$\mathrm{Var}[p] := E_{X\in\{0,1\}^N}\left[(p(X) - E[p])^2\right]. \quad [1] \tag{4.7}$$

La **influencia** de la variable i en p es

$$\mathrm{Inf}_i[p] := E_{X\in\{0,1\}^N}\left[(p(X) - p(X^{(i)}))^2\right], \tag{4.8}$$

siendo $X^{(i)}$ el resultado de aplicar un NOT en la i-ésima coordenada del vector X.

También definimos la **máxima influencia** de f como

$$\mathrm{maxInf}[f] := \max_{i\in[N]}\{\mathrm{Inf}_i[f]\}, \tag{4.9}$$

y la **influencia total** de p es

$$\mathrm{Inf}[p] := \sum_{i\in[N]} \mathrm{Inf}_i[p]. \tag{4.10}$$

Sea $j \in [N]$, la **restricción del polinomio** p para la variable x_j a un valor $K \in \mathbb{R}$ es un polinomio $p_{|x_j=K} : \mathbb{R}^{N-1} \to \mathbb{R}$ tal que

$$p_{|x_j=K}(x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_N) = p(x_1,\ldots,x_{j-1},K,x_{j+1},\ldots,x_N) \tag{4.11}$$

Por último, antes de enunciar la conjetura introducimos el siguiente lema que será necesario más adelante, cuya demostración se puede encontrar en [4, p. 19]:

[1] En diversos puntos de este trabajo escribiremos E[variable aleatoria] o E[función] como notación abreviada de la esperanza de esta variable o función sobre todos sus posibles valores. En este caso, denotamos $E[p]$ como $E_{X\in\{0,1\}^N}[p(X)]$.

Lema 4.1. *Sea* $p : \{0,1\}^N \to [0,1]$ *un polinomio multilineal real de grado* d*. Entonces,*

$$\mathrm{Inf}[p] \leq d \tag{4.12}$$

A partir de los conceptos previos, Aaronson y Ambainis expusieron la siguiente conjetura en su primera versión del artículo [4] en 2009, que es conocida como *Conjetura Aaronson-Ambainis (Conjetura AA)*.

Conjetura 4.1. *(Conjetura AA) [4, p. 139] [2]. Sea* $p : \{0,1\}^N \to [0,1]$ *un polinomio multilineal real de grado* d*, existe* $i \in [N]$ *tal que* $\mathrm{Inf}_i[p] \geq w(\mathrm{Var}[p]/d)$*, donde* $w : \mathbb{R} \to \mathbb{R}$ *es un polinomio de la forma* $w(x) = Kx^a$*, para algunas constantes* $K, a > 0$*.*

4.3.2. Relación entre conjeturas

En [4, pp. 152–153] Aaronson y Ambainis enunciaron un teorema que relaciona esta conjetura con la Conjetura Cuántica declarada en la Sección 3.2:

Teorema 4.3. *La Conjetura 4.1 (Conjetura AA) implica la Conjetura 3.1 (Conjetura Cuántica).*

La demostración que presentamos es una versión modificada de la descrita en [4], adaptada para el caso de oráculos de m qubits. La principal diferencia reside en que se basa en el polinomio descrito en el Corolario 4.1, en vez de usar el resultado del Teorema 4.2 que corresponde al caso de 1 qubit. Además, hemos cambiado algunos planteamientos y notación de la demostración original con el objetivo de justificar de forma más detallada diversos pasos.

Demostración. Estudiaremos el algoritmo correspondiente al Corolario 4.1, formado por tres sistemas Q, B, W, cuyos estados toman valor en $\mathcal{H}_Q \equiv \left(\mathbb{C}^2\right)^{\otimes n}$, $\mathcal{H}_B \equiv \left(\mathbb{C}^2\right)^{\otimes m}$, $\mathcal{H}_W \equiv \left(\mathbb{C}^2\right)^{\otimes l}$, y con T invocaciones al oráculo. Tal como señalábamos en ese corolario, la probabilidad de aceptación del algoritmo se puede modelizar como un polinomio $p : \{0,1\}^N \to [0,1]$ de grado $2mT$ y $N = m2^n$ variables booleanas de entrada.

Consideramos una variable aleatoria X con distribución uniforme en $\{0,1\}^N$. El objetivo de la demostración es calcular (o aproximar, bajo los supuestos de la Conjetura Cuántica) la evaluación del polinomio $p(X)$ para cada posible valor de $X \in \{0,1\}^N$. Como el número de coordenadas de X es exponencial respecto a n, en el caso general no será posible evaluar p con un número polinomial de queries, así que la estrategia será construir un árbol de decisión determinista que aproxime $p(X)$ con un error máximo ε en una proporción mínima de $1 - \delta$ de los inputs tal que la profundidad de este árbol sea $\mathrm{poly}(T, 1/\varepsilon, 1/\delta)$ (recordemos que la complejidad de un árbol de decisión se corresponde con su profundidad).

El siguiente algoritmo describe cómo construir una rama de un árbol de decisión para un input $X \in \{0,1\}^N$ determinado.

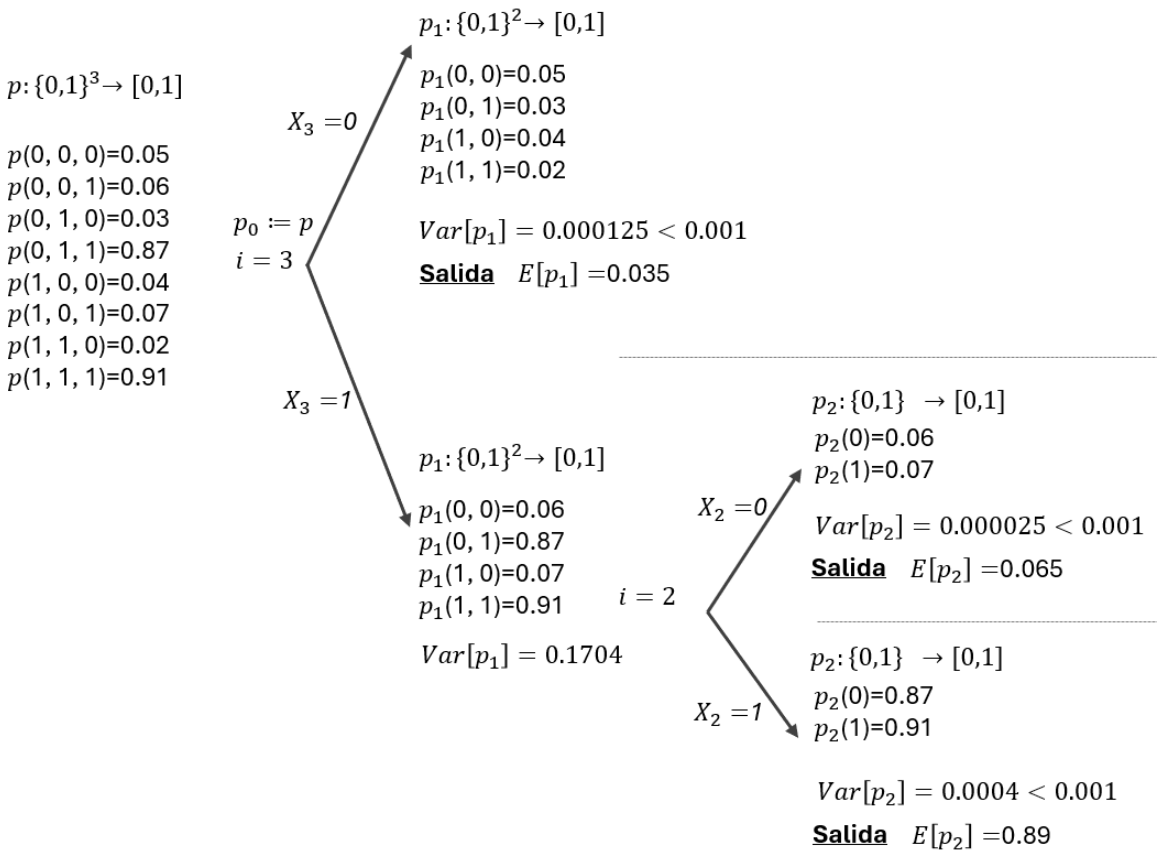
Algorithm 1 Construcción de una rama del árbol de decisión que aproxima $p(X)$

Require: $X \in \{0,1\}^N$, $p : \{0,1\}^N \to [0,1]$, $\varepsilon, \delta \in (0,1]$

$p_0 \leftarrow p$
for $j = 0 : N$ **do**
  if $\mathrm{Var}[p_j] \leq \varepsilon^2\delta/2$ **then** Finalizar y devolver $E[p_j]$
  else
    $i \leftarrow$ Variable tal que $\mathrm{Inf}_i[p_j] > w(\varepsilon^2\delta/4mT)$ (*)
    $p_{j+1} \leftarrow p_{j|Y_i=X_i}$
  end if
end for

(*) Donde $w : \mathbb{R} \to \mathbb{R}$ es el polinomio con forma $w(x) = Kx^a$ descrito en la Conjetura AA.

A continuación incluimos un diagrama que representa un posible ejemplo para una función $p : \{0,1\}^3 \to [0,1]$, ε =0.1, δ =0.2 (en este caso, la cota de la condición de salida es $\varepsilon^2\delta/2$ =0.001) [2]:



[2]Hemos optado por asignar un valor $N = 3$ por ser una dimensión adecuada para ilustrar la demostración. Sin embargo, este ejemplo no se corresponde con un algoritmo de caja negra válido, ya que $3 = m2^n$ implicaría $n = 0, m = 3$. Este hecho no resta validez al procedimiento descrito en la demostración ni a su aplicabilidad a este polinomio.

En el diagrama representamos el árbol completo (el que se generaría si ejecutáramos el algoritmo para todos los posibles $X \in \{0,1\}^N$). Observamos que para todos los inputs se identifica $i = 3$ como primera variable de alta influencia (aplicamos siempre el mismo criterio para elegir esta variable, por ejemplo tomando el índice más alto cuya influencia sea superior a la cota del algoritmo), y se genera un polinomio p_1 como restricción sobre esta coordenada para $X_3 = 0$ o $X_3 = 1$ en función de la tercera coordenada del input procesado.

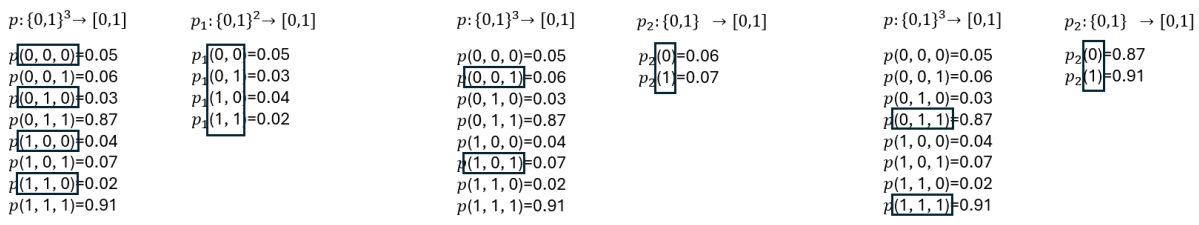
En el caso de la restricción sobre $X_3 = 0$, el polinomio generado tiene una varianza inferior a la cota de salida, así que el algoritmo devolverá $E[p_1]$ =0.035. En el caso de la restricción sobre $X_3 = 1$, la varianza es superior a la cota de salida, por lo que se itera otra vez. En este caso identificamos $i = 2$ como variable de alta influencia, y generamos el nuevo polinomio como restricción para $X_2 = 0$ o $X_2 = 1$, dependiendo otra vez del valor de la segunda coordenada del input procesado. En ambos casos la varianza del polinomio generado es inferior a la cota de salida, así que el algoritmo finaliza, devolviendo 0.065 o 0.89 en función de la rama aplicada.

Observamos que habrá conjuntos de inputs para los que el algoritmo recorrerá las mismas variables de alta influencia, aplicará las mismas restricciones, generará los mismos polinomios y devolverá el mismo output. Así pues, aunque la ejecución del algoritmo sea aislada para cada input, hay bloques de datos que generarán exactamente el mismo resultado.

Estudiemos cómo se distribuyen los distintos inputs en las distintas ramas de un árbol de decisión resultado de la ejecución del algoritmo sobre un polinomio arbitrario $p : \{0,1\}^N \to [0,1]$. Supongamos, sin pérdida de generalidad, que la primera variable de restricción fuera $i = 1$. En este caso, los conjuntos de datos que bifurcarían por la rama $X_1 = 0$ y $X_1 = 1$ serían $\{0\} \times \{0,1\}^{N-1}$ y $\{1\} \times \{0,1\}^{N-1}$, respectivamente. Al ser $X \in \{0,1\}^N$ una variable aleatoria con distribución uniforme, la restricción de X a estos dos subconjuntos seguirá también esta distribución.

Lo mismo sucederá en todas las demás bifurcaciones del árbol de decisión: el algoritmo optará por una rama en la mitad de las entradas correspondientes a la iteración previa y por la otra rama en la otra mitad. Estudiando este efecto en todo el árbol, concluimos que existe una identificación entre el bloque de inputs que llega a cada hoja y un subconjunto de $\{0,1\}^N$. Además, la colección de estos subconjuntos define una partición en $\{0,1\}^N$, y la restricción de la variable aleatoria X a cada subconjunto seguirá distribución uniforme.

Véase en el siguiente diagrama que en el ejemplo anterior esto se cumple:



A continuación verificaremos los siguientes aspectos del algoritmo:

1. Existe la variable i tal que $\text{Inf}_i[p_j] > w(\varepsilon^2\delta/4mT)$.
2. La probabilidad de que la salida del algoritmo genere un error superior a ε es inferior a $\delta/2$.
3. La probabilidad de que el algoritmo no finalice en un número de iteraciones $\text{poly}(T, 1/\varepsilon, 1/\delta)$ es inferior a $\delta/2$.
4. Los resultados previos 1, 2, 3 completan la demostración del teorema.

Empezamos:

1. Existe la variable i tal que $\text{Inf}_i[p_j] > w(\varepsilon^2\delta/4mT)$.

Sea una iteración $j \in [N]$ en la que no se ha cumplido la condición de salida, aplicando la Conjetura AA sobre p_j (recordemos que p_j tiene rango contenido en $[0,1]$) identificamos una variable $i \in [n]$ tal que $\text{Inf}_i[p_j] > w(\text{Var}[p_j]/d)$, donde d representa el grado de p_j. Como no se ha verificado la condición de salida, la varianza del polinomio p_j cumple la siguiente restricción: $\text{Var}[p_j] > \varepsilon^2\delta/2$.

Además, por el Corolario 4.1, el grado de p es $\leq 2mT$. Como para todo $j \in [N-1]$, el polinomio p_{j+1} se construye a partir de la restricción de p_j, el grado de cada nuevo polinomio será inferior o igual al del precedente, y en consecuencia deducimos por inducción que el grado de todo p_j es $\leq 2mT$.

Recopilando la información anterior, concluimos que existe $i \in [N]$ tal que:

$$\text{Inf}_i[p_j] > w\left(\frac{\text{Var}[p_j]}{d}\right) \geq w\left(\frac{\varepsilon^2\delta/2}{2mT}\right) = w\left(\frac{\varepsilon^2\delta}{4mT}\right) \tag{4.13}$$

Así que existe la variable i y el algoritmo está bien definido.

2. La probabilidad de que la salida del algoritmo genere un error superior a ε es inferior a $\delta/2$.

En primer lugar, recordemos la desigualdad de Markov, según la cual, si W es un suceso aleatorio numérico no negativo y $a \in \mathbb{R}^+$, entonces

$$P(W > a) \leq \frac{E[W]}{a} \tag{4.14}$$

Supongamos que el algoritmo está ejecutando un input $X \in \{0,1\}^N$. Sea $Z : \{0,1\}^N \to [N]$ una función que representa la profundidad de la rama del árbol generada por el algoritmo para cada posible input, y sea $p_{Z(X)}$ el polinomio multilineal que el algoritmo ha generado en la iteración de salida a partir del input X. Observamos que, por la forma en que se ha construido este polinomio, la evaluación de $p_{Z(X)}(X)$ coincide con la de $p(X)$.

Al haberse cumplido la condición de salida, se verifica la siguiente desigualdad:

$$\mathrm{Var}_{Y\in\{0,1\}^{N-Z(X)}}[p_{Z(X)}(Y)] \leq \frac{\varepsilon^2\delta}{2} \tag{4.15}$$

Teniendo esto en cuenta, estudiaremos la probabilidad de que el algoritmo produzca un error superior a ε en el output generado respecto a la evaluación de $p_{Z(X)}$:

(4.16)

$$\begin{aligned}
P_{Y\in\{0,1\}^{N-Z(X)}}\left(\left|p_{Z(X)}(Y)-E[p_{Z(X)}]\right|>\varepsilon\right) &= P_{Y\in\{0,1\}^{N-Z(X)}}\left(\left(p_{Z(X)}(Y)-E[p_{Z(X)}]\right)^2>\varepsilon^2\right)\\
&\leq \frac{E_{Y\in\{0,1\}^{N-Z(X)}}\left[\left(p_{Z(X)}(Y)-E[p_{Z(X)}]\right)^2\right]}{\varepsilon^2} && \text{por (4.14)}\\
&= \frac{\mathrm{Var}_{Y\in\{0,1\}^{N-Z(X)}}[p_{Z(X)}(Y)]}{\varepsilon^2} && \text{aplicando (4.7)}\\
&\leq \frac{\varepsilon^2\frac{\delta}{2}}{\varepsilon^2} && \text{por (4.15)}\\
&= \frac{\delta}{2}
\end{aligned}$$

Tal como hemos visto antes, existe una identificación entre un subconjunto de $\{0,1\}^N$ y el conjunto $\{0,1\}^{N-Z(X)}$ sobre el que hemos expresado la probabilidad de la ecuación anterior. Además, la variable aleatoria $Y \in \{0,1\}^{N-Z(X)}$ seguirá distribución uniforme para todo X. Así pues, deducimos de la probabilidad descrita en (4.16) que la fracción de inputs de este bloque para los que el error es superior a ε es inferior o igual a $1/\delta$. Y esto sucederá en todos los bloques correspondientes a cada hoja del árbol de decisión, ya que la ecuación (4.16) es válida para todo $X \in \{0,1\}^N$.

Por lo tanto, como antes hemos indicado que la colección de subconjuntos correspondientes a cada $\{0,1\}^{N-Z(X)}$ define una partición sobre $\{0,1\}^N$, la fracción total de inputs en los que el algoritmo devolverá un error superior a ε será también como mucho $1/\delta$. Esto justifica la siguiente desigualdad,

$$P_{X\in\{0,1\}^N}\left(\left|p(X)-E\left[p_{Z(X)}\right]\right|>\varepsilon\right) \leq \frac{\delta}{2} \tag{4.17}$$

que se corresponde con el objetivo a demostrar en este apartado.

3. La probabilidad de que el algoritmo no finalice en un número de iteraciones poly$(T, 1/\varepsilon, 1/\delta)$ es inferior a $\delta/2$.

