Materia optativa

Fundamentos de lenguajes para computación cuántica

Licenciatura en Ciencias de la Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Apunte de clase

Alejandro Díaz-Caro
Universidad Nacional de Quilmes
& Instituto de Investigación en Ciencias de la Computación (CONICET / UBA)

Versión (incompleta) del 15 de noviembre de 2022

Enfoque de este apunte

Estas notas están basadas en cursos que dí en diferentes lugares: Escuela de Ciencias Informáticas (UBA), Congreso Argentino de Ciencias de la Computación, Escuela de Verano de Río Cuarto, y materias optativas de cuántica en la Licenciatura en Ciencias de la Computación (LCC) de la UNR y la Licenciatura en Informática (LI) de la UNQ, así como materias obligatorias para la LI de la UNQ (Características de Lenguajes de Programacion, Lógica y Programación). Están pensadas para estudiantes de grado y posgrado de computación, no de física. Es por ello que el enfoque que se da es casi puramente matemático, con algún comentario aquí y allá de la física que motiva el formalismo, pero todos los razonamientos se realizan exclusivamente desde el lado de la matemática. De todas maneras, el curso contendrá cosas no incluidas en estos apuntes, y no profundizará en otros temas que sí están tratados más profundamente aquí. Se recomienda recurrir a la bibliografía sugerida.

Curso:
Fundamentos de lenguajes para computación cuántica

Alejandro Díaz-Caro

© 2015–2022 Creative Commons Attibution 4.0 Internacional.
Podés ver una copia de la licencia en http://creativecommons.org/licenses/by/4.0/.

Índice general

Capítulo 0

Organización de la materia

Cronograma

Esta materia se dictará los **viernes de 16 a 20hs** con el siguiente cronograma.

Fecha	Parte	Temas
26 de agosto	I	Introducción a la computación cuántica
~~2 de septiembre~~		*Feriado nacional decretado por intento de magnicidio*
9 de septiembre	II	El isomorfismo de Curry-Howard
16 de septiembre	I y II	Práctica
23 de septiembre		*Clase suspendida*
30 de septiembre	II	Introducción a la teoría de categorías
~~*7 de octubre*~~		*Feriado nacional*
14 de octubre	II	Semántica denotacional y lógica lineal intuicionista
21 de octubre	II	Práctica
~~*28 de octubre*~~		*Sin clases*
4 de noviembre	III	Curry-Howard en computación cuántica
11 de noviembre	III	Práctica & consulta
18 de noviembre		Examen final.

Bibliografía de referencia

La materia se encuentra resumida en este apunte. Sin embargo, si se quiere profundizar en alguno de los temas, se puede recurrir a la bibliografía de referencia que se sugiere en esta sección.

Parte I

- Michael Nielsen e Isaac Chuang. "Quantum computation and quantum information". Cambridge University Press. 2a edición, 2010.
- Noson S. Yanofsky y Mirco A. Mannucci. "Quantum computing for computer scientists". Cabridge University Press. 2008.

Parte II

Capítulo 4

- Gilles Dowek y Jean-Jacques Lévy. "Introduction to the theory of programming languages". Springer. 2011.
- Morten H. B. Sørensen y Paweł Urzyczyn. "Lectures on the Curry-Howard isomorphism". Elsevier. 2006.
- Jean-Yves Girard, Paul Taylor e Yves Lafont. "Proof and types". Cambridge University Press. 1989.
- Henk Barendregt, Wil Dekkers y Richard Statman. "Lambda calculi with types". Cambridge University Press. 2013.
- Mauricio Ayala-Rincón y Flávio L. C. de Moura. "Applied logic for computer scientists: Computational deduction and formal proofs". Springer. 2016.

Capítulo 5

- Benjamin C. Pierce. "Basic Category Theory for Computer Scientists". MIT Press. 1991.
- Roy L. Crole. "Categories for types". Cambridge University Press. 1993.
- Joachim Lambek y Philip J. Scott "Introduction to higher order categorical logic". Cambridge University Press. 1988.

Capítulo 6

- Frank Pfenning. "Linear logic". Apuntes de curso en la Carnegie Mellon University. 2002.

Parte III

Libros

- Bob Coecke y Aleks Kissinger. "Picturing quantum processes: A first course in quantum theory and diagrammatic reasoning". Cambridge University Press. 2017.
- Simon Gay and Ian Mackie. "Semantic Techniques in Quantum Computation". Cambridge University Press. 2009.

Papers

- Peter Selinger. Towards a quantum programming language. Mathematical Structures in Computer Science 14(4):527-586, 2004.
- Peter Selinger y Benoît Valiron. A lambda calculus for quantum computation with classical control. Mathematical Structures in Computer Science 16(3):527–552, 2006.
- Alejandro Díaz-Caro, Mauricio Guillermo, Alexandre Miquel y Benoît Valiron. Realizability in the unitary sphere. Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2019), 1–13, 2019.
- Alejandro Díaz-Caro, Gilles Dowek y Juan Pablo Rinaldi. Two linearities for quantum computing in the lambda calculus. Biosystems 186:104012, 2019.
- Alejandro Díaz-Caro y Octavio Malherbe. A categorical construction for the computational definition of vector spaces. Applied Categorical Structures 28(5):807-844, 2020.
- Alejandro Díaz-Caro y Gilles Dowek. A new connective in natural deduction, and its application to quantum computing. Theoretical Aspects of Computing (ICTAC 2021). Lecture Notes in Computer Science, 12819:175–193, 2021.
- Alejandro Díaz-Caro. A quick overview on the quantum control approach to the lambda calculus. Logical and Semantic Frameworks with Applications (LSFA'21). Electronic Proceedings in Theoretical Computer Science, 357:1–17, 2021.
- Alejandro Díaz-Caro y Octavio Malherbe. Quantum control in the unitary sphere: Lambda-S1 and its categorical model. Logical Methods in Computer Science (en prensa), 2022. https://arxiv.org/abs/2012.05887.
- Alejandro Díaz-Caro y Gilles Dowek. Linear lambda-calculus is linear. Formal Structures for Computation and Deduction (FSCD'22). Leibniz International Proceedings in Informatics (LIPIcs) 228:21, 2022.
- Alejandro Díaz-Caro y Octavio Malherbe. Semimodules and the (syntactically-)linear lambda calculus. Borrador enviado a revisión. https://arxiv.org/abs/2205.02142, 2022.

Parte I

Computación cuántica

Capítulo 1

Introducción a la computación cuántica

> *I feel that a deep understanding of why quantum algorithms work is still lacking. Surely the power of quantum computers has something to do with entanglement, quantum parallelism, and the vastness of Hilbert space, but I think that it should be possible to pinpoint more precisely the true essence of the matter.*
>
> John Preskill [1998]

1.1. Introducción

La computación cuántica, una rama de las ciencias de la computación teórica, tiene su origen en la física, y más precisamente en el físico estadounidense Richard Feynman, quien en 1981 dedicó una charla en el Massachusetts Institute of Technology (MIT) al problema de la simulación de la física cuántica con computadoras clásicas. Sus ya célebres palabras finales resumen su frustración de ese entonces:

> *And I'm not happy with all the analyses that go with just the classical theory, because nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy. Thank you.*
>
> (ver, por ejemplo, [Brown, 2001, pp.100])

Esta provocación, lejos de plantear soluciones, abrió las puertas a interrogantes nunca antes concebidos. ¿Qué ganancia se lograría si las computadoras fuesen regidas por las leyes de la mecánica cuántica? Fueron los algoritmos de Grover [1996] y Shor [1997] los cuales despertaron el gran interés desde las ciencias de la computación en este nuevo paradigma. El primero es un algoritmo de búsqueda sobre registros desordenados, el cual provee una ganancia cuadrática de complejidad temporal frente a cualquier algoritmo clásico conocido. El segundo es un algoritmo para la factorización de números, con una ganancia exponencial.

Actualmente existen muchas áreas de investigación dentro de la computación cuántica. Por ejemplo, desde un punto de vista práctico se plantea el problema de construir el hardware de una computadora cuántica. Desde sus orígenes, en las palabras de Feinmann,

la idea es que un algoritmo cuántico sea una simulación cuántica en hardware que se comporta de acuerdo a las leyes de la física cuántica. Es decir que un experimento cuántico en un laboratorio, puede considerarse como un algoritmo. O dicho de otro modo: podemos describir el comportamiento de un sistema cuántico a través de un algoritmo. La pregunta es, ¿podemos realizar el experimento cuántico que describe un algoritmo dado? Allí es donde se manifiesta el desafío técnico.
Otra área es la de desarrollar algoritmos que obtengan una ganancia con respecto a su contraparte clásica. En general los algoritmos de Grover y Shor mencionados anteriormente se consideran como los ejemplos canónicos de aceleración obtenida gracias a la computación cuántica. Muchos otros algoritmos cuánticos son derivados de ellos. La pregunta aquí es ¿qué otros algoritmos podemos obtener que nos den una ganancia respecto a los algoritmos clásicos?
Otra rama de investigación es la del diseño de lenguajes de programación que permitan expresar los algoritmos cuánticos de una manera amigable, y quizá permitiendo descubrir nuevos algoritmos al tener una herramienta de alto nivel para pensarlos.
Desde un punto de vista más fundamental, y como lo expresara Preskill en la cita que abre este capítulo, los fundamentos lógicos detrás de la computación cuántica, siguen siendo un misterio. Si bien existe una lógica cuántica [Birkhoff y von Neumann, 1936], ésta fue propuesta muchos años antes de la computación cuántica, por lo que encontrar la correspondencia entre computación y lógica cuántica no es trivial. Esta área tiene muchas subáreas con metodologías diferentes. En particular, el estudio de semántica de lenguajes de programación sigue este objetivo. En este caso no se persigue el estudio del lenguaje en sí mismo, sino que el objetivo es el estudio de la lógica subyacente. Estudiar la lógica detrás de la computación cuántica implica estudiar la lógica detrás de la física cuántica, lo cual puede tener influencia en el desarrollo de nuevas teorías sobre el mundo que nos rodea.

En este curso nos interesa este último aspecto: el estudio de propiedades de lenguajes de programación que nos acerquen hacia una lógica computacional de la física cuántica.

1.2. Preliminares: un poco de álgebra

1.2.1. Espacio de Hilbert

TL;DR $\mathbb{C}^n$ con la suma $(+)$ y el producto $(\cdot)$ usuales, y el producto escalar definido por

$$\langle \vec{v}, \vec{w} \rangle = \langle (v_1, v_2, \ldots, v_n), (w_1, w_2, \ldots, w_n) \rangle = \sum_{i=1}^{n} v_i^* \cdot w_i$$

donde v^* es el complejo conjugado de v, es un *espacio de Hilbert*.

En el resto de la sección se define formalmente qué es un espacio de Hilbert.

Definición 1.1 (Producto escalar) Sea E un espacio vectorial sobre el cuerpo $\mathbb{K}$ ($\mathbb{R}$ o $\mathbb{C}$). Un producto escalar (también llamado producto interno) definido sobre E es una función $\langle , \rangle : E \times E \to \mathbb{K}$ que verifica las siguientes propiedades.

Para todo $\vec{u}, \vec{v}, \vec{w} \in E$, $a, b \in \mathbb{K}$, se cumple:

$$\begin{cases} \langle \vec{u}, \vec{u} \rangle \geq 0 \\ \langle \vec{u}, \vec{u} \rangle = 0 \Leftrightarrow \vec{u} = \vec{0}_E \end{cases} \qquad \text{(Definida positiva)}$$

$$\langle \vec{w}, a\vec{u} + b\vec{v} \rangle = a\langle \vec{w}, \vec{u} \rangle + b\langle \vec{w}, \vec{v} \rangle \qquad \text{(Lineal por derecha)}$$

$$\langle a\vec{u} + b\vec{v}, \vec{w} \rangle = a^*\langle \vec{u}, \vec{w} \rangle + b^*\langle \vec{v}, \vec{w} \rangle \qquad \text{(Antilineal por izquierda)}$$

$$\langle \vec{u}, \vec{v} \rangle = \langle \vec{v}, \vec{u} \rangle^* \qquad \text{(Hermítica)}$$

Definición 1.2 (Espacio pre-Hilbert) Un espacio pre-Hilbert es un espacio vectorial sobre $\mathbb{K}$ con producto escalar.

Observación. Todo espacio pre-Hilbert es un espacio vectorial normado con la norma

$$\|\vec{v}\| = \sqrt{\langle \vec{v}, \vec{v} \rangle}$$

Definición 1.3 (Sucesión de Cauchy) Sea $\vec{v}_n$ una sucesión de vectores del espacio E. Si $\|\vec{v}_n - \vec{v}_m\| \to 0$ cuando $n, m \to \infty$, entonces la sucesión $\vec{v}_n$ es una sucesión de Cauchy. (Esto quiere decir que puedo hacer distar entre sí los términos tan poco como quiera).

Observación. Toda sucesión convergente es de Cauchy, pero no toda sucesión de Cauchy es convergente.

Definición 1.4 (Espacio completo) E es completo para la norma $\|\cdot\|$, si y sólo si toda sucesión de Cauchy converge con esa norma.

Definición 1.5 (Espacio de Hilbert) Un espacio pre-Hilbert completo en su norma se denomina espacio de Hilbert.

1.2.2. Productos tensoriales

En esta sección consideramos espacios vectoriales equipados con una base canónica.

Definición 1.6 (Producto tensorial) Sean E y F dos espacios vectoriales con bases canónicas $B = \{\vec{b}_i \mid i \in I\}$ y $C = \{\vec{c}_j \mid j \in J\}$ respectivamente. El producto tensorial $E \otimes F$ de E y F es el espacio vectorial de base canónica $\{\vec{b}_i \otimes \vec{c}_j \mid i \in I \text{ y } j \in J\}$, donde $\vec{b}_i \otimes \vec{c}_j$ es el par ordenado formado por el vector $\vec{b}_i$ y el vector $\vec{c}_j$. La operación $\otimes$ se extiende a vectores de E y F bilinearmente:

$$(\sum_i \alpha_i \vec{b}_i) \otimes (\sum_j \beta_j \vec{c}_j) = \sum_{ij} \alpha_i \beta_j (\vec{b}_i \otimes \vec{c}_j)$$

Definición 1.7 (Producto cartesiano entre dos subconjuntos de espacios vectoriales) Sean E y F dos espacios vectoriales equipados con bases B y C, y sean S y T dos subconjuntos de E y F respectivamente. Definimos el conjunto $S \times T$, subconjunto del espacio vectorial $E \otimes F$, de la siguiente manera:

$$S \times T = \{\vec{u} \otimes \vec{v} \mid \vec{u} \in S, \vec{v} \in T\}$$

Observación. $E \times F \neq E \otimes F$. Por ejemplo, si $E = F = \mathbb{C}^2$, con base canónica $\{\vec{i}, \vec{j}\}$, entonces $E \times F$ contiene a $\vec{i} \otimes \vec{i}$ y a $\vec{j} \otimes \vec{j}$, pero no a $\vec{i} \otimes \vec{i} + \vec{j} \otimes \vec{j}$, que no es producto de dos vectores de $\mathbb{C}^2$.

Definición 1.8 (Generador) Sea E un espacio vectorial equipado con una base B, y $S \subseteq E$. Escribimos $\mathcal{G}(S)$ al espacio vectorial sobre $\mathbb{C}$ generado por S, es decir, que contiene todas las combinaciones lineales de elementos de S.