Partimos de la definición de la función Z correspondiente al número de iteraciones ejecutadas por el algoritmo descrita en el apartado 1. A continuación demostraremos que

$$P\left(Z(X) > \frac{4mT}{\delta\, w\left(\frac{\varepsilon^2\delta}{4mT}\right)}\right) \leq \frac{\delta}{2} \tag{4.18}$$

Vamos a estudiar el comportamiento del algoritmo en una iteración $j \in [Z(X)-1]$ en la que no se ha cumplido la condición de salida. Sea $i \in [N]$ la variable elegida en esa iteración, observamos

que, para todo $k \in [N]$ tal que $k \neq i$ (denotaremos $Y_{-i} \in \{0,1\}^{N-j-1}$ al vector resultante de excluir la i-ésima coordenada de $Y \in \{0,1\}^{N-j}$):

$$
\begin{aligned}
E_{X_i \in \{0,1\}}\left[\mathrm{Inf}_k[p_{j+1}]\right] &= \frac{1}{2}\left(\mathrm{Inf}_k[p_j|X_i=0] + \mathrm{Inf}_k[p_j|X_i=1]\right) && X_i \text{ uniforme} \\
&= \frac{1}{2}\sum_{Y_i \in \{0,1\}} \mathrm{Inf}_k[p_j|X_i=Y_i] \\
&= \frac{1}{2}\sum_{Y_i \in \{0,1\}} \frac{1}{2^{N-j-1}} \sum_{Y_{-i} \in \{0,1\}^{N-j-1}} \left(p_j(Y) - p_j\left(Y^{(k)}\right)\right)^2 \\
&= \frac{1}{2^{N-j}} \sum_{Y \in \{0,1\}^{N-j}} \left(p_j(Y) - p_j\left(Y^{(k)}\right)\right)^2 \\
&= \mathrm{Inf}_k[p_j]
\end{aligned} \tag{4.19}
$$

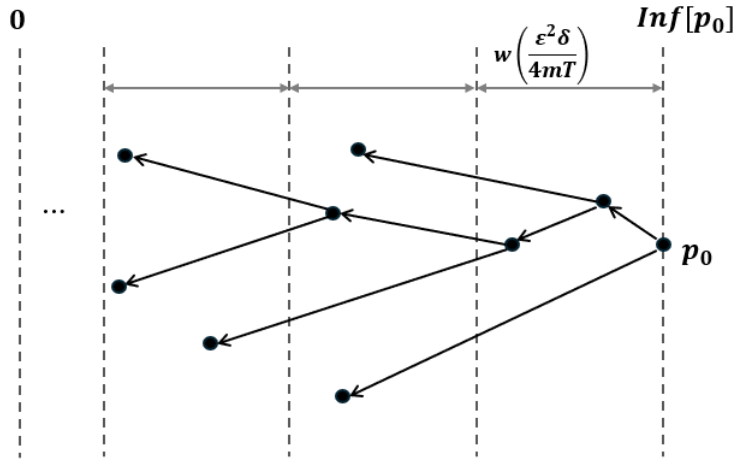
Observamos también que $\mathrm{Inf}_i[p_{j+1}] = 0$, ya que al ser i la variable de restricción, para todo $Y \in \{0,1\}^{N-j-1}$ se cumple $p_{j+1}(Y) = p_{j+1}\left(Y^{(i)}\right)$, y por lo tanto todos los términos del siguiente sumatorio se anulan:

$$
\mathrm{Inf}_i[p_{j+1}] = \frac{1}{2^{N-j-1}} \sum_{Y \in \{0,1\}^{N-j-1}} \left(p_{j+1}(Y) - p_{j+1}\left(Y^{(i)}\right)\right)^2 = 0 \tag{4.20}
$$

Así pues, si estudiamos la influencia total de p_{j+1},

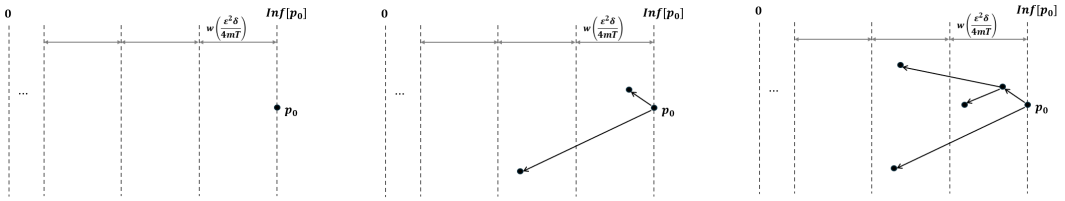
$$
\begin{aligned}
E_{X_i \in \{0,1\}}\left[\mathrm{Inf}[p_{j+1}]\right] &= E_{X_i \in \{0,1\}}\left[\sum_{k \in [N]} \mathrm{Inf}_k[p_{j+1}]\right] \\
&= E_{X_i \in \{0,1\}}\left[\left(\sum_{k \neq i \in [N]} \mathrm{Inf}_k[p_{j+1}]\right) + \mathrm{Inf}_i[p_{j+1}]\right] && \text{separando términos} \\
&= \sum_{k \neq i \in [N]} E_{X_i \in \{0,1\}}\left[\mathrm{Inf}_k[p_{j+1}]\right] && \text{aplicando (4.20)} \\
&= \sum_{k \neq i \in [N]} \mathrm{Inf}_k[p_j] && \text{aplicando (4.19)} \\
&= \mathrm{Inf}[p_j] - \mathrm{Inf}_i[p_j] \\
&\leq \mathrm{Inf}[p_j] - w\left(\frac{\varepsilon^2 \delta}{4mT}\right) && \text{aplicando (4.13)}
\end{aligned} \tag{4.21}
$$

A continuación representamos un posible árbol de decisión A, donde cada nodo representa un polinomio generado en una de las ramas del árbol. La ubicación horizontal en la que situamos cada nodo representa la influencia total del polinomio que asociado.



Evidentemente, no habrá ningún nodo situado a la izquierda de la vertical correspondiente al 0, puesto que las influencias son no negativas. Observamos también que al expandir un polinomio en sus dos restricciones se produce un decremento de la influencia de los nuevos polinomios respecto al previo. Concretamente, según la ecuación (4.21), el decremento medio es superior o igual a $w(\varepsilon^2\delta/4mT)$ o, lo que es lo mismo, la suma de los dos decrementos resultantes de la expansión es superior o igual a $2w(\varepsilon^2\delta/4mT)$.

Supongamos que definimos una secuencia de árboles $(A_k)_{k\in[K]}$, de tal forma que todo árbol de la secuencia es el resultado de expandir un nodo (un polinomio) en sus dos restricciones. El primer árbol de la secuencia contiene sólo el nodo correspondiente a p_0, y el último árbol A_K se corresponderá con el árbol A descrito antes. En el siguiente diagrama representamos el inicio de una posible secuencia, que seguiría expandiendo polinomios hasta completar el árbol A.



Para cada índice $k \in [K]$ definimos los siguientes conceptos:

- $Z_k : \{0,1\}^N \to [N]$ es la aplicación que representa la longitud de la rama del árbol A_k correspondiente al input $X \in \{0,1\}^N$. Observamos que $Z_0 = 0$ [3] y $Z_K = Z$.

- $T_k = \sum_{X\in\{0,1\}^N} Z_k(X)$ se corresponde con la suma de longitudes de las ramas de un árbol A_k para cada posible input (puede haber diversos inputs que se procesen con la misma rama, en este caso se sumará la longitud de la rama tantas veces como inputs procesados).

[3] En este caso 0 representa la función nula

- $S_k = \sum_{X\in\{0,1\}^N}(\text{Inf}[p_0] - \text{Inf}[p_{Z_k(X)}])$, donde $p_{Z_k(X)}$ representa el último polinomio generado en el árbol A_k para el input $X \in \{0,1\}^N$.

Observamos que, para $k = 0$, se cumple que $T_0 = S_0 = 0$. Además, como los árboles consecutivos se generan a partir de la expansión de un nodo del árbol previo, existe la siguiente relación en la suma de longitudes:

$$T_{k+1} = T_k + 2 \tag{4.22}$$

Por otro lado, antes hemos comprobado que el decremento de influencias resultado de una expansión es superior o igual a $2w(\varepsilon^2\delta/4mT)$, así que

$$S_{k+1} \geq S_k + 2w(\varepsilon^2\delta/4mT) \tag{4.23}$$

Aplicando inducción sobre (4.22) y (4.23) obtenemos que, para todo índice $k \in [K]$ de la secuencia de árboles,

$$T_k \leq \frac{S_k}{w\left(\frac{\varepsilon^2\delta}{4mT}\right)}. \tag{4.24}$$

Y además, como todos los polinomios tienen influencia no negativa, podemos determinar la siguiente cota para S_k:

$$S_k = \sum_{X\in\{0,1\}^N}(\text{Inf}[p_0] - \text{Inf}[p_{Z_k(X)}]) \leq \sum_{X\in\{0,1\}^N}\text{Inf}[p_0] = 2^N\text{Inf}[p_0] \tag{4.25}$$

De los resultados previos deducimos que ,

$$\begin{aligned}
E[Z] &= \frac{1}{2^N}\sum_{X\in\{0,1\}^N} Z_K(X) \\
&= \frac{1}{2^N}\, T_K(X) && \text{por definición de } T_k \\
&\leq \frac{1}{2^N}\frac{S_K}{w\left(\frac{\varepsilon^2\delta}{4mT}\right)} && \text{por (4.24)} \\
&\leq \frac{1}{2^N}\frac{2^N\text{Inf}[p_0]}{w\left(\frac{\varepsilon^2\delta}{4mT}\right)} && \text{por (4.25)} \\
&= \frac{\text{Inf}[p_0]}{w\left(\frac{\varepsilon^2\delta}{4mT}\right)} \\
&\leq \frac{2mT}{w\left(\frac{\varepsilon^2\delta}{4mT}\right)} && \text{por el Lema 4.1}
\end{aligned} \tag{4.26}$$

Con lo que concluimos que la probabilidad de que $Z(X)$ sea superior a la cota definida en (4.18) es:

$$\begin{aligned} P_{X\in\{0,1\}^N}\left(Z(X) > \frac{4mT}{\delta\ w\left(\frac{\varepsilon^2\delta}{4mT}\right)}\right) &= P_{X\in\{0,1\}^N}\left(Z(X) > \frac{2mT}{w\left(\frac{\varepsilon^2\delta}{4mT}\right)}\frac{2}{\delta}\right) \\ &\leq P_{X\in\{0,1\}^N}\left(Z(X) > \frac{2E[Z]}{\delta}\right) && \text{aplicando (4.26)} \\ &\leq \frac{E[Z]}{\frac{2E[Z]}{\delta}} = \frac{\delta}{2}, && \text{por Markov (4.14)} \end{aligned}$$

quedando demostrado (4.18). Además, la cota anterior se puede expresar como un polinomio en función de T, $1/\varepsilon$ y $1/\delta$:

$$P_{X\in\{0,1\}^N}\left(Z(X) > \frac{4mT}{\delta\ w\left(\frac{\varepsilon^2\delta}{4mT}\right)}\right) = P_{X\in\{0,1\}^N}\left(Z(X) > \text{poly}\left(T, \frac{1}{\varepsilon}, \frac{1}{\delta}\right)\right). \tag{4.27}$$

4. Los puntos previos completan la demostración del teorema.

A partir de (4.17), (4.18) y (4.27) concluimos que la probabilidad de que el algoritmo falle (que genere una salida con error superior a ε, o que no finalice en la cota de poly$(T, 1/\varepsilon, 1/\delta)$ iteraciones) es, como mucho, $\delta/2 + \delta/2 = \delta$.

Las probabilidades obtenidas en los apartados anteriores son respecto a X, y X es una variable aleatoria con distribución uniforme, así que la aproximación con error máximo ε se obtiene sobre una fracción $1-\delta$ de los inputs. Además, por (4.27), el número de iteraciones es poly$(T, 1/\varepsilon, 1/\delta)$, que se corresponde también con la profundidad de esta rama. □

Es conveniente remarcar que el algoritmo descrito en la demostración no es computable en un tiempo polinomial:

- En cada iteración se selecciona una variable con alta influencia. Como hay $N = m2^n$ variables, la obtención de esta variable involucraría un número de instrucciones no polinomial respecto a n.
- En la primera iteración del algoritmo se evalúa $E_{Y\in\{0,1\}^N}[p_0(Y)]$. Sin embargo, el valor de esta esperanza requeriría calcular la evaluación del polinomio para todos los distintos 2^N inputs de p_0, cosa que tampoco puede computarse con complejidad polinomial.

No obstante, este hecho no invalida las conclusiones del teorema, ya que el algoritmo descrito no se corresponde con el algoritmo clásico C referenciado en la Conjetura 3.1. El algoritmo C es un árbol de decisión determinista (como los descritos en la Sección 2.1.4), y el objetivo de la demostración (y del algoritmo descrito en ella) es constatar que este árbol podría construirse

(aunque sea en un tiempo no necesariamente polinomial), y que por lo tanto existe.

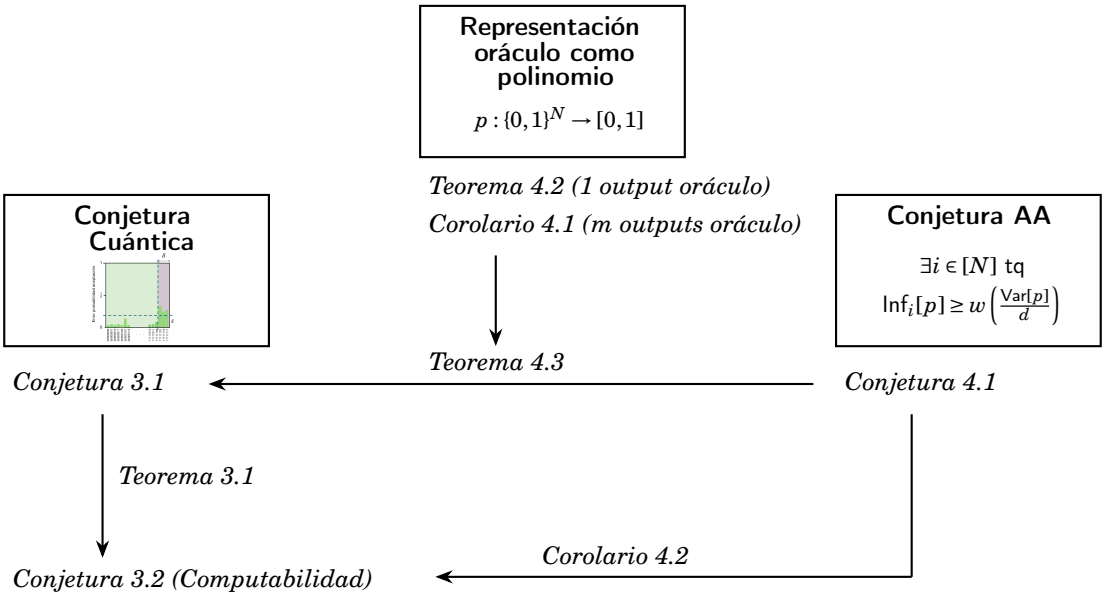
Finalmente, nos queda relacionar la Conjetura AA con la Conjeturas 3.2:

Corolario 4.2. *La Conjetura 4.1 (Conjetura AA) implica la Conjetura 3.2 (Computabilidad).*

Demostración. Es inmediata a partir de los Teoremas 3.1, 4.3. □

4.3.3. Resumen de resultados

En las secciones anteriores hemos descrito diversas conjeturas y teoremas que las relacionan. A continuación incluimos un diagrama que sintetiza lo estudiado y la relación entre resultados.



4.3.4. Consecuencias

Aaronson publicó en [3] una lista de las siete áreas que consideraba de mayor interés desde el punto de vista de la investigación sobre información cuántica. El primero de estos temas era la Quantum Query Complexity, y un punto troncal en éste es la investigación sobre esta conjetura. La conjetura también aparece en la colección de problemas abiertos recopilada por un conjunto de importantes autores en [44].

En caso de validarse, la Conjetura AA sería relevante para entender el contexto en el que pueden producirse escenarios de supremacía cuántica, y para identificar qué características deben tener los problemas que tiene sentido estudiar desde el punto de vista de la información cuántica.

Diversos matemáticos han señalado el hecho de que el resultado del paper de Aaronson y Ambainis [4] no hace más que desplazar la incertidumbre de la Conjetura Cuántica a la Conjetura

AA, en el sentido que ninguna de las dos está demostrada y que ambas parecen igualmente ”razonables”. Sin embargo, la Conjetura AA ha ido cobrando relevancia en la comunidad matemática durante los últimos 15 años y ha sido objeto de estudio por parte de múltiples expertos en el análisis de funciones y polinomios booleanos. Se han obtenido demostraciones de resultados parciales, e incluso un intento de demostración completa que finalmente resultó fallido [2] (estudiaremos estas cuestiones en la próxima sección).

Lo interesante de esta conjetura es que traslada un problema de computación cuántica, más difícil de objetivar en términos matematizables, a un supuesto expresado en términos analíticos y probabilísticos. Actualmente, el caso general sigue sin haberse demostrado, cosa que mantiene vigente el interés de múltiples investigadores para resolver el problema abierto.

CAPÍTULO 5

RESULTADOS RELACIONADOS CON LA CONJETURA AA

En este capítulo haremos una revisión de resultados en la teoría de funciones booleanas que han aportado información en el estudio de la Conjetura AA. En las primeras secciones revisaremos de forma detallada algunos de estos resultados y aportaremos demostración de cada uno de ellos. Al final del capítulo repasaremos la cronología de los distintos trabajos que se han publicado al respecto sin entrar en demasiados detalles, con el único propósito de obtener una visión general del estado del arte del estudio de la validez de esta conjetura.

Empezaremos asentando los conceptos y propiedades de la teoría de funciones booleanas que serán necesarios en siguientes secciones.

5.1. Ampliación de funciones booleanas

5.1.1. Dominio $\{\pm 1\}$

Hasta el momento hemos declarado las funciones booleanas con dominio $\{0,1\}$. La razón para haber seguido este criterio estaba en la aplicación que dábamos a estas funciones: el estudio de circuitos cuánticos y clásicos. Estos circuitos tienen conjuntos de bits como entrada y salida, que en teoría de computación se representan con valores 0 y 1.

Sin embargo, en la teoría de funciones booleanas suele utilizarse el dominio $\{-1,1\}$ (en adelante, denotaremos este conjunto como $\{\pm 1\}$). Como ya hemos comentado, el objetivo de las próximas secciones consiste en revisar diversos resultados de teoría de funciones, y lo haremos sin necesidad de conectar su estudio con la computación. Por esta razón, en adelante definiremos el conjunto booleano sobre los valores $\{\pm 1\}$ (o, en el caso de n dimensiones, sobre el *cubo discreto* $\{\pm 1\}^n$), y toda la teoría asociada se construirá sobre esta convención.

Para ello, primero necesitamos asegurarnos que los teoremas y conjeturas a los que hemos

llegado hasta ahora siguen siendo válidos sobre el nuevo dominio, y que los resultados que identifiquemos en adelante son aplicables sobre los de las secciones previas. Sea $n \in \mathbb{N}$, definimos la siguiente aplicación biyectiva

$$b : \{\pm 1\}^n \to \{0,1\}^n, \quad b(x_1,...,x_n) = \left(\frac{1-x_1}{2},...,\frac{1-x_n}{2}\right). \tag{5.1}$$

Esta aplicación transforma coordenadas booleanas de uno a otro convenio, $1 \to 0$ y $-1 \to 1$. Por ejemplo, para $n = 2$, $b(1,-1) = (0,1)$.