Observación. Si E y F son dos espacios vectoriales con bases B y C respectivamente, entonces

$$E \otimes F = \mathcal{G}(B \times C) = \mathcal{G}(E \times F)$$

La operación $\otimes$ introducida genéricamente en la Definición 1.6, puede ser definida más precisamente para matrices (y vectores, tomando matrices columna o fila) de la siguiente manera.

Definición 1.9 (Producto tensorial entre matrices) El producto tensorial de dos matrices, P y Q se define como la matriz

$$P \otimes Q = \begin{pmatrix} p_{11}Q & \dots & p_{1m}Q \\ \vdots & & \vdots \\ p_{n1}Q & \dots & p_{nm}Q \end{pmatrix}$$

Ejemplo 1.10

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \otimes \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} & 2\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \\ 3\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} & 4\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 1\begin{pmatrix} 3 \\ 4 \end{pmatrix} \\ 2\begin{pmatrix} 3 \\ 4 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 6 \\ 8 \end{pmatrix}$$

Observación. El producto escalar, o producto interno, entre dos vectores nos da un número. El producto tensorial, o producto externo, entre dos vectores nos da un vector de mayor dimensión.

Como se dijo anteriormente, $E \times F \neq E \otimes F$, y por lo tanto:

Existen vectores de $E \otimes F$ que no son producto tensorial entre uno de E y uno de F.

Ejemplo 1.11 Consideremos el espacio $\mathbb{C}^2 \otimes \mathbb{C}^2$. Una base de $\mathbb{C}^2$ es $\{\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}\}$ Por lo tanto

$$\mathbb{C}^2 \otimes \mathbb{C}^2 = \text{Gen}(\left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right\}) = \mathbb{C}^4$$

Tomemos $\vec{v} = (\alpha, 0, 0, \beta)^T$, con $\alpha, \beta \neq 0$. Es fácil verificar que $\vec{v} \in \mathbb{C}^4$. Sin embargo, no existen $\vec{v}_1, \vec{v}_2 \in \mathbb{C}^2$ tal que $\vec{v} = \vec{v}_1 \otimes \vec{v}_2$.

Demostración. Supongamos que existen $\vec{v}_1$ y $\vec{v}_2$ tales que $\vec{v}_1 \otimes \vec{v}_2 = \vec{v}$, entonces

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} = \begin{pmatrix} \alpha \\ 0 \\ 0 \\ \beta \end{pmatrix} \Rightarrow \begin{cases} ac = \alpha \\ ad = 0 \\ bc = 0 \\ bd = \beta \end{cases}$$

pero este es un sistema que no tiene solución. □

1.2.3. Notación bra–ket

Notación introducida por Paul Dirac [1939] para describir estados cuánticos.

1.2.3.1. Notación bra y ket para vectores

En lugar de escribir los vectores como $\vec{v}$ la notación ket usa $|v\rangle$.
En particular definimos:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Por lo tanto, cualquier vector de $\mathbb{C}^2$ puede escribirse como

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha|0\rangle + \beta|1\rangle$$

Podemos, por ejemplo, definir vectores como los siguientes

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \qquad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

y como estos son dos vectores ortogonales (por ende, forman una base), ahora es posible también escribir cualquier vector de $\mathbb{C}^2$ como combinación lineal de $|+\rangle$ y $|-\rangle$.
Por ejemplo:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha|0\rangle + \beta|1\rangle = \frac{1}{\sqrt{2}}(\alpha + \beta)|+\rangle + \frac{1}{\sqrt{2}}(\alpha - \beta)|-\rangle$$

Observación. Al menos que se indique lo contrario, en el resto del apunte consideraremos el espacio complejo de dimensión $N = 2^n$, $\mathbb{C}^N = \mathbb{C}^{2^n}$.

Definición 1.12 (Bra y Ket) Llamamos ket a un vector de la forma

$$|\psi\rangle = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix}$$

y bra a un vector de la forma

$$\langle\psi| = (\alpha_1^*, \ldots, \alpha_N^*)$$

donde $\alpha_i \in \mathbb{C}$ y α_i^* denota el conjugado de α_i.

Observaciones.

- Haciendo un abuso de notación, podemos escribir vectores como el siguiente:

$$|\alpha_1\psi_1 + \alpha_2\psi_2\rangle = \alpha_1|\psi_1\rangle + \alpha_2|\psi_2\rangle$$

- A partir la definición de bras y kets, llamamos "braket" al producto escalar:

$$\langle\psi|\phi\rangle = (\alpha_1^*, \ldots, \alpha_N^*)\begin{pmatrix}\beta_1\\ \vdots \\ \beta_N\end{pmatrix} = a \in \mathbb{C}$$

- Recordatorio de álgebra: Una *base ortonormal* de un espacio vectorial normado es una base donde todos los vectores tienen norma 1. Además, en una base, todos los vectores son ortogonales entre sí (es decir, el producto escalar entre ellos es 0). Por lo tanto:

  Dado un conjunto $B = \{|u_1\rangle, \ldots, |u_N\rangle\}$, B es una base ortonormal de $\mathbb{C}^N$ si y sólo si para todo i, j tenemos $\langle u_i|u_j\rangle = \delta_{ij}$, donde δ_{ij} es la delta de Kronecker (igual a 1 si $i = j$, y 0 en otro caso).

- Entonces, todo Ket $|\psi\rangle$ se puede expresar como $|\psi\rangle = \sum_{i=1}^N a_i|u_i\rangle$.

- Si tomamos la base canónica de $\mathbb{C}^N$, con $|u_i\rangle$ el vector i-ésimo de dicha base, podemos calcular la componente i-ésima de un vector cualquiera de la siguiente manera:

$$\langle u_i|\psi\rangle = \langle u_i|\sum_{j=1}^N a_j|u_j\rangle = \sum_{j=1}^N a_j \underbrace{\langle u_i|u_j\rangle}_{\delta_{ij}} = a_i$$

Teorema 1.13 Sea $B = \{|u_1\rangle, \ldots, |u_N\rangle\}$ una base ortonormal, entonces $\sum_{i=1}^N |u_i\rangle\langle u_i| = I$.

Demostración.

$$\left(\sum_{i=1}^N |u_i\rangle\langle u_i|\right)|\psi\rangle = \left(\sum_{i=1}^N |u_i\rangle\langle u_i|\right)\left(\sum_{j=1}^N a_j|u_i\rangle\right)$$
$$= \sum_{i=1}^N\sum_{j=1}^N a_j|u_i\rangle\underbrace{\langle u_i|u_j\rangle}_{\delta_{ij}} = \sum_{i=1}^N a_i|u_i\rangle = |\psi\rangle \qquad \square$$

Observaciones.

- Análogamente a los kets, todo bra $\langle\phi|$ puede ser descompuesto como $\langle\phi| = \sum_{i=1}^N b_i^*\langle u_i|$.

- Podemos ver que $b_i^* = \langle\phi|u_i\rangle \in \mathbb{C}$ ya que

$$\langle\phi| = \langle\phi|\underbrace{\left[\sum_{i=1}^N |u_i\rangle\langle u_i|\right]}_{I} = \sum_{i=1}^N \langle\phi|u_i\rangle\langle u_i| \quad \Rightarrow \quad b_i^* = \langle\phi|u_i\rangle$$

Observación. De aquí en más, trabajaremos sólo con los vectores normalizados de $\mathbb{C}^N$ (es decir, vectores cuya norma es 1). Esto es

$$1 = \|\psi\|^2 = \langle\psi|\psi\rangle = \left(\sum_{j=1}^{N} a_j^* \langle u_j|\right)\left(\sum_{i=1}^{N} a_i |u_i\rangle\right) = \sum_{i,j=1}^{N} a_j^* a_i \underbrace{\langle u_j|u_i\rangle}_{\delta_{ij}} = \sum_{i=1}^{N} |a_i|^2 = 1$$

Es decir, trabajamos con vectores cuya suma de los módulos al cuadrado de sus componentes es 1.