Dada una función $f : \{0,1\}^n \to \mathbb{R}$, podemos definir una aplicación equivalente en el nuevo dominio $f \circ b : \{\pm 1\}^n \to \mathbb{R}$. De forma análoga, toda función $g : \{\pm 1\}^n \to \mathbb{R}$ tiene una equivalente $g \circ b^{-1} : \{0,1\}^n \to \mathbb{R}$. Como tanto b como b^{-1} son lineales, en caso de que f, g sean polinomios multilineales, también lo serán $f \circ b$, $g \circ b^{-1}$. Además, si un monomio del polinomio origen f (o g) es de grado d (contiene d variables), la suma de monomios correspondiente a la imagen por $f \circ b$ (o $g \circ b^{-1}$) contendrá también d variables y por lo tanto seguirá teniendo grado d. Así pues, las aplicaciones $f \circ b$, $g \circ b^{-1}$, tienen el mismo grado que f y g respectivamente.

La invarianza respecto a linealidad y grado de esta transformación justifica que los resultados del capítulo anterior sigan siendo aplicables bajo el nuevo convenio. No obstante, volveremos a enunciar todos los conceptos de la Sección 4.1 con este nuevo dominio para desarrollar las próximas secciones sobre una base coherente. Además, introduciremos algunos conceptos nuevos que necesitaremos más adelante.

5.1.2. Funciones booleanas

Definición 5.1. Un **polinomio multilineal real de n variables y grado d** es una aplicación $f : \{\pm 1\}^n \to \mathbb{R}$ de la forma

$$f(x) = \sum_{S \subset [n], |S| \leq d} \hat{f}(S) \prod_{i \in S} x_i$$

donde $x := (x_1,...,x_n) \in \{\pm 1\}^n$ y $\hat{f}(S) \in \mathbb{R}$ es el coeficiente correspondiente al monomio $\prod_{i \in S} x_i$ (que en adelante denotaremos $x^S := \prod_{k \in S} x_k$).

Teorema 5.1. *Toda función $f : \{\pm 1\}^n \to \mathbb{R}$ se puede expresar de forma única como polinomio multilineal,*

$$f(x) = \sum_{S \subset [n]} \hat{f}(S)\, x^S, \qquad \text{donde } \hat{f}(S) \in \mathbb{R}. \tag{5.2}$$

*Denominamos a esta expresión **expansión de Fourier** de f y al número $\hat{f}(S) \in \mathbb{R}$ **coeficiente de Fourier** de f en S. Colectivamente, los coeficientes se denominan espectro de Fourier de f.*

La demostración de este teorema puede encontrarse en [34, pp.20-22]. Además, observamos que de este resultado se puede deducir el Teorema 4.1 que había quedado pendiente de demostración, ya que la función descrita en ese teorema es expresable como composición $f \circ b^{-1}$ según descrito en la sección anterior.

Sea $f : \{\pm 1\}^n \to \mathbb{R}$, $p \in \mathbb{N}$, denotaremos norma L_p y norma L_∞ de f a

$$\|f\|_p := E_{x\in\{\pm1\}^n}\left[|f(x)|^p\right]^{1/p}, \qquad \|f\|_\infty := \sup_{x\in\{\pm1\}^n} |f(x)| \tag{5.3}$$

Conviene destacar que la norma L_p que definimos para funciones booleanas difiere en un escalado de $1/2^n$ respecto a la definición tradicional $\left(\sum_{x\in\{\pm1\}^n} |f(x)|^p\right)^{1/p}$.

Para todo polinomio multilineal $f : \{\pm 1\}^n \to \mathbb{R}$, $f(x) = \sum_{S\subset[n]} \hat{f}(S)\, x^S$ de grado d, existe una función $F : \mathbb{R}^n \to \mathbb{R}$ que denominamos **extensión multilineal** de f, también de grado d, con la siguiente expresión

$$F(x) = \sum_{S\subset[n]} \hat{F}(S)\, x^S, \tag{5.4}$$

donde $\hat{F}(S) = \hat{f}(S)$ en todo $S \subset [n]$. A partir de su construcción, es inmediato que la restricción $F_{|\{\pm1\}^n}$ coincide con f.

5.1.2.1. Desacoplaje

Una técnica utilizada en algunas aplicaciones del análisis, entre ellas la teoría de computación, consiste en "desacoplar" funciones de forma que se introduce cierta independencia entre sus variables respecto a la función original. Veamos una forma de aplicar desacoplaje sobre funciones booleanas:

Definición 5.2. [36, p.2] Sea $f : \{\pm 1\}^n \to \mathbb{R}$ una aplicación booleana, su versión **desacoplada de un bloque** (one-block decoupled) es la siguiente aplicación :

$$\check{f} : \left(\{\pm1\}^n\right)^2 \to \mathbb{R}, \qquad \check{f}(y,z) = \sum_{S\subset[n]} \hat{f}(S) \sum_{k\in S} z^{S\setminus\{k\}} y^{\{k\}} \tag{5.5}$$

donde $y, z \in \{\pm 1\}^n$.

Nótese que esta definición excluye el término independiente de f en la representación de $\check{f}$ (es decir, el coeficiente de Fourier correspondiente a $\hat{f}(\emptyset)$).

Veamos un ejemplo: sea la función

$$f : \{\pm1\}^3 \to \mathbb{R}, \qquad f(x_1,x_2,x_3) = x_1x_2x_3 + 2x_2x_3 + x_1 - 2, \tag{5.6}$$

su función desacoplada de un bloque sería

$$\check{f} : \{\pm1\}^6 \to \mathbb{R}, \qquad \check{f}(y_1,y_2,y_3,z_1,z_2,z_3) = y_1z_2z_3 + z_1y_2z_3 + z_1z_2y_3 + 2y_2z_3 + 2z_2y_3 + y_1$$

5.1.2.2. Conceptos estadísticos

Sea $f : \{\pm 1\}^n \to \mathbb{R}$ un polinomio multilineal real para algún $n \in \mathbb{N}$, y $x \in \{\pm 1\}^n$ una variable aleatoria con distribución uniforme en $\{\pm 1\}^n$, definimos los siguientes conceptos estadísticos análogos a los descritos en la Sección 4.3.1:

La **varianza** de f es

$$\mathrm{Var}[f] := E_{x\in\{\pm 1\}^n}\left[(f(x) - E[f])^2\right]. \tag{5.7}$$

La **influencia** de la variable $i \in [n]$ en f es

$$\mathrm{Inf}_i[f] := E_{x\in\{\pm 1\}^n}\left[\left(f(x) - f(x^{(i)})\right)^2\right], \tag{5.8}$$

siendo $x^{(i)}$ el vector resultante de invertir la i-ésima coordenada en x.

También definimos la **máxima influencia** de f como

$$\mathrm{maxInf}[f] := \max_{i\in[n]}\{\mathrm{Inf}_i[f]\}, \tag{5.9}$$

y la **influencia total** de f

$$\mathrm{Inf}[f] := \sum_{i\in[n]} \mathrm{Inf}_i[f]. \tag{5.10}$$

Veamos algunos resultados que describen cómo representar la varianza y la influencia a partir de los coeficientes de Fourier de la función (las demostraciones de las Proposiciones y Teoremas pueden encontrarse en las referencias aportadas):

Proposición 5.1. *[34, p.27] Sea $f : \{\pm 1\}^n \to \mathbb{R}$,*

$$\mathrm{Var}[f] = E[f^2] - E[f]^2 = \sum_{S\neq\emptyset} \hat{f}(S)^2 \tag{5.11}$$

A partir de la proposición anterior obtenemos otra forma alternativa de expresar la varianza que utilizaremos más adelante:

$$\begin{aligned}
\mathrm{Var}[f] &= E[f^2] - E[f]^2 \\
&= E_{x\in\{\pm 1\}^n}[f(x)^2] - E_{x\in\{\pm 1\}^n}[f(x)]^2 \\
&= \left(\frac{1}{2}\,E_{x\in\{\pm 1\}^n}[f(x)^2] + \frac{1}{2}\,E_{y\in\{\pm 1\}^n}[f(y)^2]\right) - \left(E_{x\in\{\pm 1\}^n}[f(x)]\,E_{y\in\{\pm 1\}^n}[f(y)]\right) \\
&= \frac{1}{2}\left(E_{x\in\{\pm 1\}^n}[f(x)^2] + E_{y\in\{\pm 1\}^n}[f(y)^2] - 2E_{x\in\{\pm 1\}^n}[f(x)]\,E_{y\in\{\pm 1\}^n}[f(y)]\right) \\
&= \frac{1}{2}\,E_{x,y\in\{\pm 1\}^n}[f(x)^2 + f(y)^2 - 2f(x)f(y)] \\
&= \frac{1}{2}\,E_{x,y\in\{\pm 1\}^n}[(f(x) - f(y))^2]
\end{aligned} \tag{5.12}$$

El siguiente teorema determina una expresión análoga a (5.11) para la influencia:

Teorema 5.2. *[34, p.48] Sea* $f : \{\pm 1\}^n \to \mathbb{R}$, $i \in [n]$,

$$\mathrm{Inf}_i[f] = \sum_{S \ni i} \hat{f}(S)^2 \tag{5.13}$$

A continuación expondremos algunas consecuencias de los resultados anteriores que relacionan varianza e influencia. En todos los casos, partimos de una función $f : \{\pm 1\}^n \to \mathbb{R}$:

$$\mathrm{Var}[f] = \sum_{S \neq \emptyset} \hat{f}(S)^2 \geq \sum_{S \ni i} \hat{f}(S)^2 = \mathrm{Inf}_i[f], \qquad \text{para todo } i \in [n] \tag{5.14}$$

$$\mathrm{Var}[f] = \sum_{S \neq \emptyset} \hat{f}(S)^2 \leq \sum_{i \in [n]} \sum_{S \ni i} \hat{f}(S)^2 = \sum_{i \in [n]} \mathrm{Inf}_i[f] = \mathrm{Inf}[f] \tag{5.15}$$

Veamos también el efecto de multiplicar una función por una constante $C \in \mathbb{R}$:

$$\mathrm{Inf}_i[Cf] = \sum_{S \ni i} (C\hat{f}(S))^2 = C^2 \sum_{S \ni i} \hat{f}(S)^2 = C^2 \mathrm{Inf}_i[f] \qquad \text{para todo } i \in [n] \tag{5.16}$$

$$\mathrm{Var}_i[Cf] = \sum_{S \neq \emptyset} (C\hat{f}(S))^2 = C^2 \sum_{S \neq \emptyset} \hat{f}(S)^2 = C^2 \mathrm{Var}[f] \tag{5.17}$$

5.1.2.3. Hipercontractividad

Definimos el *operador de ruido* para $\rho \in [0,1]$ como el operador lineal T_ρ que actúa en $f : \{\pm 1\}^n \to \mathbb{R}$ de la siguiente forma

$$T_\rho f(x) := \sum_{S \subset [n]} \hat{f}(S) \rho^{|S|} x^S, \quad x \in \{\pm 1\}^n \tag{5.18}$$

El siguiente teorema nos proporciona un mecanismo para relacionar dos normas L_p (ver (5.3)) de una función $f : \{\pm 1\}^n \to \mathbb{R}$ mediante el operador ruido [34, p.259].

Teorema 5.3. *Sea* $1 < p \leq q < \infty$, $\rho \leq \sqrt{\frac{p-1}{q-1}}$,

$$\|T_\rho f\|_q \leq \|f\|_p \tag{5.19}$$

Y para el caso $p = 1$ existe el siguiente resultado (ver [34, p.262]):

$$\|f\|_2 \leq e^d \|f\|_1 \tag{5.20}$$

donde d es el grado del polinomio multilineal asociado a la función f.

5.2. Conjetura válida para funciones booleanas

En 2005, tres años antes de proponerse la Conjetura AA, O'Donnell, Saks, Schramm y Servedio demostraron un teorema que relaciona la varianza de un árbol de decisión con las influencias de sus variables [35]. Una consecuencia de este teorema en el caso particular de funciones booleanas es que existe una variable cuya influencia es superior o igual a la varianza. Esto implica la validez de la Conjetura AA restringida a funciones booleanas aunque, a diferencia del enunciado original de la conjetura, en este caso la relación entre varianza e influencia máxima no depende del grado d de la expansión de Fourier de la función.

Concretamente, el enunciado del caso particular que estudiaremos en esta sección es el siguiente:

Proposición 5.2. *Sea* $f : \{\pm 1\}^n \to \{\pm 1\}$ *una función booleana de grado* d*, existe* $i \in [n]$ *tal que* $\mathrm{Inf}_i[f] \geq (\mathrm{Var}[f]/d)^4$.

A continuación presentamos una demostración basada en ideas de [35, Corolario 1.2, Teorema 1.1] y [33].

Demostración. Empezamos declarando la siguiente aplicación a partir de la función booleana f del enunciado:

$$d_f : \{\pm 1\}^n \times \{\pm 1\}^n \to \mathbb{R} \qquad d_f(x,y) = \left(\frac{f(x)-f(y)}{2}\right)^2, \qquad \text{para todo } x,y \in \{\pm 1\}^n \tag{5.21}$$

Sean $x,y,z \in \{\pm 1\}^n$, los valores que toma esta aplicación son $d_f(x,y) = 0$ si $f(x) = f(y)$, y $d_f(x,y) = 1$ si $f(x) \neq f(y)$. Veamos que d_f es una *semimétrica*:

a. d_f es positiva, ya que $d_f(x,y) = ((f(x)-f(y))/2)^2 \geq 0$. Además, $d_f(x,x) = ((f(x)-f(x))/2)^2 = 0$. No necesariamente $d_f(x,y) = 0$ implica que $x = y$, ya que f no es necesariamente inyectiva, por lo que d_f no sería métrica sino semimétrica.

b. d_f es simétrica, al ser $d_f(x,y) = ((f(x)-f(y))/2)^2 = ((f(y)-f(x))/2)^2 = d_f(y,x)$.

c. d_f cumple la desigualdad triangular. Para verlo, estudiamos las dos posibles situaciones siguientes: (i) si $f(x) = f(z)$, verificamos que $d_f(x,y)+d_f(y,z) \geq 0 = d_f(x,z)$; (ii) si $f(x) \neq f(z)$, entonces se da exactamente una de estas dos circunstancias: o $f(y) = f(x)$, o bien $f(y) = f(z)$, y en ambos casos $d_f(x,y)+d_f(y,z) = 1 = d_f(x,z)$.

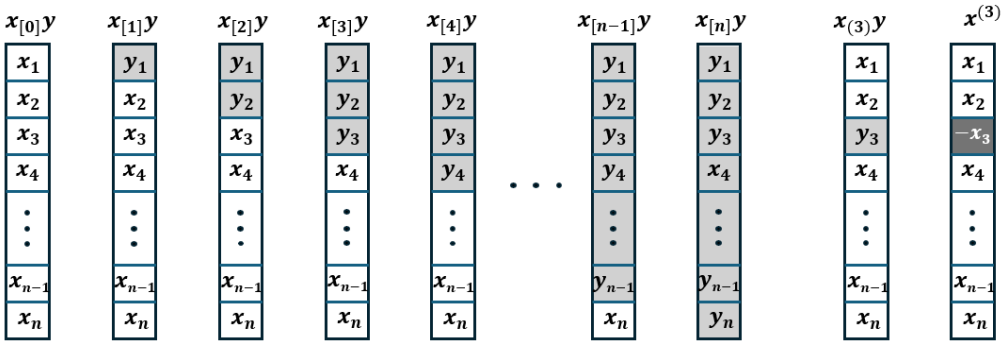
A partir de un par de elementos $x,y \in \{\pm 1\}^n$ construiremos diversos conjuntos de elementos resultantes de combinar coordenadas de x y de y.

En primer lugar definiremos el conjunto $\{x_{[j]}y := x^{[n]\setminus[j]}y^{[j]}\}_{j\in\{0,\cdots,n\}}$, [1] cuyos elementos van sustituyendo progresivamente coordenadas de x por coordenadas de y. Los elementos correspondientes a los índices $0, n$ son

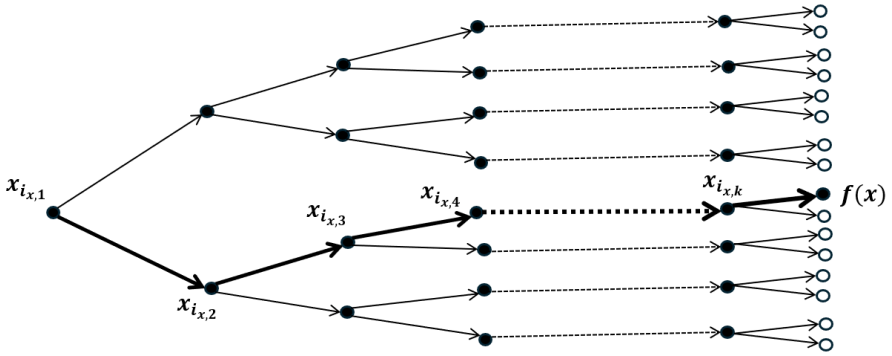
$$x_{[0]}y = x, \qquad x_{[n]}y = y. \tag{5.22}$$

Denotamos también $\{x_{(j)}y := x^{[n]\setminus\{j\}}y^{\{j\}}\}_{j\in\{0,\cdots,n\}}$ [2] para representar el resultado de sustituir la j-ésima coordenada de x por la correspondiente de y. Recordemos también la notación $x^{(j)}$ para la inversión de la j-ésima coordenada de x.

A continuación representamos un diagrama con algunos ejemplos de los elementos correspondientes a estas notaciones:



Vamos a definir conceptos análogos a los anteriores en el contexto de un árbol de decisión T que evalúa f. Supondremos que T es homogéneo (en caso de no serlo, podemos completar las ramas de longitud inferior a k añadiendo variables arbitrarias) y que tiene profundidad k. Cada input $x \in \{\pm 1\}^n$ de la función tiene asociada una rama en T que determina una secuencia de variables de índices $\{i_{x,1}, i_{x,2}, \cdots, i_{x,k}\}$ que son las que se recorren en el árbol de decisión para evaluar $f(x)$.



[1] En caso $j = 0$ identificamos [j] con el conjunto vacío

[2] En caso de contrastarlo con la demostración de [35], hay que tener en cuenta que ahí se usa esta misma terminología para conceptos distintos a los aquí descritos.

A partir de esta rama denotaremos nuevos conjuntos de elementos $\{u[t]_{x,y}\}_{t\in\{0,\cdots,k\}}$ que contienen la coordenada y en las posiciones $i_{x,1}, i_{x,2}\cdots i_{x,t}$ y la de x en el resto de coordenadas. A diferencia de lo descrito en (5.23), en esta ocasión no necesariamente $u[k]_{x,y} = y$. Sin embargo, sí que hay correspondencia entre la evaluación por f de los elementos extremos de este conjunto y la de x, y:

$$f(u[0]_{x,y}) = f(x), \qquad f(u[k]_{x,y}) = f(y). \tag{5.23}$$

Análogamente, denotamos también $\{u(t)_{x,y} := x_{(i_{x,t})}y\}_{t\in\{0,\cdots,k\}}$.