Propiedad

$$\langle ab|cd\rangle = \sum_{i=1}^{N} a_i^* c_i \sum_{j=1}^{N} b_j^* d_j = \langle a|c\rangle\langle b|d\rangle \quad \forall a, b, c, d \in \mathbb{C}^N$$

1.2.3.2. Notación bra y ket para matrices

Para toda matriz cuadrada de dimensión N a coeficientes complejos A, tenemos la siguiente representación:

$$A = \left(\underbrace{\sum_{i=1}^{N} |u_i\rangle\langle u_i|}_{I}\right) A \left(\underbrace{\sum_{j=1}^{N} |u_j\rangle\langle u_j|}_{I}\right) = \sum_{i=1}^{N}\sum_{j=1}^{N} |u_i\rangle \underbrace{\langle u_i| \overbrace{A|u_j\rangle}^{|vector\rangle}}_{\alpha_{ij}} \langle u_j| = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{ij}|u_i\rangle\langle u_j|$$

donde α_{ij} es la componente ij de la matriz.
Con esta representación, podemos representar el producto de una matriz por un vector de la siguiente manera:

$$\begin{aligned} A|\psi\rangle &= \left(\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{ij}|u_i\rangle\langle u_j|\right)\left(\sum_{k=1}^{N} a_k|u_k\rangle\right) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_{ij}a_k|u_i\rangle \underbrace{\langle u_j|u_k\rangle}_{\delta_{jk}} = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{ij}a_j|u_i\rangle \end{aligned}$$

Es decir, las componentes del vector $A|\psi\rangle$ son $b_i = \sum_{j=1}^{N} \alpha_{ij}a_j$.

1.3. Bits cuánticos y operadores

1.3.1. Primera intuición

En computación clásica la unidad mínima de información es el bit, el cual puede estar en un estado 0 o 1. Leer un bit es una operación que no conlleva ninguna particularidad. En contraposición, un bit cuántico o qubit puede estar en un estado que sea una superposición de los estados 0 y 1. Un qubit es un vector de $\mathbb{C}^2$, por lo tanto lo podemos representar como $\alpha|0\rangle + \beta|1\rangle$, lo cual representa el estado que es 0 y en 1 a la vez. Leer un qubit en cambio se produce a través de una operación llamada medición, y al medir un qubit, éste colapsa, cambia su estado (dependiendo de la medición puede cambiar por ejemplo a $|0\rangle$ o $|1\rangle$, pero también podría usarse otro operador de medición que lo colapse a otra base).

1.3.2. Bits cuánticos

Definición 1.14 (Qubit) Un qubit o bit cuántico es un vector normalizado (es decir, con norma 1) del espacio de Hilbert $\mathbb{C}^2$.

Observación. Considerando la base $\{|0\rangle, |1\rangle\}$ de $\mathbb{C}^2$, cualquier qubit puede escribirse como $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, con $|\alpha|^2 + |\beta|^2 = 1$.

Definición 1.15 (n-qubits) Un sistema de n-qubits es un vector normalizado del espacio $\mathbb{C}^{2^n} = \bigotimes_{i=1}^{n} \mathbb{C}^2$.

Observaciones.

- En lugar de escribir $|0\rangle \otimes |1\rangle \otimes \cdots \otimes |0\rangle$ escribimos $|01\ldots0\rangle$.
- De la misma manera, en ocasiones, en lugar de $|0\rangle \otimes (\alpha|0\rangle + \beta|1\rangle)$ escribimos simplemente $|0\rangle(\alpha|0\rangle + \beta|1\rangle)$.
- La base canónica del espacio $\mathbb{C}^{2^n}$ es $\{|0\ldots00\rangle, |0\ldots01\rangle, \ldots, |1\ldots11\rangle\}$.

Un algoritmo cuántico consiste en la evolución (Definición 1.24) de un sistema representado por n–qubits.

1.3.3. Operadores

Definición 1.16 (Operador) Un operador de $\mathbb{C}^N$ es una matriz cuadrada de dimensión N a coeficientes complejos.

Definición 1.17 (Adjunto) El adjunto de un operador A se nota por $A^\dagger$ y se define como el operador transpuesto y conjugado de A. Es decir, si $\alpha_{ij} = \langle u_i|A|u_j\rangle$ son las componentes de A, las componentes de $A^\dagger$ son $\alpha^*_{ji} = \langle u_j|A|u_i\rangle^* = \langle u_i|A^\dagger|u_j\rangle$.

Propiedades Sean A y B operadores de $\mathbb{C}^N$, $a \in \mathbb{C}$ y $|\psi\rangle \in \mathbb{C}^N$

- $(A^\dagger)^\dagger = A$
- $(aA)^\dagger = a^* A^\dagger$
- $\langle A\psi| = \langle\psi|A^\dagger$
- $(A+B)^\dagger = A^\dagger + B^\dagger$
- $(AB)^\dagger = B^\dagger A^\dagger$

Definición 1.18 (Proyector) A los operadores de la forma $P = |\phi\rangle\langle\phi|$ se les llama proyectores, ya que proyecta ortogonalmente un ket $|\psi\rangle$ cualquiera sobre el ket $|\phi\rangle$:

$$P|\psi\rangle = |\phi\rangle \underbrace{\langle\phi|\psi\rangle}_{a\in\mathbb{C}} = a|\phi\rangle$$

Ejemplo 1.19 Tomemos la base $\{|0\rangle, |1\rangle\}$, con $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$ y $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$. Un vector $|\psi\rangle$ cualquiera puede escribirse como $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Por lo tanto

$$|0\rangle\langle 0|\psi\rangle = |0\rangle\langle 0|(\alpha|0\rangle + \beta|1\rangle) = |0\rangle(\alpha \underbrace{\langle 0|0\rangle}_{1} + \beta \underbrace{\langle 1|0\rangle}_{0}) = \alpha|0\rangle$$

Definición 1.20 (Operador hermítico) Un operador A es hermítico si $A = A^\dagger$.

Observación. Si es hermítico, su diagonal debe ser real, ya que $\alpha_{ij} = \alpha_{ji}^*$, por lo tanto $\alpha_{ii} = \alpha_{ii}^*$.

Definición 1.21 (Operador unitario) Un operador U es unitario si $U^\dagger U = UU^\dagger = I$, o lo que es lo mismo $U^\dagger = U^{-1}$.

Propiedades Para cualquier operador U unitario vale:

- U preserva el producto interno: $\langle U\phi|U\psi\rangle = \langle\phi|U^\dagger U|\psi\rangle = \langle\phi|\psi\rangle$
- U^{-1} es unitario.
- Si $\{|\psi_1\rangle, \ldots, |\psi_N\rangle\}$ es base ortonormal, entonces $\{U|\psi_1\rangle, \ldots, U|\psi_N\rangle\}$ también lo es.

Definición 1.22 (Operador de medición) Un conjunto de proyectores $\{M_1, \ldots, M_k\}$ se dice que es un operador de medición si satisface

$$\sum_{i=1}^{k} M_i M_i^\dagger = I$$

Definición 1.23 (Compuertas cuánticas) A los operadores unitarios se les llama compuertas cuánticas, como analogía a las compuertas lógicas de la computación clásica, ya que serán esos los que se utilizan para realizar el cómputo.

Observación. La mayoría de las compuertas cuánticas que usaremos a lo largo del curso serán además de operadores unitarios, hermíticos, por lo que coinciden con su inversa.

Definición 1.24 (Evolución) Se dice que un sistema representado por un ket $|\psi\rangle$ evoluciona al sistema $|\phi\rangle$, cuando se realiza una de las siguientes operaciones:

- Se premultiplica por una compuerta cuántica U:

$$|\phi\rangle = U|\psi\rangle$$

- Se aplica un operador de medición $M = \{M_1, \ldots, M_k\}$ de la siguiente manera:

$$|\phi\rangle = \frac{M_i|\psi\rangle}{\sqrt{\langle\psi|M_i^\dagger M_i|\psi\rangle}} \text{ para algún } 1 \leq i \leq k$$

  La elección del M_i no se conoce de antemano, sólo se conoce la probabilidad para cada i, la cual viene dada por la siguiente ley:

$$p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle$$

Observaciones.

- Usaremos también la notación $|\psi\rangle \xrightarrow{U} |\phi\rangle$ o $|\psi\rangle \xrightarrow{M} |\phi\rangle$ para indicar que el ket $|\psi\rangle$ evoluciona al ket $|\phi\rangle$.

- Cuando se quiera hacer evolucionar sólo un qubit de un sistema de n-qubits, digamos el qubit i, se premultiplica tensorialmente $i-1$ veces y se postmultiplica $n-i$ veces la compuerta a aplicar por la matriz identidad. Ejemplo: U aplicada al segundo qubit de un sistema de 2–qubits, será la compuerta $I \otimes U$.

Ejemplo 1.25 Consideramos el operador medición de $\{M_0, M_1\}$ con

$$M_0 = |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \qquad M_1 = |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Podemos verificar que $M_0 M_0^\dagger + M_1 M_1^\dagger = M_0 + M_1 = I$, y por lo tanto es un operador de medición.
Sea $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, entonces, la probabilidad de que el proyector que se aplique sea M_0 es

$$\begin{aligned} p(0) &= \langle\psi|M_0^\dagger M_0|\psi\rangle \\ &= (\alpha^*\langle 0| + \beta^*\langle 1|)M_0(\alpha|0\rangle + \beta|1\rangle) \\ &= |\alpha|^2\langle 0|M_0|0\rangle + \alpha^*\beta\langle 0|M_0|1\rangle + \alpha\beta^*\langle 1|M_0|0\rangle + |\beta|^2\langle 1|M_0|1\rangle \\ &= |\alpha|^2\underbrace{\langle 0|0\rangle}_{1}\underbrace{\langle 0|0\rangle}_{1} + \alpha^*\beta\underbrace{\langle 0|0\rangle}_{1}\underbrace{\langle 0|1\rangle}_{0} + \alpha\beta^*\underbrace{\langle 1|0\rangle}_{0}\underbrace{\langle 0|0\rangle}_{1} + |\beta|^2\underbrace{\langle 1|0\rangle}_{0}\underbrace{\langle 0|1\rangle}_{0} \\ &= |\alpha|^2 \end{aligned}$$

Análogamente, $p(1) = \langle\psi|M_1^\dagger M_1|\psi\rangle = \cdots = |\beta|^2$.
Dado que el vector está normalizado, $p(0) + p(1) = |\alpha|^2 + |\beta|^2 = \|\psi\| = 1$.
Luego de aplicar este operador de medición, la evolución es la siguiente. Si se aplicó el proyector M_0, el sistema queda en el siguiente estado:

$$\frac{M_0|\psi\rangle}{\sqrt{\langle\psi|M_0^\dagger M_0|\psi\rangle}} = \frac{M_0|\psi\rangle}{\sqrt{p(0)}} = \frac{\alpha}{|\alpha|}|0\rangle$$

Este estado está normalizado ya que $\left|\frac{\alpha}{|\alpha|}\right|^2 = \frac{|\alpha|^2}{|\alpha|^2} = 1$.

Análogamente si se aplicó M_1 se obtiene $\frac{M_1|\psi\rangle}{\sqrt{p(1)}} = \frac{\beta}{|\beta|}|1\rangle$.

Definición 1.26 (Compuertas más comunes y operadores de Pauli) Las compuertas cuánticas más importantes, por su utilidad en el diseño de algoritmos, son las siguientes:

- La compuerta H de Hadamard:

$$\begin{aligned} H|0\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \qquad \text{donde:} \quad H = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

- La identidad I:

$$\begin{aligned} I|0\rangle &= |0\rangle \\ I|1\rangle &= |1\rangle \end{aligned} \qquad \text{donde:} \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- La negación X:

$$\begin{matrix} X|0\rangle = |1\rangle \\ X|1\rangle = |0\rangle \end{matrix} \quad \text{donde:} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

- El cambio de fase Z:

$$\begin{matrix} Z|0\rangle = |0\rangle \\ Z|1\rangle = -|1\rangle \end{matrix} \quad \text{donde:} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- La No-controlada $CNOT$:

$$\begin{matrix} CNOT|0x\rangle = |0x\rangle \\ CNOT|1x\rangle = |1\rangle \otimes X|x\rangle \end{matrix} \quad \text{donde:} \quad CNOT = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix}$$

En particular, las matrices I, X, iXZ y Z son las llamadas *matrices de Pauli* en honor a Wolfgang Pauli

1.4. Teorema del no-clonado

El teorema de no-clonado [Wootters y Zurek, 1982] dice que es imposible construir una máquina universal de clonado. Es decir, no podemos copiar un qubit arbitrario, ya que no existe ningún método que pueda copiarlo sin saber su estado preciso, y como la medición cambia el qubit, no podemos saber su estado preciso. En consecuencia, no podemos copiar un qubit arbitrario.

Teorema 1.27 (No-cloning) No existe ninguna compuerta cuántica U tal que para algún $|\phi\rangle \in \mathbb{C}^N$ y $\forall|\psi\rangle \in \mathbb{C}^N$ se cumpla $U|\psi\phi\rangle = |\psi\psi\rangle$.

Demostración. Supongamos que existe la operación U de la cual se habla en el teorema, entonces, dados cualesquiera $|\psi\rangle, |\phi\rangle \in \mathbb{C}^N$, se cumple

$$\begin{aligned} U|\psi\phi\rangle &= |\psi\psi\rangle \\ U|\varphi\phi\rangle &= |\varphi\varphi\rangle \end{aligned}$$

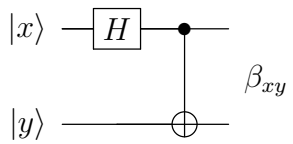
Por lo tanto, $\langle U\psi\phi|U\varphi\phi\rangle = \langle\psi\psi|\varphi\varphi\rangle$. Sin embargo, por un lado

$$\langle U\psi\phi|U\varphi\phi\rangle = \langle\psi\phi|U^\dagger U|\varphi\phi\rangle = \langle\psi\phi|\varphi\phi\rangle = \langle\psi|\varphi\rangle\langle\phi|\phi\rangle = \langle\psi|\varphi\rangle$$

Mientras por el otro $\langle\psi\psi|\varphi\varphi\rangle = \langle\psi|\varphi\rangle\langle\psi|\varphi\rangle = \langle\psi|\varphi\rangle^2$
Pero si $\langle\psi|\phi\rangle = \langle\psi|\phi\rangle^2$, entonces $\langle\psi|\phi\rangle = 0$ o $\langle\psi|\phi\rangle = 1$, lo cual es imposible: 0 implica que los dos vectores tomados al azar son ortogonales, y 1 que son iguales. □

1.5. Estados de Bell

Consideremos el siguiente *circuito* cuántico



Es decir, partiendo del estado inicial $|xy\rangle$, se aplica H al primer qubit. Luego se aplica $CNOT$ a ambos, donde el primero es el de control (marcado con el punto negro). En otras palabras, este circuito representa la siguiente ecuación:

$$\beta_{xy} = CNOT(H \otimes I)|xy\rangle$$

Las posibles salidas de este circuito, cuando x e y varían entre 0 y 1 son las siguientes:

$$|00\rangle \xrightarrow{H(1)} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \xrightarrow{CNOT(1,2)} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \beta_{00}$$
$$|01\rangle \xrightarrow{H(1)} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|1\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle) \xrightarrow{CNOT(1,2)} \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = \beta_{01}$$
$$|10\rangle \xrightarrow{H(1)} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |10\rangle) \xrightarrow{CNOT(1,2)} \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = \beta_{10}$$
$$|11\rangle \xrightarrow{H(1)} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|1\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |11\rangle) \xrightarrow{CNOT(1,2)} \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = \beta_{11}$$

Observación. $\beta_{00} = (X \otimes I)\beta_{01} = (Z \otimes I)\beta_{10} = (XZ \otimes I)\beta_{11}$.