Veamos algunas propiedades de los elementos descritos hasta ahora. Aplicando la desigualdad triangular sobre $d_f(x,y)$, obtenemos

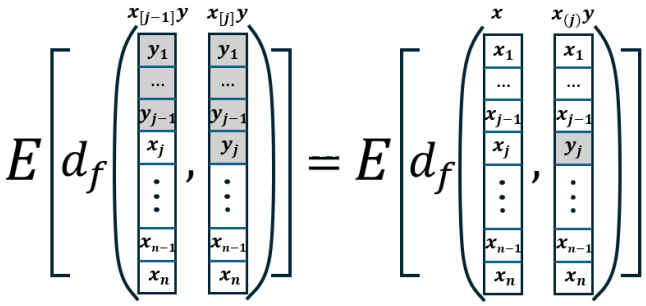
$$\begin{aligned} d_f(x,y) &= d_f\left(u[0]_{x,y}, u[k]_{x,y}\right) && \text{por (5.23)} \\ &\leq \sum_{t\in[k]} d_f\left(u[t-1]_{x,y}, u[t]_{x,y}\right) && \text{por la desigualdad triangular} \end{aligned} \tag{5.24}$$

Además observamos que para todo $j \in [n]$

$$E_{x,y\in\{\pm1\}^n}\left[d_f\left(x_{[j-1]}y, x_{[j]}y\right)\right] = E_{x,y\in\{\pm1\}^n}\left[d_f\left(x, x_{(j)}y\right)\right] \tag{5.25}$$

Esto es debido a que, tal como se aprecia en el siguiente diagrama, los pares de elementos $\left(x_{[j-1]}y, x_{[j]}y\right)$ y $\left(x, x_{(j)}y\right)$ coinciden en todas las coordenadas excepto en la j-ésima, y como para todo $i \in [n]\setminus\{j\}$ las variables aleatorias x_i, y_i son independientes y ambas tienen distribución uniforme, el hecho de que aparezca una u otra en los pares de elementos comparados no afecta al cálculo del valor esperado.

Representamos (5.25) en el siguiente diagrama:



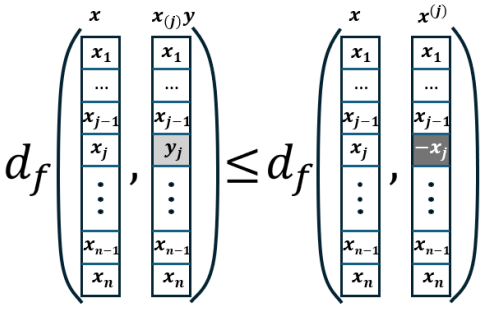
Razonando de la misma forma sobre las coordenadas $i_{x,1}, i_{x,2}, \cdots, i_{x,j}$ en vez de $1, 2, \cdots, j$, obtenemos la siguiente propiedad análoga a (5.25):

$$E_{x,y\in\{\pm1\}^n}\left[d_f\left(u[t-1]_{x,y}, u[t]_{x,y}\right)\right] = E_{x,y\in\{\pm1\}^n}\left[d_f\left(u(t-1)_{x,y}, u(t)_{x,y}\right)\right] \tag{5.26}$$

Además, para todo $j \in [n]$

$$d_f\left(x_{(j-1)}y, x_{(j)}y\right) \leq d_f\left(x, x^{(j)}\right), \tag{5.27}$$

puesto que $x^{(j)}$ invierte siempre la j-ésima coordenada respecto a x, mientras que $x_{(j)}y$ sólo lo hace si y difiere de x en esta coordenada (ver siguiente diagrama).



Una vez más, la desigualdad (5.27) tiene la siguiente correspondencia en el recorrido de variables en la rama de T:

$$d_f\left(u(t-1)_{x,y}, u(t)_{x,y}\right) \leq d_f\left(x, x^{(i_{x,t})}\right), \tag{5.28}$$

A partir de los resultados anteriores:

$$\begin{aligned}
\mathrm{Var}[f] &= \frac{1}{2} E_{x,y\in\{\pm1\}^n}\left[(f(x)-f(y))^2\right] && \text{por la propiedad (5.12)}\\
&= 2\, E_{x,y\in\{\pm1\}^n}\left[\left(\frac{f(x)-f(y)}{2}\right)^2\right]\\
&= 2E_{x,y\in\{\pm1\}^n}\left[d_f(x,y)\right] && \text{por definición (5.21)}\\
&\leq 2\sum_{t\in[k]} E_{x,y\in\{\pm1\}^n}\left[d_f(u[t-1]_{x,y}, u[t]_{x,y})\right] && \text{por (5.24)}\\
&= 2\sum_{t\in[k]} E_{x,y\in\{\pm1\}^n}\left[d_f(u(t-1)_{x,y}, u(t)_{x,y})\right] && \text{por (5.26)}\\
&\leq 2\sum_{t\in[k]} E_{x\in\{\pm1\}^n}\left[d_f\left(x, x^{(i_x,t)}\right)\right] && \text{por (5.28)} \qquad (5.29)\\
&= 2\sum_{t\in[k]} E_{x\in\{\pm1\}^n}\left[\left(\frac{f(x)-f(x^{(i_x,t))})}{2}\right)^2\right] && \text{por definición (5.21)}\\
&= \frac{1}{2}\sum_{t\in[k]} \mathrm{Inf}_{i_x,t}[f] && \text{por definición de influencia (5.28)}\\
&\leq \frac{1}{2}\sum_{t\in[k]} \mathrm{maxInf}[f]\\
&= \frac{k}{2}\,\mathrm{maxInf}[f]
\end{aligned}$$

La cota que hemos obtenido en (5.29) relaciona la influencia máxima con la varianza y la profundidad del árbol de decisión. Sin embargo, tanto la Conjetura AA como el enunciado de la proposición que estamos demostrando incluyen en su expresión el grado del polinomio en vez de la profundidad del árbol, por lo que nos será útil encontrar alguna relación entre ambos factores.

Con este propósito referenciaremos el siguiente resultado de Buhrman y De Wolf [12, p.35, Thm 12] según el cual, si d es el grado de un polinomio multilineal y k es la profundidad mínima del árbol de decisión que lo evalúe, entonces

$$k \leq 2d^4 \tag{5.30}$$

Por lo tanto,

$$\begin{aligned} \max\mathrm{Inf}[f] &\geq 2\frac{\mathrm{Var}[f]}{k} \\ &\geq 2\frac{\mathrm{Var}[f]}{2d^4} && \text{por(5.30)} \\ &\geq \left(\frac{\mathrm{Var}[f]}{d}\right)^4 && \text{ya que } \mathrm{Var}[f] \leq 1 \end{aligned} \tag{5.31}$$

□

5.3. Versión exponencial de la conjetura

Igual que sucedió con el artículo de O'Donnell et al. (Sección 5.1), también el resultado que expondremos aquí es previo a la Conjetura AA. En 2006, Dinur y Friedgut demostraron en [18] un teorema que se corresponde con una versión más relajada de la conjetura. Recordemos que ésta determina que, para toda función booleana real que toma valor en $[0,1]$, existe una variable cuya influencia es superior a una cota dependiente polinomialmente de la varianza y del inverso del grado d de la función. El resultado descrito por Dinur y Friedgut es equivalente a ese enunciado modificando la referencia al grado d del polinomio f por $\exp(d)$.

Veamos su enunciado:

Teorema 5.4. *Sea $f : \{\pm 1\}^n \to [0,1]$ un polinomio multilineal real de grado d, existe $i \in [n]$ tal que*

$$\mathrm{Inf}_i[f] \geq \left(\frac{\mathrm{Var}[f]}{O(d^2)\, e^{d-1}}\right)^2 \tag{5.32}$$

Hay diversas demostraciones publicadas de este teorema. Algunas de ellas son: la propia demostración elaborada por Dinur y Friedgut [18]; como corolario de un resultado obtenido por Defant, Mastylo y Pérez [14]; o como corolario de otro resultado de O'Donnell y Zhao [36]. En este trabajo describiremos una demostración basada en ideas de Defant, Mastylo y Pérez, aunque nos apoyaremos en dos lemas de O'Donnell y Zhao.

Empezaremos con el lema principal:

Lema 5.1. *[36, p. 10 Lema 4.1] Sea $f : \{\pm 1\}^n \to \mathbb{R}$ un polinomio multilineal de grado d, y $F : \mathbb{R}^n \to \mathbb{R}$ su extension multilineal (ver (5.4))*

$$F(x) = \sum_{S \subset [n]} \hat{F}(S)\, x^S, \tag{5.33}$$

también de grado d, existen $m \leq 2(d+1)$ y $\alpha, \beta, c \in \mathbb{R}^m$ tales que, para todo $y, z \in \{\pm 1\}^n$:

$$\check{f}(y,z) = \sum_{k \in [m]} c_k F(\alpha_k y + \beta_k z), \tag{5.34}$$

donde $\check{f} : (\{\pm 1\}^n)^2 \to \mathbb{R}$ *es la versión desacoplada de un bloque de* f *(ver (5.5)), y además*

$$\|c\|_1 \leq O(d^2), \tag{5.35}$$

$$|\alpha_k| + |\beta_k| = 1, \qquad \text{para todo } k \in [m]. \tag{5.36}$$

En [36, pp. 12–16] se describe la línea de la demostración para un resultado más general, identificando las ideas pero omitiendo bastantes cálculos y detalles en la parte correspondiente a este lema. La demostración se estructura en dos supuestos: caso homogéneo y caso general (no necesariamente homogéneo). Aquí desarrollaremos el caso homogéneo de forma detallada, pero abordaremos el caso general de una forma similar a [36]: describiendo los conceptos en los que se basa la demostración, pero omitiendo el detalle de sus pasos. El propósito es evitar dedicar un espacio excesivo a este lema, aunque igualmente ha acabado resultando la demostración más extensa de todo el trabajo.

Demostración. En el supuesto de F homogeneo (es decir, cuando todos los coeficientes no nulos $\hat{F}(S)$ cumplen $|S| = d$), tomaremos $m = d+1$ y construiremos tres vectores $\alpha, \beta, c \in \mathbb{R}^m$ tales que, para todo $t \in \{0, \cdots, d\}$,

$$\sum_{k \in [d+1]} c_k \alpha_k^{d-t} \beta_k^t = \begin{cases} 1 & \text{si } t = d-1 \\ 0 & \text{en caso contrario.} \end{cases} \tag{5.37}$$

Veamos primero que esta condición implica la ecuación (5.34) del enunciado del teorema, ya que para todo $y, z \in \{\pm 1\}^n$:

(5.38)

$$\begin{aligned}
\sum_{k \in [d+1]} c_k F(\alpha_k y + \beta_k z) &= \sum_{k \in [d+1]} c_k \sum_{S \subset [n]} \hat{F}(S) \prod_{j \in S} (\alpha_k y_j + \beta_k z_j) && \text{aplicando (5.33)} \\
&= \sum_{S \subset [n]} \hat{F}(S) \sum_{k \in [d+1]} c_k \prod_{j \in S} (\alpha_k y_j + \beta_k z_j) && \text{reordenando términos} \\
&= \sum_{S \subset [n]} \hat{F}(S) \sum_{j \in S} y^{\{j\}} z^{S \setminus \{j\}} && \text{simplificación al aplicar (5.37)} \\
&= \sum_{S \subset [n]} \hat{f}(S) \sum_{j \in S} y^{\{j\}} z^{S \setminus \{j\}} && \text{ya que } \hat{F}(S) = \hat{f}(S) \text{ para todo } S \subset n \\
&= \check{f}(y, z)
\end{aligned}$$

Proseguimos estudiando cómo deberían ser α, β, c para que cumplan (5.37). Sea el conjunto de índices $I = \{1, 2, \cdots, d, \frac{1}{2}\}$ y $k \in I$, definimos los siguientes valores para α_k, β_k y denotamos $\Delta = (\Delta_j)_{j \in I} \in \mathbb{R}^m$ de la siguiente forma:

$$\alpha_k = \frac{k^2}{d^2 + k^2}, \qquad \beta_k = \frac{d^2}{d^2 + k^2}, \qquad \Delta_k = \frac{\beta_k}{\alpha_k} = \frac{d^2}{k^2}, \qquad \text{para toda } k \in I. \tag{5.39}$$

Con esta asignación se verifica de forma inmediata la condición (5.36) del enunciado del lema, según la cual $|\alpha_k| + |\beta_k| = 1$. El objetivo ahora es obtener una expresión de c compatible con las

ecuaciones (5.34), (5.35) dados los valores α, β que hemos elegido. Para ello definimos, a partir de los vectores α, Δ, las matrices de Vandermonde V y diagonal A siguientes:

$$V = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ \Delta_1 & \Delta_2 & \cdots & \Delta_d & \Delta_{\frac{1}{2}} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \Delta_1^{d-1} & \Delta_2^{d-1} & \cdots & \Delta_d^{d-1} & \Delta_{\frac{1}{2}}^{d-1} \\ \Delta_1^{d} & \Delta_2^{d} & \cdots & \Delta_d^{d} & \Delta_{\frac{1}{2}}^{d} \end{pmatrix} \qquad A = \begin{pmatrix} \alpha_1^d & 0 & \cdots & 0 & 0 \\ 0 & \alpha_2^d & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \alpha_d^d & 0 \\ 0 & 0 & \cdots & 0 & \alpha_{\frac{1}{2}}^d \end{pmatrix} \tag{5.40}$$

Si denotamos $e_d = \left(\delta_k^d\right)_{k\in I} \in \{0,1\}^m$ al vector cuyas coordenadas son 0 excepto en la d-ésima posición, en la que toma valor 1, y estudiamos la siguiente ecuación,

$$VAc = e_d, \tag{5.41}$$

verificamos que ésta generará $m = d+1$ ecuaciones, una para cada $t \in \{0, \cdots, d\}$, del tipo

$$\sum_{k\in I} \alpha_k^d \Delta_k^t c_k = \sum_{k\in I} \alpha_k^d \frac{\beta_k^t}{\alpha_k^t} c_k = \sum_{k\in I} c_k \alpha_k^{d-t} \beta_k^t = \begin{cases} 1 & \text{si } t = d-1 \\ 0 & \text{en caso contrario} \end{cases}$$

así que de la ecuación (5.41) se deriva (5.37).

Al ser V la matriz de Vandermonde con factores Δ_k distintos para cada $k \in I$, V es invertible (ver [22, pp.95–96]). Así pues, como A es diagonal y por lo tanto también invertible, (5.41) equivale a la siguiente expresión con la que calculamos el valor de c:

$$c = A^{-1}V^{-1}e_d \tag{5.42}$$

Partiendo del procedimiento para evaluar V^{-1} para matrices de Vandermonde descrito en [22, pp.95–96], llegamos a la siguiente expresión para la k-ésima coordenada de c, siendo $k \in I$:

$$c_k = \left(A^{-1}V^{-1}e_d\right)_k = \frac{1}{\alpha_k^d} \cdot \frac{\Delta_k - \sum_{j\in I} \Delta_j}{\prod\limits_{j\in I, j\neq k} (\Delta_k - \Delta_j)} \tag{5.43}$$

Sólo nos queda calcular una cota superior de $\|c\|_1$ con el objetivo de comprobar la validez de (5.35). Para ello revisaremos individualmente cada uno de los coeficientes $|c_k|$ para $k \in I$. Estudiaremos por separado los subíndices $k \in [d]$ y $k = 1/2$, y finalmente combinaremos las cotas obtenidas en cada caso para determinar una cota global de $\|c\|_1$.

1. Caso $k \in [d]$:

Desarrollamos a partir de la expresión descrita en (5.43):

$$
\begin{aligned}
|c_k| &= \left| \frac{1}{\alpha_k^d} \cdot \frac{\Delta_k - \sum_{j\in I} \Delta_j}{\prod_{j\in I, j\neq \frac{1}{2}} (\Delta_k - \Delta_j)} \right| \\
&= \frac{1}{\left(\frac{k^2}{d^2+k^2}\right)^d} \cdot \frac{\left| \frac{d^2}{k^2} - \left(4d^2 + \sum_{j\in[d]} \frac{d^2}{j^2}\right)\right|}{\left|\left(\frac{d^2}{k^2} - 4d^2\right) \cdot \prod_{j\in[d], j\neq k} \left(\frac{d^2}{k^2} - \frac{d^2}{j^2}\right)\right|} && \text{sustituyendo} \\
&= \frac{1}{\left(\frac{k^2}{d^2+k^2}\right)^d} \cdot \frac{4d^2 - \frac{d^2}{k^2} + \sum_{j\in[d]} \frac{d^2}{j^2}}{\frac{d^{2d}}{k^{2d}} \left(4k^2-1\right) \cdot \prod_{j=1}^{k-1} \frac{k^2-j^2}{j^2} \cdot \prod_{j=k+1}^{d} \frac{j^2-k^2}{j^2}} && \text{eliminamos el valor absoluto} \\
&= \left(\frac{k^2(d^2+k^2)}{k^2 d^2}\right)^d \frac{4d^2 - \frac{d^2}{k^2} + \sum_{j\in[d]} \frac{d^2}{j^2}}{\left(4k^2-1\right) \cdot \prod_{j=1}^{k-1} \frac{k^2-j^2}{j^2} \cdot \prod_{j=k+1}^{d} \frac{j^2-k^2}{j^2}} && \text{agrupando exponenciales} \\
&= \left(\frac{d^2+k^2}{d^2}\right)^d \frac{d^2\left(4 - \frac{1}{k^2} + \sum_{j\in[d]} \frac{1}{j^2}\right)}{\left(4k^2-1\right) \cdot \prod_{j=1}^{k-1} \frac{k-j}{j} \cdot \prod_{j=k+1}^{d} \frac{j-k}{j} \cdot \prod_{j=1}^{k-1} \frac{k+j}{j} \cdot \prod_{j=k+1}^{d} \frac{j+k}{j}} \\
&= \left(\frac{d^2+k^2}{d^2}\right)^d \frac{d^2\left(4 - \frac{1}{k^2} + \sum_{j\in[d]} \frac{1}{j^2}\right)}{\left(4k^2-1\right) \cdot \frac{(k-1)!}{(k-1)!} \cdot \frac{(d-k)!}{\frac{d!}{k!}} \cdot \left(\prod_{j=1}^{d} \frac{k+j}{j}\right) \cdot \frac{k}{k+k}} && \text{expresamos como factorial} \\
&= \left(\frac{d^2+k^2}{d^2}\right)^d \frac{d^2\left(4 - \frac{1}{k^2} + \sum_{j\in[d]} \frac{1}{j^2}\right)}{\left(4k^2-1\right) \cdot \frac{(d-k)!}{\frac{d!}{k!}} \cdot \frac{\frac{(d+k)!}{k!}}{d!} \cdot \frac{1}{2}} && \text{simplificando} \\
&= \left(\frac{d^2+k^2}{d^2}\right)^d \frac{d^2\left(4 - \frac{1}{k^2} + \sum_{j\in[d]} \frac{1}{j^2}\right) 2d!d!}{\left(4k^2-1\right)(d-k)!\,(d+k)!} && \text{volviendo a simplificar}
\end{aligned}
\tag{5.44}
$$

Para estudiar una cota de la expresión anterior, diferenciaremos los casos $1 \leq k \leq \sqrt{d}$ y $\sqrt{d} \leq d$ [3].