A estos cuatro estados se les llama *Estados de Bell*, en honor a John S. Bell. Estos son estados *entrelazados*, es decir, estados que no pueden representarse como el producto tensorial de dos estados individuales.

A los estados entrelazados también se les llama estados EPR por Einstein, Podolsky, y Rosen [1935] quienes detectaron, en pleno auge de las formulaciones de la teoría cuántica, que existía una acción a distancia que parecía no razonable. Por muchos años se llamó la "paradoja EPR". Lo que determinaron es que cuando se tiene un par entrelazado (físicamente el estado representa por ejemplo el *spin* en un par de electrones, o la polarización de un par de fotones), sucede que cuando se colapsa (por acción de la medición) un estado del par, el segundo también colapsará, incluso cuando físicamente se encuentren a años luz de distancia. Con el tiempo se demostró experimentalmente que esto es exactamente lo que sucede, y por lo tanto no hay paradoja. También se demuestra que esto no contradice la teoría de la relatividad (que entre otras cosas determina que nada puede viajar a mayor velocidad que la luz, ni siquiera la información), ya que no hay trasmisión de información en este colapso a distancia.

Matemáticamente la acción de medir un estado de un par se ve con el siguiente ejemplo:

Ejemplo 1.28 Consideremos el siguiente operador de medición: $M = \{M_0, M_1\}$ donde $M_0 = |0\rangle\langle 0|$ y $M_1 = |1\rangle\langle 1|$.

Aplicando este operador al primer qubit del estado β_{00}, se obtiene uno de los siguientes resultados:

- Si se aplica el proyector M_0 (el cual lo expresamos como $M_0 \otimes I$ para que se aplique M_0 al primer qubit y la identidad al segundo), el estado resultante será

$$\frac{(M_0 \otimes I)\beta_{00}}{\sqrt{p(0)}} = \frac{(|00\rangle\langle 00| + |01\rangle\langle 01|)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)}{\sqrt{\frac{1}{\sqrt{2}}(\langle 00| + \langle 11|)(|00\rangle\langle 00| + |01\rangle\langle 01|)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)}}$$
$$= \frac{\frac{1}{\sqrt{2}}(|00\rangle\langle 00|00\rangle)}{\sqrt{\frac{1}{2}\langle 00|00\rangle\langle 00|00\rangle}} = |00\rangle$$

Análogamente, si se aplica M_1 se obtiene $|11\rangle$

Es decir, al medir el primer qubit del estado entrelazado β_{00}, se obntiene $|00\rangle$ o $|11\rangle$, es decir que ambos qubits colapsan.

1.6. Usando los estados de Bell

Como se mencionó en la sección anterior, el colapso de un par entrelazado no transmite información (y por eso no viola la teoría de la relatividad), sin embargo, es posible utilizar dicho colapso como canal de comunicación, el cual necesita también de un canal clásico para terminar la transmisión (y por ende, el canal clásico implica todas las limitaciones impuestas por la relatividad).
El algoritmo cuántico descripto en la sección 1.6.1, descripto por primera vez por Bennett y Wiesner [1992], permite transmitir dos bit clásicos, enviando sólo un bit cuántico, utilizando un par entrelazado como canal de comunicación. Es llamado "codificación superdensa" ya que se trata de codificar dos bits de información en un bit cuántico, o dicho de otro modo: dos bits de información en el estado de una partícula cuántica.
El algoritmo descripto en la sección 1.6.2, descripto por primera vez por Bennett, Brassard, Crépeau, Jozsa, Peres, y Wootters [1993], permite enviar un bit cuántico enviando dos bit clásicos y utilizando un par entrelazado como canal de comunicación. Es llamado "teleportación cuántica" ya que se trata de mover el valor de un bit cuántico (recordemos que un bit cuántico no puede ser copiado (ver Teorema 1.27)) a otro bit cuántico, o dicho de otro modo: se trata de teletransportar el estado de una partícula a una nueva partícula, destruyendo la primera.

1.6.1. Codificación superdensa

El objetivo de esta técnica es transmitir 2 bits clásicos enviando tan sólo 1 qubit.
Los pasos a seguir por el emisor (a quien llamaremos "Alice") y el receptor (a quien llamaremos "Bob") son los siguientes.

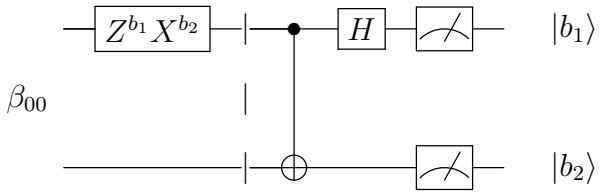
1. Alice y Bob preparan un estado β_{00}.
2. Alice se queda con el primer qubit del par y Bob se lleva el segundo. Podemos considerar que estos dos pasos son la preparación del canal cuántico.

   Observación. El estado entrelazado no se puede separar en el sentido de que no puede considerarse matemáticamente como un qubit multiplicado tensorialmente por otro qubit. Debemos considerarlos como un vector del espacio $\mathbb{C}^2 \otimes \mathbb{C}^2$, es decir, un vector de dimensión 4. Pero físicamente son un par de electrones, o fotones (u otra partícula elemental), las cuales sí pueden ser separadas físicamente (más allá de que no es trivial el problema experimental que representa manipular dichas partículas sin que interaccionen con el ambiente).
3. Alice aplica una transformación a su qubit, de acuerdo a los bits que quiere enviar: $Z^{b_1}X^{b_2}$, donde $C^0 = I$ y $C^1 = C$.
4. Alice envía su qubit a Bob.

5. Bob aplica CNOT a los dos elementos del par y luego Hadamard al primero.

6. Bob realiza una medición.

El circuito completo queda de la siguiente manera



donde la línea punteada determina el paso 4, en el que Alice envía su qubit a Bob.

Ejemplo 1.29 Se quiere enviar los bits 11. Por lo tanto se aplica $(ZX \otimes I)$ a β_{00}, con lo que se obtiene β_{11} (en general, la aplicación de la compuerta $Z^{b_1}X^{b_2}$ cambia el estado β_{00} a $\beta_{b_1 b_2}$):

$$\begin{aligned}(ZX \otimes I)\beta_{00} &= (Z \otimes I)\left((X \otimes I)\beta_{00}\right) \\ &= (Z \otimes I)\left((X \otimes I)\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right) \\ &= (Z \otimes I)\left(\frac{1}{\sqrt{2}}(|10\rangle + |01\rangle)\right) \\ &= \frac{1}{\sqrt{2}}(-|10\rangle + |01\rangle) = \beta_{11}\end{aligned}$$

El resto del circuito (a partir de la línea punteada vertical) es el circuito inverso al de Bell, y como toda compuerta unitaria es tal que $U = U^{-1}$, aplicando el circuito inverso al de Bell se obtiene los estados iniciales. En este caso, $|11\rangle$.