1.1. Caso $1 \leq k \leq \sqrt{d}$

[3] Los valores posibles de k están restringidos al conjunto de los naturales, pero $\sqrt{d}$ puede ser un número no entero (incluso no racional). En todas la referencias a asignaciones $k = \sqrt{d}$,$k \leq \sqrt{d}$ o $k \geq \sqrt{d}$ daremos por hecho que, en el caso de que $\sqrt{d}$ no sea natural, tomamos el número natural inmediatamente superior (o inmediatamente inferior, según el contexto), sin entrar a especificar este hecho para no añadir más complejidad a la notación.

Seguimos a partir de la ecuación (5.44):

$$
\begin{aligned}
|c_k| &= \left(\frac{d^2+k^2}{d^2}\right)^d \frac{d^2\left(4-\frac{1}{k^2}+\sum_{j\in[d]}\frac{1}{j^2}\right)2d!d!}{\left(4k^2-1\right)(d-k)!\,(d+k)!} \\
&\le \left(\frac{d^2+k^2}{d^2}\right)^d \frac{2d^2\left(4-\frac{1}{k^2}+\sum_{j\in[d]}\frac{1}{j^2}\right)}{\left(4k^2-1\right)} && \text{ya que } \frac{d!d!}{(d-k)!\,(d+k)!}\le 1 \\
&\le \left(\frac{d^2+k^2}{d^2}\right)^d \frac{2d^2\left(4+\frac{\pi^2}{6}\right)}{k^2} && -\frac{1}{k^2}<0,\ \sum_{j\in[d]}\frac{1}{j^2}\le\sum_{j\in\mathbb{N}}\frac{1}{j^2}=\frac{\pi^2}{6} \\
&\le \left(\frac{d^2+k^2}{d^2}\right)^d \frac{12d^2}{k^2} && \frac{\pi^2}{6}<2 \\
&\le \left(1+\frac{1}{d}\right)^d 12\frac{d^2}{k^2} && \text{aplicando que } k\le\sqrt{d} \\
&\le 12e\frac{d^2}{k^2} && \text{por definición de e}
\end{aligned}
\tag{5.45}
$$

1.2. Caso $\sqrt{d}\le k\le d$

Nuestra intención es demostrar que la cota del apartado anterior también es válida en este intervalo. Es decir, que

$$
|c_k|\le 12e\frac{d^2}{k^2}, \qquad \text{para todo } \sqrt{d}\le k\le d. \tag{5.46}
$$

Esto quedaría demostrado por inducción si verificáramos que

$$
\frac{(k+1)^2\,|c_{k+1}|}{k^2\,|c_k|}\le 1, \tag{5.47}
$$

ya que por un lado en (5.45) hemos verificado que para $k=\sqrt{d}$, $|c_k|\le 12ed^2/k^2$, y por otro lado si esta cota fuera válida para determinado $k\in[\sqrt{d},d-1]$, entonces lo sería también para $k+1$, ya que

$$
\begin{aligned}
|c_{k+1}| &\le |c_k|\frac{k^2}{(k+1)^2} && \text{por (5.47)} \\
&\le 12e\frac{d^2}{k^2}\frac{k^2}{(k+1)^2} && \text{por hipótesis de inducción (5.46)} \\
&= 12e\frac{d^2}{(k+1)^2}
\end{aligned}
\tag{5.48}
$$

Así que el objetivo será validar (5.47). A partir de (5.44) obtenemos la expresión de ese cociente:

$$
\frac{(k+1)^2|c_{k+1}|}{k^2|c_k|}=\frac{(k+1)^2}{k^2}\left(\frac{d^2+(k+1)^2}{d^2+k^2}\right)^d\frac{4-\frac{1}{(k+1)^2}+\sum_{j\in[d]}\frac{1}{j^2}}{4-\frac{1}{k^2}+\sum_{j\in[d]}\frac{1}{j^2}}\frac{\left(4k^2-1\right)\,(d-k)!\,(d+k)!}{\left(4(k+1)^2-1\right)\,(d-k-1)!\,(d+k+1)!} \tag{5.49}
$$

Para simplificar esta ecuación veamos que

$$
\begin{aligned}
&\frac{(k+1)^2}{k^2}\cdot\frac{4-\frac{1}{(k+1)^2}+\sum_{j\in[d]}\frac{1}{j^2}}{4-\frac{1}{k^2}+\sum_{j\in[d]}\frac{1}{j^2}}\cdot\frac{\left(4k^2-1\right)}{\left(4(k+1)^2-1\right)}\\
&\qquad\le \frac{(k+1)^2}{k^2}\cdot\frac{4-\frac{1}{(k+1)^2}}{4-\frac{1}{k^2}}\cdot\frac{\left(4k^2-1\right)}{\left(4(k+1)^2-1\right)} && 0<a<b,c>0 \implies \frac{b}{a}\le\frac{a-c}{b-c}\\
&\qquad= \frac{(k+1)^2}{k^2}\cdot\frac{\frac{4(k+1)^2-1}{(k+1)^2}}{\frac{4k^2-1}{k^2}}\cdot\frac{\left(4k^2-1\right)}{\left(4(k+1)^2-1\right)} && \text{operando}\\
&\qquad= 1 && \text{se cancelan todos los términos}
\end{aligned}
\tag{5.50}
$$

Por lo tanto, podemos ajustar la cota de (5.49) prescindiendo de los términos anteriores:

$$
\begin{aligned}
\frac{(k+1)^2|c_{k+1}|}{k^2|c_k|} &\le \left(\frac{d^2+(k+1)^2}{d^2+k^2}\right)^d\frac{(d-k)!\,(d+k)!}{(d-k-1)!\,(d+k+1)!}\\
&\le \left(\frac{d^2+k^2+2k+1}{d^2+k^2}\right)^d\frac{(d-k)!\,(d+k)!}{(d-k-1)!\,(d+k+1)!}\\
&\le \left(1+\frac{1}{\frac{d^2+k^2}{2k+1}}\right)^d\frac{(d-k)}{(d+k+1)}\\
&\le e^{d\frac{2k+1}{d^2+k^2}}\frac{(d-k)}{(d+k+1)} && \text{por definición de } e
\end{aligned}
\tag{5.51}
$$

Para acotar esta expresión estudiaremos el crecimiento de las funciones $g,h:[\sqrt{d},d]\to\mathbb{R}$ siguientes

$$
g(x)=e^{d\frac{2x+1}{d^2+x^2}}\frac{(d-x)}{(d+x+1)}, \qquad h(x)=\log(g(x))=d\frac{2x+1}{d^2+x^2}+\log(d-x)-\log(d+x+1)
\tag{5.52}
$$

Empezaremos con el de h:

$$
\begin{aligned}
h'(x) &= d\frac{2(d^2+x^2)-(2x+1)2x}{(d^2+x^2)^2}-\frac{1}{d-x}-\frac{1}{d+x+1}\\
&\le d\frac{2(d^2+x^2)-(2x+1)2x}{(d^2+x^2)^2}-\frac{1}{d-x}-\frac{1}{d+x}\\
&= 2d\frac{-2d^5+d^4-4d^3x^2-d^2(x^2+x)-2dx^4+x^3+x^4}{(d^2+x^2)^2(d^2-x^2)} && \text{operando}\\
&= 2d\frac{-2d^5+d^4-4d^3x^2+(x^2-d^2)(x^2+x)-2dx^4}{(d^2+x^2)^2(d^2-x^2)} && \text{reordenando algunos términos}\\
&\le 2d\frac{-2d^5+d^4-4d^3x^2-2dx^4}{(d^2+x^2)^2(d^2-x^2)} && \text{ya que } x^2-d^2\le 0\\
&\le 0 && \text{ya que } -2d^5+d^4\le 0
\end{aligned}
\tag{5.53}
$$

Por lo tanto, h es no creciente en todo su dominio. Como el logaritmo es una función monótona creciente, g también es no creciente en este intervalo. Nos queda sólo comprobar que $g(\sqrt{d})\le 1$, en cuyo caso deduciríamos que el rango de g está incluido en $[0,1]$, y como el último término

de (5.51) se corresponde con la restricción de g a los naturales $\mathbb{N} \cap [\sqrt{d}, d]$, también quedaría confirmado que $|c_k| \leq 12ed^2/k^2$ (5.46).

Evaluamos $g(\sqrt{d})$:

$$g(\sqrt{d}) = e^{\frac{2\cdot\sqrt{d}+1}{d+1}} \frac{(d-\sqrt{d})}{(d+\sqrt{d}+1)} \tag{5.54}$$

No es inmediato estimar si este resultado es o no inferior a 1 para los distintos posibles valores del grado d, así que nuevamente recurriremos al estudio de una función que extiende la expresión anterior al dominio continuo

$$j : [1,\infty) \to \mathbb{R}, \qquad j(x) = e^{\frac{2\cdot\sqrt{x}+1}{x+1}} \frac{(x-\sqrt{x})}{(x+\sqrt{x}+1)} \tag{5.55}$$

Observamos que

$$\lim_{x\to\infty} j(x) = 1, \tag{5.56}$$

ya que el exponente $(2\sqrt{x}+1)/(x+1) \to 0$ y el cociente $(x-\sqrt{x})/(x+\sqrt{x}+1) \to 1$.

Además, si estudiamos la derivada de j

$$j'(x) = e^{\frac{2\sqrt{x}+1}{x+1}} \frac{5x^4 + 16x^{\frac{7}{2}} + 29x^3 + 32x^{\frac{5}{2}} + 22x^2 + 8x^{\frac{3}{2}} - x - 2\sqrt{x} - 1}{2(x+1)^2\left(x^{\frac{9}{2}} + 4x^4 + 10x^{\frac{7}{2}} + 16x^3 + 19x^{\frac{5}{2}} + 16x^2 + 10x^{\frac{3}{2}} + 4x + \sqrt{x}\right)}, \tag{5.57}$$

es fácil verificar si estudiamos sus coeficientes que $j'(x) > 0$ para todo $x > 1$. Así pues, j crece en $[1,\infty)$ y se aproxima asintóticamente a 1 cuando x incrementa, concluyendo que $j(x) \leq 1$ en todo su dominio. Por lo tanto, $g(\sqrt{d}) \leq 1$ para todo grado d, concluyendo la validez de (5.46), que era el objetivo de este apartado.

Revisaremos ahora la cota para el caso que nos queda por estudiar.

2. Caso $k = 1/2$:

Partimos de la expresión (5.43):

$$\begin{aligned}
\left|c_{\frac{1}{2}}\right| &= \left|\frac{1}{\alpha_{\frac{1}{2}}^d} \cdot \frac{\Delta_{\frac{1}{2}} - \sum_{j\in I}\Delta_j}{\prod_{j\in I, j\neq\frac{1}{2}}(\Delta_{\frac{1}{2}} - \Delta_j)}\right| \\
&= \frac{1}{\left(\frac{1}{4d^2+1}\right)^d} \cdot \frac{\left|4d^2 - \left(4d^2 + \sum_{j\in[d]}\frac{d^2}{j^2}\right)\right|}{\prod_{j\in I, j\neq\frac{1}{2}}(4d^2 - \frac{d^2}{j^2})} && \text{sustituyendo } \alpha, \Delta \\
&= \left(4d^2+1\right)^d \cdot \frac{\sum_{j\in[d]}\frac{d^2}{j^2}}{\prod_{j\in I, j\neq\frac{1}{2}}(4d^2 - \frac{d^2}{j^2})} && \text{simplificando}
\end{aligned} \tag{5.58}$$

$$
\begin{aligned}
&= \left(\frac{4d^2+1}{d^2}\right)^d \cdot \frac{\sum_{j\in[d]} \frac{d^2}{j^2}}{\prod_{j\in[d]}(4-\frac{1}{j^2})} && \text{agrupando exponenciales}\\
&\leq 4\left(1+\frac{1}{4d^2}\right)^d \cdot \frac{d^2\sum_{j\in[d]}\frac{1}{j^2}}{3^d} && \text{ya que } \frac{1}{j^2}\leq 1\\
&\leq 4\sqrt[4d]{e}\,\frac{d^2\frac{\pi^2}{6}}{3^d} && \text{por definición de } e \\
& && \text{y } \textstyle\sum_{j\in[d]}\frac{1}{j^2}\leq \sum_{j\in\mathbb{N}}\frac{1}{j^2}\leq \frac{\pi^2}{6}\\
&\leq \frac{2\pi^2}{3}d^2 && \text{ya que } 3^d > \sqrt[4d]{e}
\end{aligned}
\tag{5.59}
$$

Nos queda combinar las cotas obtenidas para los distintos rangos de k y acotar con esto la norma $\|c\|_1$.

3. Combinar casos de $k \in I$

Aplicando (5.45), (5.47), (5.58) concluimos que

$$\|c\|_1 = c_{\frac{1}{2}} + \sum_{j\in[d]} |c_j| \leq \frac{2\pi^2}{3}d^2 + \sum_{j\in[d]} 12e\frac{d^2}{j^2} \leq \frac{2\pi^2}{3}d^2 + 12e\frac{\pi^2}{6}d^2 = O(d^2), \tag{5.60}$$

lo que completa la demostración del lema para el caso homogéneo.

Faltaría por demostrar el caso no homogéneo. No incluiremos aquí su demostración detallada, pero sí daremos un esbozo de las ideas en las que se basa. En primer lugar, veamos que el desarrollo (5.38) dejaría de ser válido en esta situación, ya que los términos del sumatorio

$$\sum_{S\subset[n]} \hat{F}(S) \sum_{k\in[d+1]} c_k \prod_{k\in S}(\alpha_k y + \beta_k z)$$

pueden corresponderse con conjuntos $S \subset [n]$ tales que $|S| < d$, monomios que quedarían anulados si aplicáramos las condiciones descritas en la ecuación (5.37).

En la demostración descrita en [36, pp.12–16] se plantea la incorporación de $d+1$ condiciones adicionales (y, por lo tanto, $d+1$ coeficientes adicionales en α, β, c) para resolver este caso. La idea en la que se basan estas nuevas condiciones es que, para todo $d' < d$,

$$
\begin{aligned}
\sum_{k\in[d'+1]} c_j \alpha_k^{d'-t}\beta_k^t &= \sum_{k\in[d'+1]} c_j \alpha_k^{d'-t}\beta_k^t(\alpha_k+\beta_k) && \text{ya que } \alpha_k+\beta_k = |\alpha_k|+|\beta_k| = 1\\
&= \sum_{k\in[d+1]} c_j \alpha_k^{d'-t+1}\beta_k^t + \sum_{k\in[d+1]} c_j \alpha_k^{d'-t}\beta_k^{t+1}\\
&= \sum_{k\in[d+1]} c_j \alpha_k^{(d'+1)-t}\beta_k^t + \sum_{k\in[d+1]} c_j \alpha_k^{(d'+1)-(t+1)}\beta_k^{t+1}
\end{aligned}
\tag{5.61}
$$

Es decir, si denotamos

$$\Theta_{d',t} := \sum_{k\in[d'+1]} c_j \alpha_k^{d'-t}\beta_k^t,$$

la ecuación (5.61) equivaldría a

$$\Theta_{d',t} = \Theta_{d'+1,t} + \Theta_{d'+1,t+1} \tag{5.62}$$

$\Theta_{d',t}$		d'							
		1	2		...	$d-3$	$d-2$	$d-1$	d
	1	d	1			0	0	0	0
	2	0	$d-1$			0	0	0	0
				...	...				
t	...			...	...				
	$d-3$	0	0			3	1	0	0
	$d-2$	0	0			0	2	1	0
	$d-1$	0	0			0	0	1	1
	d	0	0			0	0	0	0

Teniendo en cuenta las restricciones que ya habíamos descrito en (5.37), la expresión de recurrencia de (5.62) genera una asignación de valores como los del diagrama anterior.

Con lo que, para cada $d' \in [d-1]$ y $t \in \{0, \cdots, d'\}$,

$$\sum_{k\in[d+1]} c_j \alpha_k^{d'-t} \beta_k^t = \begin{cases} d-d' & \text{si } t = d' \\ 1 & \text{si } t = d'-1 \\ 0 & \text{en caso contrario} \end{cases}$$

Las condiciones correspondientes al caso $t = d'$ son las "problemáticas" cuando aplicamos la solución descrita antes al caso no homogéneo. Veamos una idea para ver cómo anular estas casillas. Para cada $0 \le t \le d' \le d$,

$$\frac{1}{2}\sum_{k\in[d+1]} c_k \alpha_k^{d'-t}\beta_k^t - \frac{1}{2}\sum_{k\in[d+1]} c_k(-\alpha_k)^{d'-t}\beta_k^t = \frac{1-(-1)^{d'-t}}{2}\sum_{k\in[d+1]} c_k\alpha_k^{d'-t}\beta_k^t$$
$$= \begin{cases} 1 & \text{si } t = d'-1 \\ 0 & \text{en caso contrario} \end{cases}$$

Mediante esta idea incorporamos $m+1$ nuevas condiciones y coeficientes en α, β, c, ampliando la dimensión de los vectores al valor $m = 2(d+1)$ descrito en el enunciado. No detallamos estas ecuaciones por involucrar una complejidad que excede al espacio que pretendíamos dedicar a este resultado. □

El segundo lema relaciona el supremo de f con el de su versión desacoplada de un bloque $\check{f}$ de la siguiente forma: [36, p.5]:

Lema 5.2. *Sea $f : \{\pm 1\} \to \mathbb{R}$ de grado d, entonces $\|\check{f}\|_\infty \le O(d^2)\|f\|_\infty$*

Demostración. Sea $F : \mathbb{R}^n \to \mathbb{R}$, $F(x) = \sum_{S\subset[n]} \hat{F}(S)x^S$ la extensión multilineal de f, por el Lema 5.1 existen unos vectores $\alpha, \beta, c \in \mathbb{R}^m$, donde $m \le 2(d+1)$, tales que

$$\check{f}(y,z) = \sum_{k\in[m]} c_k F(\alpha_k y + \beta_k z), \qquad \text{para todo } y, z \in \{\pm 1\}^n,$$

y además $\|c\|_1 \le O(d^2), |\alpha_k| + |\beta_k| = 1$.

Teniendo esto en cuenta,

$$
\begin{aligned}
\|\check{f}\|_\infty &= \sup_{y,z\in\{\pm1\}^n} \left|\check{f}(y,z)\right| \\
&= \sup_{y,z\in\{\pm1\}^n} \left|\sum_{k\in[m]} c_k F(\alpha_k y+\beta_k z)\right| && \text{por el Lema 5.1} \\
&\le \sup_{y,z\in\{\pm1\}^n} \left\{\sum_{k\in[m]} |c_k|\left|F(\alpha_k y+\beta_k z)\right|\right\} \\
&\le \sup_{y,z\in\{\pm1\}^n} \left\{\sum_{k\in[m]} |c_k| \sup_{l\in[m]}\left|F(\alpha_l y+\beta_l z)\right|\right\} \qquad (5.63)\\
&= \sup_{\substack{y,z\in\{\pm1\}^n\\ l\in[m]}} \left\{\left|F(\alpha_l y+\beta_l z)\right| \sum_{k\in[m]} |c_k|\right\} \\
&= \|c\|_1 \sup_{\substack{y,z\in\{\pm1\}^n\\ l\in[m]}} \left|F(\alpha_l y+\beta_l z)\right| && \text{por definición de } \|c\|_1 \\
&= O(d^2) \sup_{\substack{y,z\in\{\pm1\}^n\\ l\in[m]}} \left|F(\alpha_l y+\beta_l z)\right| && \text{por (5.35) del Lema 5.1}
\end{aligned}
$$

Aquí tendremos en cuenta que para todo $y,z\in\{\pm1\}^n$, $j\in[n]$, $l\in[m]$ se cumple $|y_j|=|z_j|=1$, $|\alpha_l|+|\beta_l|=1$ (por (5.36) del Lema 5.1), y que por lo tanto todas las coordenadas del elemento $\alpha_l y+\beta_l z$ estarán contenidas en $[-1,1]$. Como además F es convexa (por ser afín respecto a cada una de sus variables),

$$
\sup_{\substack{y,z\in\{\pm1\}^n\\ l\in[m]}} \left|F(\alpha_l y+\beta_l z)\right| \le \sup_{x\in\{\pm1\}^n} |F(x)| = \sup_{x\in\{\pm1\}^n} |f(x)| = \|f\|_\infty \tag{5.64}
$$

Por lo tanto, a partir de (5.63) y (5.64),

$$\|\check{f}\|_\infty \le O(d^2)\|f\|_\infty$$

□

Procedemos ahora a demostrar el Teorema 5.4.

Demostración. Sea $f:\{\pm1\}\to\mathbb{R}$ una función expansión de Fourier tiene grado d, vamos a estudiar una cota sobre su varianza.

$$
\begin{aligned}
\mathrm{Var}[f] &\le \sum_{k\in[n]} \mathrm{Inf}_k[f] && \text{debido a (5.15)} \\
&\le \sqrt{\mathrm{maxInf}[f]} \sum_{k\in[n]} \sqrt{\mathrm{Inf}_k[f]} \\
&= \sqrt{\mathrm{maxInf}_k(f)} \sum_{k\in[n]} \sqrt{\sum_{S\ni k} \hat{f}(S)^2} && \text{aplicando (5.13)} \qquad (5.65)\\
&\le \sqrt{\mathrm{maxInf}[f]} \sum_{k\in[n]} E_{x\in\{\pm1\}^n}\left[\left(\sum_{S\ni k} \hat{f}(S)\, x^S\right)^2\right]^{\frac{1}{2}}
\end{aligned}
$$

(5.66)

$$
\begin{aligned}
\mathrm{Var}[f] &\le \sqrt{\mathrm{maxInf}[f]} \sum_{k\in[n]} E_{x\in\{\pm1\}^n}\left[\left(\sum_{S\ni k}\hat{f}(S)\,x^{S\setminus\{k\}}\right)^2\right]^{\frac{1}{2}} && x^{\{k\}}\in\{\pm1\} \text{ implica } \left(x^{\{k\}}\right)^2=1\\
&= \sqrt{\mathrm{maxInf}[f]} \sum_{k\in[n]} \left\|\left(\sum_{S\ni k}\hat{f}(S)\,x^{S\setminus\{k\}}\right)\right\|_2 && \text{por definición de norma en (5.3)}\\
&\le \sqrt{\mathrm{maxInf}[f]} \sum_{k\in[n]} e^{d-1}\left\|\left(\sum_{S\ni k}\hat{f}(S)\,x^{S\setminus\{k\}}\right)\right\|_1 && \text{aplicando (5.20) sobre polinomio de grado } d-1\\
&= e^{d-1}\sqrt{\mathrm{maxInf}[f]} \sum_{k\in[n]} E_{x\in\{\pm1\}^n}\left[\left|\sum_{S\ni k}\hat{f}(S)\,x^{S\setminus\{k\}}\right|\right] && \text{por (5.3)}\\
&= e^{d-1}\sqrt{\mathrm{maxInf}[f]}\, E_{x\in\{\pm1\}^n}\left[\sum_{k\in[n]}\left|\sum_{S\ni k}\hat{f}(S)\,x^{S\setminus\{k\}}\right|\right] && \text{por linealidad de la esperanza}\\
&\le e^{d-1}\sqrt{\mathrm{maxInf}[f]} \sup_{x\in\{\pm1\}^n}\sum_{k\in[n]}\left|\sum_{S\ni k}\hat{f}(S)\,x^{S\setminus\{k\}}\right|\\
&= e^{d-1}\sqrt{\mathrm{maxInf}[f]} \sup_{x\in\{\pm1\}^n}\sum_{k\in[n]}\sup_{y_k\in\{\pm1\}}\left(\sum_{S\ni k}\hat{f}(S)\,x^{S\setminus\{k\}}\right)y_k && \text{eligiendo } y_k \text{ según el signo dentro del valor absoluto}\\
&= e^{d-1}\sqrt{\mathrm{maxInf}[f]} \sup_{x,y\in\{\pm1\}^n}\sum_{k\in[n]}\sum_{S\ni k}\hat{f}(S)\,x^{S\setminus\{k\}}y^{\{k\}} && \text{tomando } y=(y_1,\cdots,y_n)\\
&\le e^{d-1}\sqrt{\mathrm{maxInf}[f]} \sup_{x,y\in\{\pm1\}^n}\left|\sum_{k\in[n]}\sum_{S\ni k}\hat{f}(S)\,x^{S\setminus\{k\}}y^{\{k\}}\right|\\
&= e^{d-1}\sqrt{\mathrm{maxInf}[f]} \sup_{y,z\in\{\pm1\}^n}\left|\sum_{k\in[n]}\sum_{S\ni k}\hat{f}(S)\,y^{S\setminus\{k\}}z^{\{k\}}\right| && \text{cambio de variables}\\
&= e^{d-1}\sqrt{\mathrm{maxInf}[f]} \sup_{y,z\in\{\pm1\}^n}\left|\sum_{S\subset[n]}\hat{f}(S)\sum_{k\in S}y^{S\setminus\{k\}}z^{\{k\}}\right| && \text{reordenando términos}\\
&= e^{d-1}\sqrt{\mathrm{maxInf}[f]} \sup_{y,z\in\{\pm1\}^n}\left|\check{f}(y,z)\right| && \text{por (5.5) en Definición 5.2}\\
&= e^{d-1}\sqrt{\mathrm{maxInf}[f]}\ \|\check{f}\|_\infty\\
&\le e^{d-1}\sqrt{\mathrm{maxInf}[f]}\, O(d^2)\,\|f\|_\infty && \text{aplicando el Lema 5.2}
\end{aligned}
$$

Y, sea $i\in[n]$ el índice tal que $\mathrm{Inf}_i[f]=\mathrm{maxInf}[f]$, concluimos de (5.66) que

$$\mathrm{Var}[f]\le O(d^2)\,e^{d-1}\sqrt{\mathrm{Inf}_i[f]}$$

Por lo que:

(5.67)
$$\mathrm{Inf}_i[f]\ge\left(\frac{\mathrm{Var}[f]}{O(d^2)\,e^{d-1}}\right)^2$$

□

5.4. Conjetura válida para polinómios simétricos

En 2012 Arturs Backurs, a resultas de una serie de conversaciones con Scott Aaronson, publicó en la página web de Aaronson [9] algunos resultados relativos a la conjetura. Uno de ellos describe que la Conjetura AA es válida para polinomios multilineales simétricos (polinomios invariantes bajo permutación de sus variables), e incluye un esbozo de la demostración. Posteriormente, en 2019, Ivanishvili publicó una versión más elaborada de esta demostración en su blog Zeros and Ones [24].

En esta sección describiremos este resultado e incluiremos una demostración basada en las ideas de Ivanishvili. Empezaremos con la definición de función simétrica.

Definición 5.3. [34, p.45] Una función $f : \{\pm 1\}^n \to \mathbb{R}$ es **simétrica** si, para todo elemento $x \in \{\pm 1\}^n$ y permutación $\pi \in S_n$,

$$f(x^\pi) = f(x), \tag{5.68}$$

donde x^π es el vector resultante de permutar las coordenadas de x según π. Es decir, $x^\pi := (x_1, x_2, \cdots, x_n)^\pi = (x_{\pi(1)}, x_{\pi(2)}, \cdots, x_{\pi(n)})$.

Una característica importante de la expansión de Fourier de funciones booleanas simétricas es que los coeficientes de Fourier correspondientes a monomios del mismo grado son todos iguales. Es decir, para dos conjuntos $S, S' \subset [n]$ tales que $|S| = |S'|$, necesariamente $\hat{f}(S) = \hat{f}(S')$. De no ser así, una permutación de variables podría alterar la evaluación de la función, con lo que ésta dejaría de ser simétrica. Este hecho será clave en los razonamientos que detallaremos más adelante.

Antes de abordar el resultado principal de la sección, introduciremos el siguiente lema:

Lema 5.3. *Todo polinomio multilineal real simétrico $f : \{\pm 1\}^n \to \mathbb{R}$ de grado d se puede expresar en función de un polinomio real $p : \mathbb{R} \to \mathbb{R}$, también de grado d, de la siguiente forma:*

$$f(x_1, \ldots, x_n) = p(x_1 + \ldots + x_n), \qquad \text{donde } (x_1, \ldots, x_n) \in \{\pm 1\}^n. \tag{5.69}$$

Demostración. Demostraremos este resultado por inducción sobre el grado del polinomio.

Para polinomios de grado 0 el resultado es inmediato, ya que la función es una constante.

Sea $k \in \mathbb{N}$, $k \leq n$, supongamos ahora que el enunciado es válido para cualquier polinomio multilineal simétrico de grado $k-1$, y estudiemos el caso en que f tiene grado k. Tal como hemos indicado antes, al ser f simétrico, todos los monomios de grado k tendrán el mismo coeficiente. Denotemos $C := \hat{f}(S)$ a este coeficiente, para todo $S \subset [n]$ tal que $|S| = k$. A partir de la expansión

de Fourier de f obtenemos que:

$$
\begin{aligned}
f(x_1,\cdots,x_n) &= \sum_{S\subset[n]} \hat{f}(S)x^S \\
&= \sum_{\substack{S\subset[n]\\|S|=k}} \hat{f}(S)x^S + \sum_{\substack{S\subset[n]\\|S|<k}} \hat{f}(S)x^S \\
&= \sum_{\substack{S\subset[n]\\|S|=k}} Cx^S + \sum_{\substack{S\subset[n]\\|S|<k}} \hat{f}(S)x^S && \text{por ser los coeficientes iguales} \\
&= C \sum_{1\le i_1<\cdots<i_k\le n} \prod_{j\in[k]} x_{i_j} + \sum_{\substack{S\subset[n]\\|S|<k}} \hat{f}(S)x^S
\end{aligned} \tag{5.70}
$$

Estudiemos ahora el desarrollo del polinomio multilineal simétrico $(x_1+\cdots+x_n)^k$. Observamos que el grado máximo de este polinomio se corresponderá con monomios de k variables, y que además la expansión de esta potencia generará $k!$ términos de cada uno de estos monomios (uno para cada posible ordenación de sus variables). Así pues,

$$
(x_1+\ldots+x_n)^k = q(x_1,\ldots,x_n) + k! \sum_{1\le i_1<\ldots<i_k\le n} \prod_{j\in[k]} x_{i_j}, \tag{5.71}
$$

donde $q:\{\pm1\}^n\to\mathbb{R}$ es un polinomio multilineal de grado máximo $k-1$. Además, q es simétrico, por serlo también el resto de componentes de la ecuación (5.71).

Teniendo esto en cuenta, seguimos con el desarrollo de (5.70):

$$
\begin{aligned}
f(x_1,\cdots,x_n) &= C \sum_{1\le i_1<\cdots<i_k\le n} \prod_{j\in[k]} x_{i_j} + \sum_{\substack{S\subset[n]\\|S|<k}} \hat{f}(S)x^S \\
&= \frac{C}{k!}(x_1+\cdots+x_n)^k - \frac{C}{k!}\, q(x_1,\cdots,x_n) + \sum_{\substack{S\subset[n]\\|S|<k}} \hat{f}(S)x^S && \text{por (5.71)}
\end{aligned} \tag{5.72}
$$

El término $-\frac{C}{k!}\ q(x_1,\ldots,x_n)+\sum_{S\subset[n],|S|<k}\hat{f}(S)x^S$ es un polinomio multilineal simétrico de grado máximo $k-1$, así que por hipótesis de inducción se puede descomponer como un polinomio (de grado inferior a k) sobre $x_1+\ldots+x_n$. Y el término $\frac{C}{k!}(x_1+\ldots+x_n)^k$ es un monomio de grado k, también sobre $x_1+\ldots+x_n$. Por lo tanto, queda verificado (5.69) y el grado del polinomio descrito en la ecuación (5.72) es de grado k. □

A continuación enunciamos el teorema principal y detallamos su demostración.

Teorema 5.5. *Sea $f:\{\pm1\}^n\to[0,1]$ un polinomio multilineal real simétrico de grado d, existe $i\in[n]$ tal que*

$$
\mathrm{Inf}_i[f] \ge \frac{1}{48}\left(\frac{\mathrm{Var}[f]}{d}\right)^4 \tag{5.73}
$$

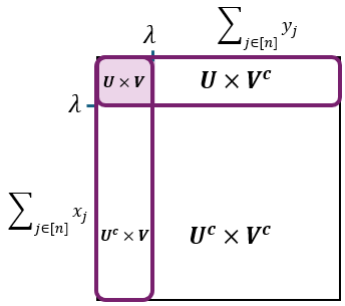
Demostración. Empezaremos estudiando una cota para la varianza:

(5.74)
$$\begin{aligned}\mathrm{Var}[f] &= \frac{1}{2}E_{x,y\in\{\pm1\}^n}\left[(f(x)-f(y))^2\right] && \text{por (5.12)}\\ &= \frac{1}{2}E_{x,y\in\{\pm1\}^n}\left[(p(x_1+\cdots+x_n)-p(y_1+\cdots+y_n))^2\right] && \text{aplicando (5.69)}\end{aligned}$$

Para un parámetro $\lambda > 0$ que fijaremos más adelante, definimos los conjuntos:

(5.75)
$$\begin{aligned}U &= \{(x_1,\cdots,x_n)\in\{\pm1\}^n : |x_1+\cdots+x_n| < \lambda\}\\ V &= \{(y_1,\cdots,y_n)\in\{\pm1\}^n : |y_1+\cdots+y_n| < \lambda\}\end{aligned}$$

En el siguiente diagrama representamos su producto cartesiano (y el de sus complementos) en $\{\pm1\}^n \times \{\pm1\}^n$:



La estrategia que seguiremos para acotar la última expresión de (5.74) consistirá en estudiar la esperanza sólo sobre los puntos $(x,y)\in U\times V$. Es decir, los puntos tales que

(5.76)
$$|x_1+\cdots+x_n| < \lambda \quad \text{y} \quad |y_1+\cdots+y_n| < \lambda$$

El objetivo es que esta esperanza esté suficientemente acotada, y que al mismo tiempo la probabilidad de que $x,y\in\{\pm1\}^n$ no esté en este supuesto sea también reducida.

Comprobemos que esto es así para el siguiente valor de λ:

(5.77)
$$\lambda = \sqrt{\frac{6n}{\mathrm{Var}[f]}}$$

Denotamos $1_U : \{\pm1\}^n \to \{0,1\}$ la función tal que $1_U(x) = 1$ cuando $x\in U$ y $1_U(x) = 0$ en caso contrario. De forma análoga, definimos las funciones 1_V, 1_{U^c}, y 1_{V^c} para los conjuntos V, $U^c := \{\pm1\}^n \setminus U$ y $V^c := \{\pm1\}^n \setminus V$ respectivamente.

Seguimos a partir de la ecuación (5.74):

(5.78)
$$\begin{aligned}\mathrm{Var}[f] &= E_{x,y\in\{\pm1\}^n}\left[(p(x_1+\cdots+x_n)-p(y_1+\cdots+y_n))^2\right]\\ &= E_{x,y\in\{\pm1\}^n}\left[(p(x_1+\cdots+x_n)-p(y_1+\cdots+y_n))^2\,(1_U(x)1_V(y)+1_{U^c}(x)1_V(y)+1_U(x)1_{V^c}(y)+1_{U^c}(x)1_{V^c}(y))\right]\\ &= E_{x,y\in\{\pm1\}^n}\left[(p(x_1+\cdots+x_n)-p(y_1+\cdots+y_n))^2\,(1_{U^c}(x)1_V(y)+1_U(x)1_{V^c}(y)+1_{U^c}(x)1_{V^c}(y))\right]\\ &\quad + E_{x,y\in\{\pm1\}^n}\left[(p(x_1+\cdots+x_n)-p(y_1+\cdots+y_n))^2\,1_U(x)1_V(y)\right]\end{aligned}$$

Acotamos el primer sumando de la ecuación anterior teniendo en cuenta que

(5.79)
$$(p(x_1+\cdots+x_n)-p(y_1+\cdots+y_n))^2 \le 1, \qquad \text{para todo } x:=(x_1,\cdots,x_n), y:=(y_1,\cdots,y_n)\in\{\pm1\}^n$$

ya que $p(x_1+\cdots+x_n)=f(x)\in[0,1]$ para todo $x:=(x_1,\cdots,x_n)\in\{\pm1\}^n$, así que la diferencia entre dos evaluaciones de p está contenida en $[-1,1]$ y su cuadrado en $[0,1]$:

Además,

$$P((x,y)\in U^c\times V) \;\le\; P((x,y)\in U^c\times\{\pm1\}^n) \;\le\; P(|x_1+\cdots+x_n|>\lambda) \qquad \text{por (5.75)} \tag{5.80}$$

y análogamente

$$\begin{aligned} P((x,y)\in U\times V^c) &\;\le\; P(|x_1+\cdots+x_n|>\lambda)\\ P((x,y)\in U^c\times V^c) &\;\le\; P(|x_1+\cdots+x_n|>\lambda) \end{aligned} \tag{5.81}$$

Por lo tanto:

(5.82)
$$\begin{aligned}
&E_{x,y\in\{\pm1\}^n}\left[(p(x_1+\cdots+x_n)-p(y_1+\cdots+y_n))^2\,(1_{U^c}(x)1_V(y)+1_U(x)1_{V^c}(y)+1_{U^c}(x)1_{V^c}(y))\right]\\
&\quad\le E_{x,y\in\{\pm1\}^n}[1_{U^c}(x)1_V(y)+1_U(x)1_{V^c}(y)+1_{U^c}(x)1_{V^c}(y)] && \text{por (5.79)}\\
&\quad= E_{x,y\in\{\pm1\}^n}[1_{U^c}(x)1_V(y)]+E_{x,y\in\{\pm1\}^n}[1_U(x)1_{V^c}(y)]+E_{x,y\in\{\pm1\}^n}[1_{U^c}(x)1_{V^c}(y)]\\
&\quad\le P(|x_1+\cdots+x_n|\ge\lambda))+P(|x_1+\cdots+x_n|\ge\lambda))+P(|x_1+\cdots+x_n|\ge\lambda)) && \text{(5.80),(5.81)}\\
&\quad\le 3\;P(|x_1+\cdots+x_n|\ge\lambda)\\
&\quad= 3\;P(|x_1+\cdots+x_n|^2\ge\lambda^2)\\
&\quad\le 3\;\frac{E[|x_1+\cdots+x_n|^2]}{\lambda^2} && \text{por (4.14)}\\
&\quad= 3\;E[|x_1+\cdots+x_n|^2]\;\frac{1}{\left(\sqrt{\frac{6n}{\mathrm{Var[f]}}}\right)^2} && \text{por (5.77)}\\
&\quad= 3n\frac{\mathrm{Var[f]}}{6n} && (*)\\
&\quad= \frac{1}{2}\,\mathrm{Var}[f]
\end{aligned}$$

(*) La penúltima igualdad es debida a que la expansión de $|x_1+\cdots+x_n|^2$ genera n términos cuadrados $\{x_j^2\}_{j\in[n]}$, cada uno de los cuales toma valor 1, y una colección de términos $\{2x_ix_j\}_{i\neq j\in[n]}$ cuya suma esperada se anula, con lo que la esperanza de la expresión es n.