1.6.2. Teleportación cuántica

El objetivo de esta técnica es transmitir un qubit mediante el envío de dos bits clásicos. Los pasos a seguir por Alice y Bob son los siguientes.

1. Alice y Bob preparan un estado β_{00}.

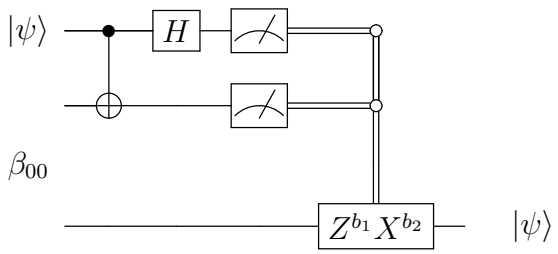
2. Alice se queda con el primer qubit del par y Bob se lleva el segundo.

3. Alice aplica CNOT entre el qubit a transmitir y el primero del par β_{00}, y luego Hadamard al primero.

4. Alice realiza una medición sobre los dos qubits en su posesión y envía el resultado de la medición (2 bits clásicos) a Bob.

5. Bob aplica una transformación sobre su qubit, de acuerdo a los bits recibidos: $Z^{b_1}X^{b_2}$.

El circuito completo queda de la siguiente manera



donde $|\psi\rangle$ es el qubit a transmitir (o "teleportar").

Ejemplo 1.30 Se quiere transmitir el qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, entonces

$$\begin{aligned}
|\psi\rangle \otimes \beta_{00} &= (\alpha|0\rangle + \beta|1\rangle)\left(\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right) \\
&= \frac{1}{\sqrt{2}}\left(\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle)\right) \\
&\xrightarrow{CNOT(1,2)} \frac{1}{\sqrt{2}}\left(\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle)\right) \\
&\xrightarrow{H(1)} \frac{1}{\sqrt{2}}\left(\alpha\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)\right) \\
&= \frac{1}{2}\left[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)\right] \\
&= \frac{1}{2}\sum_{b_1=0}^{1}\sum_{b_2=0}^{1}|b_1 b_2\rangle(X^{b_2}Z^{b_1})|\psi\rangle
\end{aligned}$$

Por lo tanto, aplicando $Z^{b_1}X^{b_2}$, Bob obtendrá el estado original $|\psi\rangle$. (Nótese que para toda compuerta U, $U = U^{-1}$).

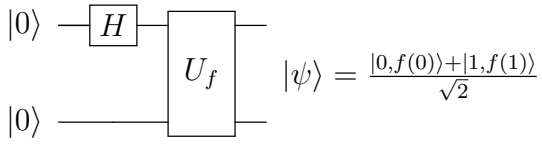
Observación. Si se quiere escribir la compuerta —$\boxed{Z^{b_1}X^{b_2}}$— como dos dos compuertas, debe escribirse —$\boxed{X^{b_2}}$—$\boxed{Z^{b_1}}$—, ya que en $Z^{b_1}X^{b_2}$ primero se aplica la compuerta X^{b_2} y luego Z^{b_1}.

1.7. Paralelismo Cuántico

Consideremos una función $f : \{0,1\} \to \{0,1\}$. Clásicamente para obtener todos los resultados posibles de esta función, es necesario evaluarla tantas veces como sea el cardinal del dominio (2 en este caso, una evaluación para la entrada 0, y otra para la entrada 1). Esta es una función que toma un bit y devuelve un bit. Si fuese un bit cuántico, sería posible evaluar la función en una superposición de 0 y 1 (por ejemplo $\frac{1}{2}(|0\rangle + |1\rangle)$), lo cual nos daría como resultado una superposición de f aplicada a 0 y a 1.
El método es el siguiente. Primero se debe construir una matriz unitaria U_f de $\mathbb{C}^4$ que calcule la función, de la siguiente manera:

$$U_f|x, 0\rangle = |x, f(x)\rangle$$

En realidad, aunque vamos a usar la definición que acabamos de dar, se debe definir también qué sucede cuando el segundo qubit es $|1\rangle$, por lo que esta compuerta se define más generalmente como $U_f|x,y\rangle = |x, y \oplus f(x)\rangle$, donde $\oplus$ es la suma módulo 2.
Lo que se pretende es aplicar f a todas las entradas posibles, por lo que primero se aplicará Hadamard al $|0\rangle$, a fin de obtener una superposición, y luego se aplicará la compuerta U_f. El circuito es el siguiente:



Es decir:

$$|00\rangle \xrightarrow{H(1)} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \xrightarrow{U_f} \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle)$$

La salida de este circuito es un estado que es superposición de todos los resultados posibles de la aplicación de la función f. Y la compuerta U_f fue utilizada una sola vez. El problema ahora pasa porque el resultado es una superposición de todos los resultados posibles, y al querer leerlo (es decir, al medirlo), éste colapsará a uno de los dos. El problema de los algoritmos cuánticos pasa por utilizar la superposición de manera inteligente para aprovechar el paralelismo, pero obteniendo el resultado buscado y no una superposición de resultados sin utilidad.
En el siguiente capítulo mostraremos algunos de los algoritmos que, haciendo uso del paralelismo, consiguen ganancias en complejidad respecto a su contrapartida clásica.

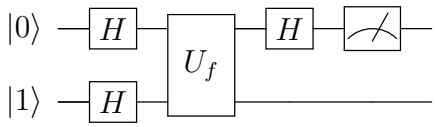
Capítulo 2

Algoritmos cuánticos y aplicación a criptografía

En este capítulo veremos algunos de los algoritmos cuánticos más conocidos. En particular, los algoritmos de Deutsch [1985] y de Deutsch y Jozsa [1992], que pueden considerarse como los primeros algoritmos cuánticos que hacen uso del paralelismo (ver Sección 1.7). El algoritmo de Grover [1996], que es uno de los que motivó que los investigadores en computación se interesaran en el área. No se incluye el algoritmo de Shor [1997], el otro importante algoritmo que motivó a investigadores en computación a adentrarse en el área. Finalmente, el último ejemplo es una aplicación directa de la física cuántica en criptografía, diseñado por Bennett y Brassard [1984], la cual no sigue el esquema de los otros algoritmos cuánticos presentados, pero es también el puntapié de un área de investigación activa dentro de la computación cuántica.

2.1. Algoritmo de Deutsch

El objetivo de este algoritmo es saber si una función que toma un bit y devuelve un bit, es constante o no.
El algoritmo se resume en el siguiente circuito



Observación. U_f es la compuerta definida en la Sección 1.7, la cual consideraremos que existe sin dar más detalles de su construcción.

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle$$

Las primeras dos compuertas Hadamard, aplicadas a $|0\rangle$ y $|1\rangle$, producen lo siguiente:

$$|01\rangle \xrightarrow{H(1,2)} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |x\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}(|x, 0\rangle - |x, 1\rangle) \quad (2.1)$$

donde $|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ es una abreviación introducida por comodidad.