Para acotar el segundo sumando de la ecuación (5.78) nos basaremos en la desigualdad de los hermanos Markov [4], que relaciona el máximo valor de la k-ésima derivada de un polinomio $q:\mathbb{R}\to\mathbb{R}$ de grado d en un intervalo $[a,b]$ con el máximo del polinomio en este mismo intervalo. Para el caso $k=1$, su expresión es la siguiente [30]:

$$\max_{x\in[a,b]}\left|q'(x)\right| \le \frac{d^2}{\frac{b-a}{2}}\max_{x\in[a,b]}|q(x)| \tag{5.83}$$

[4] No confundir con la desigualdad de Markov descrita en (4.14), a la que también referimos en esta demostración.

Además, al ser p continua y derivable en $[-1,1]$, por el teorema del valor medio existe $\chi \in [-1,1]$ tal que

$$p(x_1+\cdots+x_n)-p(y_1+\cdots+y_n) = p'(\chi)((x_1+\cdots+x_n)-(y_1+\cdots+y_n)) \tag{5.84}$$

Con esta información acotamos el segundo sumando de (5.78):

(5.85)

$$\begin{aligned}
&E_{x,y\in\{\pm1\}^n}\left[(p(x_1+\cdots+x_n)-p(y_1+\cdots+y_n))^2\ 1_U(x)1_V(y)\right] \\
&= E_{x,y\in\{\pm1\}^n}\left[\left(p'(\chi)((x_1+\cdots+x_n)-(y_1+\cdots+y_n))\right)^2 1_U(x)1_V(y)\right] && \text{aplicando (5.84)} \\
&\le E_{x,y\in\{\pm1\}^n}\left[\left(\frac{d^2}{\frac{n-(-n)}{2}}((x_1+\cdots+x_n)-(y_1+\cdots+y_n))\right)^2 1_U(x)1_V(y)\right] && \text{por (5.83) y } \|p\|_\infty \le 1 \\
&= \frac{d^4}{n^2}E_{x,y\in\{\pm1\}^n}\left[((x_1+\cdots+x_n)-(y_1+\cdots+y_n))^2\ 1_U(x)1_V(y)\right] \\
&\le \frac{d^4}{n^2}E_{x,y\in\{\pm1\}^n}\left[(2\lambda)^2\right] && \text{por (5.75), ya que } x\in U, y\in V \\
&= 4\frac{d^4}{n^2}\lambda^2 \\
&= 4\frac{d^4}{n^2}\frac{6n}{\mathrm{Var}[f]} && \text{sustituyendo } \lambda \\
&= \frac{24d^4}{n\mathrm{Var}[f]}
\end{aligned}$$

Combinando (5.78), (5.82) y (5.85) obtenemos

$$\mathrm{Var}[f] \le \frac{1}{2}\mathrm{Var}[f] + \frac{24d^4}{n\mathrm{Var}[f]} \quad \Rightarrow \quad (\mathrm{Var}[f])^2 \le \frac{48d^4}{n}$$

Por lo tanto,

$$(\mathrm{Var}[f])^2\,\frac{n}{d^4} \le 48 \tag{5.86}$$

Finalmente, al ser f simétrico, todas sus variables tienen la misma influencia, así que para todo $i \in [n]$:

$$\begin{aligned}
\mathrm{Inf}_i[f] &= \frac{\mathrm{Inf}[f]}{n} \\
&\ge \frac{\mathrm{Var}[f]}{n} && \text{por (5.15)} \\
&= \frac{1}{(\mathrm{Var}[f])^2\,\frac{n}{d^4}}\frac{(\mathrm{Var}[f])^3}{d^4} && \text{reorganizando términos} \\
&\ge \frac{(\mathrm{Var}[f])^3}{48d^4} && \text{sustituyendo (5.86)} \\
&\ge \frac{(\mathrm{Var}[f])^4}{48d^4} && \text{por ser } \mathrm{Var}[f] \le 1
\end{aligned}$$

Por lo tanto,

$$\mathrm{Inf}_i[f] \geq \frac{1}{48}\left(\frac{\mathrm{Var}[f]}{d}\right)^4 \tag{5.87}$$

□

5.5. Funciones desacopladas de un bloque

En 2015 O'Donnell y Zhao expusieron en [36] la equivalencia de estudiar la Conjetura AA en el caso general a hacerlo sobre funciones desacopladas de un bloque (ver Definición 5.2).

Tal como introducíamos en la Sección 5.1.2 , el desacoplamiento de una función consiste en modificarla de forma que se introduce cierta independencia entre sus variables de entrada. Esta independencia puede favorecer el estudio de características sobre estas funciones y, en particular, la obtención de cotas sobre ellas. El resultado de O'Donnell y Zhao que estudiamos en esta sección sienta las bases para que eventuales futuros resultados sobre la versión desacoplada pudieran servir para demostrar, total o parcialmente, la versión general de la Conjetura AA.

Veamos el teorema y su demostración:

Teorema 5.6. *[36, p.6] La Conjetura AA es cierta si y sólo si lo es para funciones desacopladas de un bloque.*

Demostración. La condición necesaria es evidente. Veamos la condición suficiente.

Sea $f : \{\pm 1\}^n \to [0,1]$ un polinomio multilineal de grado d, y $\check{f}$ su versión desacoplada de un bloque. Por el Lema 5.2,

$$\|\check{f}\|_\infty \leq O(d^2)\|f\|_\infty \leq O(d^2).$$

Por lo tanto, existe una constante $C > 0$ tal que

$$\|\check{f}\|_\infty \leq Cd^2 \tag{5.88}$$

Si definimos la siguiente función

$$g : \{\pm 1\}^{2n} \to [0,1], \qquad g(x) = \frac{\check{f}(x)}{Cd^2} \text{ para todo } x \in \{\pm 1\}^{2n},$$

podemos aplicar la Conjetura AA sobre ella, ya que g es una función desacoplada de un bloque (por corresponderse con $\check{f}$ multiplicado por una constante), del mismo grado d que f, y además $\|g\|_\infty = \|\check{f}\|_\infty/(Cd^2) \leq (Cd^2)/(Cd^2) = 1$ (por lo tanto, $g\left(\{\pm 1\}^{2n}\right) \subset [0,1]$).

En consecuencia, existe un índice $k^* \in [2n]$ tal que

$$\mathrm{Inf}_{k^*}[g] \geq w\left(\frac{\mathrm{Var}[g]}{d}\right), \tag{5.89}$$

donde $w : \mathbb{R} \to \mathbb{R}$ es el polinomio descrito en la conjetura AA de la forma $w(x) = Kx^a$.

Veamos cómo aplicar una desigualdad análoga sobre $\check{f}$:

(5.90)
$$\begin{aligned}
\mathrm{Inf}_{k^*}[\check{f}] &= \mathrm{Inf}_{k^*}[Cd^2 g] \\
&= C^2 d^4 \,\mathrm{Inf}_{k^*}[g] && \text{por (5.16)} \\
&\geq C^2 d^4 K\left(\frac{\mathrm{Var}[g]}{d}\right)^a && \text{por (5.89)} \\
&= C^2 d^4 C^{-2a} d^{-4a} K\left(\frac{C^2 d^4 \mathrm{Var}[g]}{d}\right)^a \\
&= C^{2(1-a)} d^{4(1-a)} K\left(\frac{\mathrm{Var}[Cd^2 g]}{d}\right)^a && \text{por (5.17)} \\
&= C^{2(1-a)} K \frac{1}{d^{4(a-1)}}\left(\frac{\mathrm{Var}[\check{f}]}{d}\right)^a && \text{por definición de } g \\
&\geq C^{2(1-a)} K\left(\frac{\mathrm{Var}[\check{f}]}{d}\right)^{4(a-1)}\left(\frac{\mathrm{Var}[\check{f}]}{d}\right)^a && \text{por ser } \mathrm{Var}[\check{f}] \leq 1 \\
&= C^{2(1-a)} K\left(\frac{\mathrm{Var}[\check{f}]}{d}\right)^{5a-4}
\end{aligned}$$

Denotamos el índice $k' \in [n]$ tal que $k' = k^*$ si $k^* \leq n$, o $k' = k^* - n$ en caso contrario. Recordemos la definición de $\check{f}$ descrita en (5.5):

$$\check{f} : \left(\{\pm 1\}^n\right)^2 \to \mathbb{R}, \qquad \check{f}(y,z) = \sum_{S \subset [n]} \hat{f}(S) \sum_{k \in S} z^{S \setminus \{k\}} y^{\{k\}} \qquad \text{para } y, z \in \{\pm 1\}^n$$

Observamos de la ecuación anterior que para cada coeficiente $\hat{f}(S)$ del espectro de Fourier de f se generan $|S|$ coeficientes en $\check{f}$. Por lo tanto, aplicando (5.13) obtendremos $\mathrm{Inf}_{k^*}[\check{f}] = \sum_{S \ni k'} |S| \hat{f}(S)^2$, a partir de lo cual podemos acotar superiormente la influencia de k^* en $\check{f}$:

$$\mathrm{Inf}_{k^*}[\check{f}] = \sum_{S \ni k'} |S| \hat{f}(S)^2 \leq \sum_{S \ni k'} d \hat{f}(S)^2 = d \,\mathrm{Inf}_{k'}[f] \tag{5.91}$$

De forma análoga, acotamos inferiormente la varianza de $\check{f}$:

$$\mathrm{Var}[\check{f}] = \sum_{S \neq \emptyset} |S| \hat{f}(S)^2 \geq \sum_{S \neq \emptyset} \hat{f}(S)^2 = \mathrm{Var}[f] \tag{5.92}$$

De los resultados previos concluimos que

(5.93)
$$\begin{aligned}
\mathrm{Inf}_{k'}[f] &\geq \frac{\mathrm{Inf}_{k^*}[\check{f}]}{d} && \text{por (5.91)} \\
&\geq C^{2(1-a)} K\left(\frac{\mathrm{Var}[\check{f}]}{d}\right)^{5a-4} \frac{1}{d} && \text{por (5.90)} \\
&\geq C^{2(1-a)} K\left(\frac{\mathrm{Var}[\check{f}]}{d}\right)^{5a-3} && \text{por ser } \mathrm{Var}[\check{f}] \leq 1 \\
&\geq C^{2(1-a)} K\left(\frac{\mathrm{Var}[f]}{d}\right)^{5a-3} && \text{aplicando (5.92)}
\end{aligned}$$

Por lo tanto, existen constantes universales $K' = C^{2(1-a)}K$ y $a' = 5a - 3$ tales que

$$\mathrm{Inf}_{k'}[f] \geq K' \left(\frac{\mathrm{Var}[f]}{d} \right)^{a'}, \tag{5.94}$$

así que la Conjetura AA aplica también sobre f. □

5.6. Cronología de resultados

En esta sección vamos a revisar los diversos resultados que se han obtenido en relación con las Conjeturas AA y Cuántica. El objetivo es obtener una visión general de las líneas de trabajo que hay abiertas y de su cronología, por lo que evitaremos entrar en demostraciones o en una presentación demasiado detallada de los teoremas ni de los conceptos asociados (aunque sí incorporaremos referencias a todos ellos).

En ocasiones, los autores han dado indicaciones de la forma en la que consideran que su resultado podría contribuir a una demostración global de la Conjetura AA o Cuántica, y posibles lineas de investigación. En los casos que sea así, lo señalaremos también en el párrafo correspondiente a ese resultado.

Para evitar que esta cronología sea parcial, incorporaremos también los resultados de las secciones anteriores, aunque aquí sólo los mencionaremos, sin repetir lo descrito antes.

(2005) Conjetura AA válida para funciones booleanas

Tal como describimos en la Sección 5.2, en 2005, tres años antes de enunciarse la Conjetura AA, O'Donnell, Saks, Schramm y Servedio demostraron en [35] un teorema del que se deriva la validez de la Conjetura AA para funciones con rango booleano.

(2006) Conjetura AA válida para la versión exponencial

También previamente al enunciado de la Conjetura AA, Dinur y Friedgut demostraron en [18] un teorema que se corresponde con una versión más débil de ésta, en la que se sustituye la referencia al grado del polinomio por el exponencial de este grado. En 2019 Defant, Mastylo y Pérez publicaron una demostración alternativa de este teorema [14]. Hemos revisado este resultado en la Sección 5.3.

(2008) Conjetura AA

Aaronson y Ambainis enunciaron la Conjetura AA en [4], y demostraron que de ésta se derivaría la Conjetura Cuántica. Según indicaron, la inspiración para sugerir esta conjetura proviene de los trabajos de Dinur y Friedgut, y de O'Donnell et al. En el Capítulo 4 de este trabajo se puede consultar más detalle de los resultados descritos en ese artículo.

(2012) Conjetura AA válida para polinomios simétricos

En 2012, Arturs Backurs publicó una demostración de que la Conjetura AA es válida para polinomios multilineales simétricos (polinomios invariantes bajo permutación de sus variables). En 2019, Ivanishvili publicó otra versión más detallada en su blog Zeros and Ones. En la Sección 5.4 describimos con detalle este resultado e incluimos una adaptación de las demostraciones anteriores.

Ivanishvili incluye una observación interesante en su blog [24]: afirma que la clase de funciones simétricas en el cubo discreto es de tal importancia que la validez de la conjetura en éstas sugiere que es muy plausible que también sea cierta en el caso general. No da más detalle de esta intuición.

(2012) Conjetura AA válida para formas multilineales tales que todos los coeficientes tienen la misma magnitud

Sean $k < n \in \mathbb{N}$, denotamos que un polinomio homogeneo $f : (\{\pm 1\}^n)^k \to \mathbb{R}$ es una *forma multilineal* si su representación en forma $f(y_1, \cdots, y_k)$ para $y_j \in \{\pm 1\}^n$, $j \in [k]$, es lineal en cada input [31, p.11]. Por ejemplo, si $y' \in \{\pm 1\}^n$, entonces $f(y_1 + y', y_2, ..., y_k) = f(y_1, y_2, ..., y_k) + f(y', y_2, ..., y_k)$, y lo mismo sucede para el resto de coordenadas.

En 2012, Montanaro demostró en [31, pp.12–14] que la Conjetura AA es válida para toda forma multilineal cuyos coeficientes tienen todos la misma magnitud.

(2018) Conjetura AA válida si lo es para funciones desacopladas de un bloque

En el artículo de O'Donnell y Zhao [36] se demuestra la equivalencia de que la Conjetura AA sea válida en el caso de funciones desacopladas de un bloque a que lo sea en el caso general. En la Sección 5.5 se detalla y demuestra este resultado.

(2019) Demostración desestimada

En noviembre de 2019, Keller y Klein publicaron en [25] una demostración del caso general de la Conjetura AA. Poco después, Ivanishvili identificó un error en uno de los lemas de la demostración que invalidaba el argumento, y al cabo de un mes fue despublicada por sus autores.

(2022) Los polinomios de aceptación de algoritmos cuánticos son polinomios de Fourier completamente acotados

A partir del trabajo de Arunachalam, Briët y Palazuelos [8], en 2022 Escudero obtuvo diversos resultados interesantes en [19].

Sea $f : \{\pm 1\}^n \to \mathbb{R}$ un polinomio multilineal de grado máximo d, Escudero define una norma que denomina d-norma de Fourier completamente acotada, y que denota $\|f\|_{fcb,d}$ [19, Def 2.2, p.6]. A partir de ésta obtiene dos resultados:

(1) El polinomio que evalúa la probabilidad de aceptación de un algoritmo cuántico de d-queries cumple que $\|f\|_{fcb,2d} \leq 1$ (ver [19, Teorema 1.4, pp.9–11]).

(2) Se demuestra la Conjetura AA en el caso particular de polinomios $f : \{\pm 1\}^n \to \mathbb{R}$ homogéneos de grado d y $\|f\|_{fcb,d} \leq 1$ (ver [19, Teorema 1.6, pp.15–16]).

Recopilando los resultados anteriores, observamos que el único obstáculo para completar la demostración de la Conjetura Cuántica es el requisito de homogeneidad en el polinomio del Teorema 1.6 de [19]. Escudero conjetura en [19, Conjetura 1.5, p.4] que este requisito de homogeneidad es prescindible, y propone algunas ideas de cómo podría abordarse su demostración. El problema sigue abierto, pero en caso de confirmarse la validez de esta otra conjetura quedaría demostrada la Conjetura Cuántica y las consecuencias en la necesidad de inputs altamente estructurados en los problemas en los que se alcanza la supremacía cuántica.

(2023) ε-sensitividad de bloque y ε-complejidad de certificado

En 2023, Lovett y Zhang publicaron un artículo [29] en el que introducen una nueva conjetura a partir de la cual podría deducirse la Conjetura AA. Veamos primero algunos conceptos en los que se basa: sea $f : \{\pm 1\}^n \to [0,1]$ un polinomio multilineal de grado d, $x \in \{\pm 1\}^n$ y $\varepsilon > 0$,

la *ε-sensitividad de bloque* en x ($BS_\varepsilon(f,x)$) es el número máximo $k \in \mathbb{N}$ de bloques de índices disjuntos $B_1, \cdots, B_k \subset [n]$ tales que, para todo $j \in [k]$, $|f(x) - f(x^{(B_j)}| \geq \varepsilon$, donde $x^{(B_j)}$ denota el resultado de invertir en x las coordenadas correspondientes a los índices de B_j,

la *ε-sensitividad de bloque fraccional* ($FBS_\varepsilon(f,x)$) se corresponde con el valor máximo de $BS_\varepsilon(f,x)$ sobre todas las posibles relajaciones sobre $\{\pm 1\}^n$,

la *ε-complejidad de certificado* ($CS_\varepsilon(f,x)$) representa el mínimo número de índices contenidos en los bloques B_j descritos antes, y

la *ε-complejidad de certificado fraccional* ($FCS_\varepsilon(f,x)$) denota el mínimo $k \in \mathbb{N}$ tal que, para alguna distribución de probabilidad π sobre $[n]$, la probabilidad de que cualquier índice $j \in N$ pertenezca a cualquiera de los bloques $\{B_i \subset [n]\}_{i \in \mathbb{N}}$ descritos en el concepto de $BS_\varepsilon(f,x)$ sea, por lo menos, $P(j \in B_i) \geq 1/k$.

A partir de estos conceptos demuestran los siguientes resultados:

(1) $FCS_\varepsilon(f,x) \leq \text{poly}(d, 1/\varepsilon, \log n)$
(es decir, todos los inputs tienen una complejidad de certificado fraccional pequeña).

(2) Si $\text{Var}[f] = O(\varepsilon)$, en al menos una ε-fracción de los inputs $x \in \{\pm 1\}^n$ hay un bloque $B_i \subset [n]$ de tamaño $|B_i| \leq r$ tal que $|f(x) - f(x^{(B_i)})| \geq \varepsilon$, donde $r = \text{poly}(FCS_\varepsilon(f,x), \log(1/\varepsilon))$
(es decir, la mayoría de los inputs tienen una sensibilidad de bloque reducida).

Finalmente, conjeturan que la desigualdad de Talagrand (ver [13]) puede extenderse de la norma L_2 (para la que está definida) a la norma L_∞, y demuestran que esta conjetura, combinada con los resultados previos, permite deducir la Conjetura AA.

En vista de este resultado, los autores sugieren estudiar esta nueva conjetura sobre la desigualdad de Talagrand como vía para demostrar la Conjetura AA.

(2024) Conjetura AA válida para restricciones aleatorias

Veamos primero un concepto de restricción de función más general que el descrito en (4.11). Sea $f : \{\pm 1\} \to \mathbb{R}$, $S \subset [n], y \subset \{\pm 1\}^{[n]\setminus S}$ [5] , una *restricción* $\rho = (S, y)$ sobre f es una función $f_\rho : \{\pm 1\}^S \to \mathbb{R}$ tal que, para todo $z \in \{\pm 1\}^S$, si $x \in \{\pm 1\}^n$ toma valor $x_i = z_i$ en índices $i \in S$, $x_i = y_i$ en índices $i \in [n] \setminus S$, entonces $f_\rho(z) = f(x)$.

Una *restricción aleatoria con probabilidad de supervivencia* $p \in (0,1]$ es una muestra $\rho = (S, y)$ donde y toma valor con distribución uniforme en $\{\pm 1\}^{[n]\setminus S}$, y cada índice $i \in [n]$ está incluido en S con probabilidad p.

En 2024, pocas semanas antes de iniciarse este TFM, Bhattacharya publicó en [11] un resultado en el que demuestra que

$$P\left(\text{maxInf}[f_\rho] \geq \text{poly}\left(\frac{\text{Var[f]}}{d^{C_2}}\right)\right) \geq \frac{\text{Var}[f]\ \log(d)}{50 C_1 d},$$

donde $C_1, C_2 > 0$ son constantes universales y ρ denota una restricción aleatoria con probabilidad de supervivencia $\log(d)/C$.

Combinando esta ecuación con los resultados obtenidos por Lovett y Zhang en [27], concluye que la Conjetura AA es válida para restricciones aleatorias en una fracción considerable de los datos, suponiendo que su varianza no es demasiado baja.

Al final de su artículo, Bhattacharya sugiere una posible vía de demostración del caso general de la Conjetura AA para cualquier aplicación $f : \{\pm 1\}^n \to [0,1]$ de grado d. La idea consistiría en identificar una función $g : \{\pm 1\}^m \to \{\pm 1\}^n$ de grado más reducido que f, e "insesgada" de tal forma que la composición $f \circ g : \{\pm 1\}^m \to [0,1]$ siguiera siendo de bajo grado, preservara la varianza de f, y al mismo tiempo mantuviera cierta correlación entre sus coordenadas de forma que el hecho de que una amplia fracción de las restricciones aleatorias sobre $f \circ g$ tengan una variable con alta influencia, permita identificar también una variable de alta influencia en f.

(2025) Conjetura simplificada sobre pseudoaleatoriedad cuántica

Decimos que una variable aleatoria $X \in \{0,1\}^N$ es δ-densa si, para todo subconjunto de coordenadas $S \subseteq [N]$, su distribución marginal restringida a las coordenadas en S tiene una entropía mínima mayor o igual a $(1-\delta)|S|$.

En [28], Liu, Mutreja y Yuen exponen una conjetura que denominan "Conjetura simplificada sobre pseudoaleatoriedad cuántica" (Simplified quantum pseudorandomness conjecture), según la cual, la diferencia entre las probabilidades de aceptación de un algoritmo cuántico Q respecto a una distribución uniforme y respecto a una distribución δ-densa es inferior o igual a $\text{poly}(T)\cdot\text{poly}(\delta)$ (T representa el número de queries ejecutadas por el algoritmo de caja negra):

$$\left| P_{X\sim\mathcal{X}}(Q^X = 1) - P_{X\sim\mathcal{U}}(Q^X = 1) \right| \leq \text{poly}(T)\cdot\text{poly}(\delta),$$

[5] Sea $S \subset [n]$, denotaremos $\{\pm 1\}^S$ al conjunto $\{\pm 1\}^{|S|}$ con una reasignación de los índices $[k] \to S$ que mantiene su orden. Por ejemplo, si $S = \{2,3,5\} \subset [5]$, escribiremos las coordenadas de $y \in \{\pm 1\}^S$ con la notación y_2, y_3, y_5, y las de $y' \in \{\pm 1\}^{[5]\setminus S}$ con y_1, y_4.

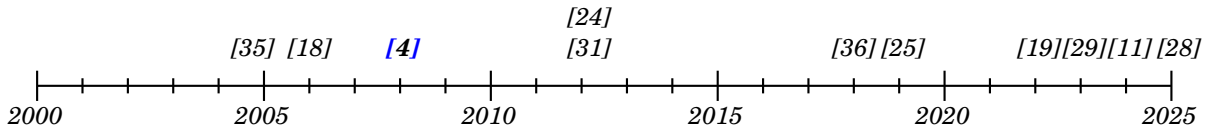
donde $\mathscr{X}$ es una distribución δ-densa, $\mathscr{U}$ uniforme, y $Q^X = 1$ representa la aceptación del input $X \in \{0,1\}^N$ por parte del algoritmo Q.

Dicho de otra forma, ningún algoritmo cuántico de caja negra puede diferenciar en un número reducido de queries entre inputs aleatorios con distribución uniforme respecto a inputs que sigan una distribución suficientemente densa. Esta conjetura había sido demostrada previamente por [42] para algoritmos clásicos probabilísticos de caja negra.

En su trabajo, Liu, Mutreja y Yuen demuestran que, si fuera válida también para algoritmos cuánticos, esta conjetura implicaría la Conjetura Cuántica. Adicionalmente, demuestran una versión débil de esta conjetura en la que la cota no es polinómica sino exponencial respecto a T, resultado análogo a la versión exponencial de la Conjetura AA demostrada por Dinur y Friedgut y que detallamos en la Sección 5.3.

A partir de esta demostración especulan que la conjetura podría ser válida en el caso general y, por lo tanto, serlo también la Conjetura Cuántica.

Como punto final de esta sección, incorporamos una linea temporal relacionando la cronología de los distintos artículos que acabamos de describir (en color azul la Conjetura AA). El principal objetivo es identificar el ritmo de producción de nuevos resultados, y observar que el interés por la Conjetura AA no ha disminuido con el paso de los años. Al contrario: este ritmo parece haberse incrementado recientemente.



CAPÍTULO 6

CONCLUSIONES

Hay diversas cuestiones que resultan fascinantes en relación con la temática tratada en este trabajo. En primer lugar, la computación cuántica en sí misma. Al igual que las leyes de la mecánica cuántica desafían nuestra intuición, su paradigma computacional resulta sorprendente para cualquiera familiarizado con la computación clásica. Como informático de formación y profesión, este modelo de programación me ha llevado a replantear mis conocimientos previos y adoptar una nueva perspectiva. Es particularmente interesante analizar en profundidad algoritmos cuánticos conocidos, sobre todo porque las herramientas necesarias para su estudio son más próximas al ámbito del álgebra lineal que al de la informática tradicional.

Toda persona que se adentre en la programación cuántica advertirá que no todos los problemas tienen sentido en este marco. Sin duda, es posible implementar una operación aritmética mediante computación cuántica, pero no de forma más eficiente que un computador clásico. La Conjetura Cuántica formaliza esta intuición en términos precisos, lo que proporciona un marco teórico que delimita las ventajas potenciales de esta tecnología. La verificación de que la conjetura se cumple en dos algoritmos fundamentales (Deutsch-Jozsa y Simon) permite conectar esta hipótesis abstracta con ejemplos concretos de su aplicabilidad.

Sin embargo, el principal interés de este trabajo radica en las matemáticas subyacentes a estas ideas. En este contexto, el resultado de Beals et al. [10], que establece que la probabilidad de aceptación de un algoritmo cuántico de caja negra puede expresarse como un polinomio multilineal, proporciona las herramientas necesarias para abordar estas cuestiones desde la teoría de funciones booleanas. En este marco surge la Conjetura AA, que abstrae el problema en términos puramente matemáticos. La demostración de que la Conjetura Cuántica puede derivarse de la Conjetura AA constituye la piedra angular sobre la que se sustenta todo lo posterior. En este trabajo hemos reproducido dicha demostración con detalle, lo que completa algunos pasos que no

estaban explícitamente desarrollados en el trabajo original de Aaronson y Ambainis, y en algunos casos profundizando en algunas ideas que no se describían en la demostración original. Además, hemos ampliado el resultado a oráculos de m qubits, circunstancia que no ha sido estudiada en ninguna de las referencias encontradas.

Seguramente, la parte que ha implicado más esfuerzo de este trabajo ha sido el análisis de las demostraciones de casos particulares de la conjetura (el Capítulo 5). No ha sido posible detallarlos todos, ya que su complejidad excedería con creces el alcance y extensión de este trabajo, pero los presentados ilustran técnicas diversas y abarcan múltiples áreas de las matemáticas, como funciones booleanas, probabilidad y estadística, análisis funcional, espacios métricos, teoría de operadores, combinatoria, álgebra lineal y espacios de Hilbert.

En todos los resultados presentados, las demostraciones incluidas se basan en ideas de los trabajos referenciados. Sin embargo, en algunos casos el desarrollo ha diferido significativamente de los planteamientos originales, lo que ha requerido la formulación de técnicas propias. En ningún momento se ha pretendido reinventar la rueda: cada modificación o reinterpretación ha tenido una motivación concreta. En algunas situaciones, las demostraciones referenciadas consistían en aplicar un caso particular de un resultado más general, por lo que se ha optado por diseñar una demostración alternativa más directa. En otras, los argumentos contenían lagunas, pasos poco detallados o aspectos susceptibles de simplificación. Cuando ha sido necesario, se han formulado nuevos lemas, se han reutilizado resultados y conceptos entre secciones, y se ha buscado una mayor coherencia y claridad en la exposición.

En todas estas tareas ha sido de gran valor la ayuda y orientación del tutor, especialmente en la sección correspondiente al caso de la versión exponencial de la conjetura, seguramente la sección que más complejidad ha involucrado y que, sin apoyo, me habría sido imposible resolver. También en la demostración de la equivalencia de conjeturas han sido sumamente enriquecedoras las preguntas que me ha lanzado el tutor, y su ayuda en encontrar las respuestas.

Desde una perspectiva personal, me queda la sensación de estar describiendo un campo en plena evolución. Todo lo relativo a la Conjetura AA es reciente y está sujeto a continuas innovaciones. Como curiosidad, cabe destacar que prácticamente nada de lo tratado en este trabajo existía cuando finalicé mis estudios universitarios en Informática en 1993, y que la mayoría de los resultados expuestos han sido obtenidos en los últimos veinte años. De hecho, en la línea temporal del final de la Sección 5.6 apreciamos que en los últimos años se han seguido generando muchos resultados. Pocas semanas antes del inicio de la elaboración de este trabajo se publicó un artículo con un nuevo enfoque en el estudio de la Conjetura AA y, recientemente, este mismo año, otro planteando una nueva posible línea de demostración de la Conjetura Cuántica.

Por ello, este trabajo no debe entenderse como una síntesis cerrada, sino como una instantánea de un proceso en marcha. La evolución de las ideas matemáticas alrededor de la Conjetura AA y su vinculación con la computación cuántica representa un ejemplo perfecto de cómo conceptos abstractos pueden acabar teniendo un profundo impacto en nuestra comprensión de los límites

del cálculo.

Mirando hacia adelante, resulta tentador preguntarse si en el futuro será posible demostrar la Conjetura AA y la Conjetura Cuántica en toda su generalidad, o si surgirá algún contraejemplo que obligue a reformularlas. Personalmente, apostaría por la primera opción, pero en cualquier caso parece indudable que su estudio seguirá proporcionando conexiones profundas entre disciplinas, desafíos matemáticos estimulantes y una mejor comprensión del potencial y los límites de la computación cuántica.

Bibliografía

[1] Aaronson, S.
Quantum lower bound for the collision problem.
In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing* (2002), pp. 635–642.

[2] Aaronson, S.
The aaronson-ambainis conjecture (2008-2019).
https://scottaaronson.blog/?p=4414, 2019.
The blog of Scott Aaronson.

[3] Aaronson, S.
Research projects in quantum complexity theory.
https://scottaaronson.blog/?p=471, 2019.
The blog of Scott Aaronson.

[4] Aaronson, S., and Ambainis, A.
The need for structure in quantum speedups.
Theory of Computing 10 (11 2009).

[5] Aaronson, S., and Shi, Y.
Quantum lower bounds for the collision and the element distinctness problems.
Journal of the ACM (JACM) 51, 4 (2004), 595–605.

[6] Ambainis, A.
Quantum lower bounds by quantum arguments.
In *Proceedings of the thirty-second annual ACM symposium on Theory of computing* (2000), pp. 636–643.

[7] Ambainis, A.
Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range.
arXiv preprint quant-ph/0305179 (2003).

[8] Arunachalam, S., Briët, J., and Palazuelos, C.

Quantum query algorithms are completely bounded forms.
SIAM Journal on Computing 48, 3 (Jan. 2019), 903–925.

[9] BACKURS, A.
Influences in low-degree polynomials.
https://www.scottaaronson.com/showcase2/report/arturs-backurs.pdf, 12 2012.
The blog of Scott Aaronson.

[10] BEALS, R., BUHRMAN, H., CLEVE, R., MOSCA, M., AND WOLF, R. D.
Quantum lower bounds by polynomials.
Journal of the ACM (JACM) 48, 4 (2001), 778–797.

[11] BHATTACHARYA, S. K.
Aaronson-ambainis conjecture is true for random restrictions.
arXiv preprint arXiv:2402.13952 (2024).

[12] BUHRMAN, H., AND WOLF, R. D.
Complexity measures and decision tree complexity: a survey.
Theoretical Computer Science 288, 1 (2002), 21–43.

[13] CORDERO-ERAUSQUIN, D., AND ESKENAZIS, A.
Talagrand's influence inequality revisited.
Analysis & PDE 16, 2 (2023), 571–612.

[14] DEFANT, A., MASTYŁO, M., AND PÉREZ, A.
On the fourier spectrum of functions on boolean cubes.
Mathematische Annalen 374 (2019), 653–680.

[15] DELÉPINE, D.
Geometría Diferencial I.
Instituto de Física de la Universidad de Guanajuato, 01 2005.

[16] DEUTSCH, D.
Quantum theory, the church–turing principle and the universal quantum computer.
Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 400, 1818 (1985), 97–117.

[17] DEUTSCH, D., AND JOZSA, R.
Rapid solution of problems by quantum computation.
Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 439, 1907 (1992), 553–558.

[18] DINUR, I., FRIEDGUT, E., KINDLER, G., AND O'DONNELL, R.

On the fourier tails of bounded functions over the discrete cube.
In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 2006), STOC '06, Association for Computing Machinery, p. 437–446.

[19] ESCUDERO, F.
Influences of fourier completely bounded polynomials and classical simulation of quantum algorithms.
Chicago Journal of Theoretical Computer Science 2024 (2024), 2.

[20] FEYNMAN, R. P.
Simulating physics with computers.
In *Feynman and computation*. cRc Press, 2018, pp. 133–153.

[21] GÁCS, P.
Lecture notes on descriptional complexity and randomness.
arXiv preprint arXiv:2105.04704 (2021).

[22] GOLUB, G. H., AND LOAN, C. F. V.
Matrix computations.
JHU press, 2013.

[23] GROVER, L. K.
A fast quantum mechanical algorithm for database search.
In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (1996), pp. 212–219.

[24] IVANISHVILLI, P.
Aaronson-ambainis conjecture.
https://extremal010101.wordpress.com/2019/10/29/aaronson-ambainis-conjecture/, 2019.

[25] KELLER, N., AND KLEIN, O.
Quantum speedups need structure.
arXiv preprint arXiv:1911.03748 (2019).

[26] KITAEV, A. Y., SHEN, A., AND VYALYI, M. N.
Classical and quantum computation.
No. 47. American Mathematical Soc., 2002.

[27] KULKARNI, R., AND TAL, A.
On fractional block sensitivity.
Chicago J. Theor. Comput. Sci 8 (2016), 1–16.

[28] LIU, J., MUTREJA, S., AND YUEN, H.
Qma vs. qcma and pseudorandomness.
arXiv preprint arXiv:2411.14416 (2024).

[29] LOVETT, S., AND ZHANG, J.
Fractional certificates for bounded functions.
In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)* (2023), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 84–1.

[30] MARKOV, A.
On a question by D.I. Mendelev, vol. 62.
Zap. Imp. Akad. Nauk. St. Petersburg, 1890.

[31] MONTANARO, A.
Some applications of hypercontractive inequalities in quantum information theory.
Journal of Mathematical Physics 53, 12 (2012).

[32] NIELSEN, M. A., AND CHUANG, I. L.
Quantum Computation and Quantum Information: 10th Anniversary Edition.
Cambridge University Press, 2010.

[33] O'DONNELL, R.
Lecture 26: Influences and decision trees.
https://www.cs.cmu.edu/~odonnell/boolean-analysis/lecture26.pdf, 2007.

[34] O'DONNELL, R.
Analysis of Boolean Functions.
Cambridge University Press, 2021.

[35] O'DONNELL, R., SAKS, M., SCHRAMM, O., AND SERVEDIO, R. A.
Every decision tree has an influential variable.
In *46th annual IEEE symposium on foundations of computer science (FOCS'05)* (2005), IEEE, pp. 31–39.

[36] O'DONNELL, R., AND ZHAO, Y.
Polynomial bounds for decoupling, with applications.
arXiv preprint arXiv:1512.01603 (2015).

[37] PORTUGAL, R.
Basic quantum algorithms.
arXiv preprint arXiv:2201.10574 (2022).

[38] PRESKILL, J.

Lecture notes for ph219/cs219: Quantum information.
Accesible via http://www.theory.caltech.edu/people/preskill/ph229 1997 (2015).

[39] SAVAGE, J. E.
Models of computation, vol. 136.
Addison-Wesley Reading, MA, 1998.

[40] SHOR, P. W.
Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.
SIAM review 41, 2 (1999), 303–332.

[41] UNRUH, D.
Random oracles and auxiliary input.
In *Annual International Cryptology Conference* (2007), Springer, pp. 205–223.

[42] UNRUH, D.
Random oracles and auxiliary input.
In *Advances in Cryptology - CRYPTO 2007* (Berlin, Heidelberg, 2007), A. Menezes, Ed., Springer Berlin Heidelberg, pp. 205–223.

[43] WOOTTERS, W. K., AND ZUREK, W. H.
A single quantum cannot be cloned.
Nature, 299 (May 1982), 802–803.

[44] YUVAL FILMUS HAMED HATAMI STEVEN HEILMAN ELCHANAN MOSSEL RYAN O'DONNELL SUSHANT SACHDEVA ANDREW WAN, K. W.
Real analysis in computer science: A collection of open problems.
https://simons.berkeley.edu/sites/default/files/openprobsmerged.pdf, 2014.