La aplicación de U_f sobre el estado (2.1) produce el siguiente estado:

$$\begin{aligned} & U_f(\frac{1}{\sqrt{2}}(|x,0\rangle - |x,1\rangle)) \\ & = \frac{1}{\sqrt{2}}\left(U_f|x,0\rangle - U_f|x,1\rangle\right) \\ & = \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}(|0,f(0)\rangle + |1,f(1)\rangle) - \frac{1}{\sqrt{2}}(|0,1\oplus f(0)\rangle + |1,1\oplus f(1)\rangle)\right) \\ & = \frac{1}{2}\left(|0,f(0)\rangle + |1,f(1)\rangle - |0,1\oplus f(0)\rangle - |1,1\oplus f(1)\rangle\right) \end{aligned} \tag{2.2}$$

Si $f(0) \neq f(1)$, (2.2) es igual a

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

en cambio si $f(0) = f(1)$,

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

Es decir, el primer qubit es $\pm|-\rangle$, si $f(0) \neq f(1)$ y $\pm|+\rangle$ si $f(0) = f(1)$. Aplicando Hadamard al primer qubit, obtenemos $|1\rangle$ si éste era $|-\rangle$ y $|0\rangle$ si éste era $|+\rangle$.

$$\text{Si } f(0) \neq f(1), \text{ aplicando Hadamard se obtiene } \pm|1\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$
$$\text{Si } f(0) = f(1), \text{ aplicando Hadamard se obtiene } \pm|0\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

es decir, aplicando Hadamard, se obtiene

$$\pm|f(0)\oplus f(1)\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

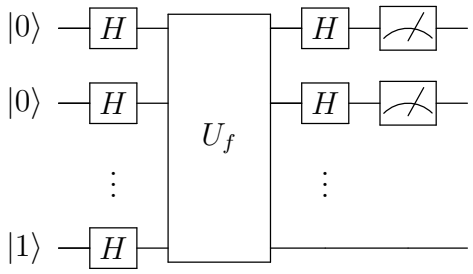
Dado que el primer qubit es $|0\rangle$ o $|1\rangle$, podemos medirlo y nos dará con probabilidad 1 el valor 0 si f es constante y con probabilidad 1 el valor 1 si f no lo es.

Observación. Este algoritmo hace uso del paralelismo, ya que la evaluación de la función se realiza una vez sobre el estado en superposición de 0 y 1. El algoritmo clásico equivalente haría dos evaluaciones de la función y una comparación.

2.2. Algoritmo de Deutsch-Jotza

Este algoritmo es una generalización del anterior. Dada una función que toma n bits y devuelve uno, el algoritmo permite distinguir si la función es constante o balanceada (o sea, con la mitad de las entradas devuelve 0 y con la otra mitad 1). Sólo se distinguen esos dos casos, el algoritmo no es útil para otro tipo de funciones.

El circuito es el siguiente:



La entrada de este algoritmo son $n+1$ qubits: $|0\rangle^{\otimes n}|1\rangle = |0\ldots01\rangle$.
Aplicando las $n+1$ compuertas Hadamard sobre la entrada, se obtiene

$$\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)^{\otimes n}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) = \sum_{\overline{x}\in\{0,1\}^n}\frac{|\overline{x}\rangle}{\sqrt{2^n}}\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \tag{2.3}$$

La compuerta U_f que se utiliza es una generalización del caso anterior definida por

$$U_f|\overline{x},y\rangle = |\overline{x},y\oplus f(\overline{x})\rangle$$

donde $\overline{x}$ son cadenas de n bits.
Es decir

$$U_f|\overline{x},0\rangle = |\overline{x},f(\overline{x})\rangle \qquad\qquad U_f|\overline{x},1\rangle = |\overline{x},1\oplus f(\overline{x})\rangle$$

Por lo tanto, aplicando U_f sobre el estado (2.3) se obtiene

$$\begin{aligned}
U_f\left(\sum_{\overline{x}\in\{0,1\}^n}\frac{|\overline{x}\rangle}{\sqrt{2^n}}\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]\right) &= \sum_{\overline{x}\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}U_f|\overline{x}\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \\
&= \sum_{\overline{x}\in\{0,1\}^n}\frac{1}{\sqrt{2^{n+1}}}\left(U_f|\overline{x},0\rangle - U_f|\overline{x},1\rangle\right) \\
&= \sum_{\overline{x}\in\{0,1\}^n}\frac{1}{\sqrt{2^{n+1}}}\left(|\overline{x},f(\overline{x})\rangle - |\overline{x},1\oplus f(\overline{x})\rangle\right) \\
&= \sum_{\overline{x}\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}|\overline{x}\rangle\left(\frac{|f(\overline{x})\rangle - |1\oplus f(\overline{x})\rangle}{\sqrt{2}}\right)
\end{aligned} \tag{2.4}$$

Para simplificar la notación, la compuerta Hadamard puede expresarse como sigue

$$\left.\begin{array}{l} H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) \\ H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle-|1\rangle) \end{array}\right\} \Rightarrow H|x\rangle = \frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}}(-1)^{xy}|y\rangle$$

De la misma manera, es posible generalizar la aplicación de H a n qubits como sigue:

$$\begin{aligned}
H^{\otimes n}|x_1\ldots x_n\rangle &= \left(\frac{1}{\sqrt{2}}\sum_{z_1\in\{0,1\}}(-1)^{x_1z_1}|z_1\rangle\right)\cdots\left(\frac{1}{\sqrt{2}}\sum_{z_n\in\{0,1\}}(-1)^{x_nz_n}|z_n\rangle\right) \\
&= \frac{1}{\sqrt{2^n}}\sum_{\overline{z}\in\{0,1\}^n}(-1)^{\overline{x}\cdot\overline{z}}|\overline{z}\rangle
\end{aligned}$$

donde $\overline{x}\cdot\overline{z} = x_1z_1+\ldots+x_nz_n$.

Con esta notación, se aplica Hadamard a los primeros n qubits del estado (2.4) (es decir, al ket $|\overline{x}\rangle$), obteniendo

$$\begin{aligned}\sum_{\overline{x}\in\{0,1\}^n} \frac{1}{\sqrt{2^n}} \left(\frac{1}{\sqrt{2^n}} \sum_{\overline{z}\in\{0,1\}^n} (-1)^{\overline{x}\cdot\overline{z}} |\overline{z}\rangle \right) \left(\frac{|f(\overline{x})\rangle - |1 \oplus f(\overline{x})\rangle}{\sqrt{2}} \right) \\ = \sum_{\overline{x}\in\{0,1\}^n} \sum_{\overline{z}\in\{0,1\}^n} \frac{(-1)^{\overline{x}\cdot\overline{z}} |\overline{z}\rangle}{2^n} \left(\frac{|f(\overline{x})\rangle - |1 \oplus f(\overline{x})\rangle}{\sqrt{2}} \right)\end{aligned} \tag{2.5}$$

Casos:

- Si f es constante, el estado (2.5) es

$$\pm \sum_{\overline{x}\in\{0,1\}^n} \sum_{\overline{z}\in\{0,1\}^n} \frac{(-1)^{\overline{x}\cdot\overline{z}} |\overline{z}\rangle}{2^n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

  Cuando $\overline{z} = 0$, los primeros n qubits son

$$\pm \sum_{\overline{x}\in\{0,1\}^n} \frac{|0\rangle^{\otimes n}}{2^n} = \pm \frac{2^n}{2^n} |0\rangle^{\otimes n} = \pm |0\rangle^{\otimes n}$$

  Por lo tanto, dado que este vector tiene norma 1, el resto de los términos de la suma deben anularse, debido a que el resultado tiene que ser forzosamente un vector normalizado. Por lo tanto, cuando f es constante, el estado (2.5) es

$$\pm |0\rangle^{\otimes n} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]$$

  Es decir, midiendo los primeros n qubits se obtiene $0 \ldots 0$ en este caso.

- Si f es balanceada (50 % de las veces devuelve 0 y 50 % devuelve 1), entonces para $\overline{z} = 0$

$$\sum_{\overline{x}\in\{0,1\}^n} \frac{|0\rangle^{\otimes n}}{2^n} \left(\frac{|f(\overline{x})\rangle - |1 \oplus f(\overline{x})\rangle}{\sqrt{2}} \right) = \sum_{\overline{x}\in\{0,1\}^n} (-1)^{\overline{x}} \frac{|0\rangle^{\otimes n}}{2^n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) = 0$$

  Es decir que los primeros n qubits no incluyen al qubit $|0\rangle^{\otimes n}$, y por lo tanto, al medir los primeros n qubits no se puede obtener $0 \ldots 0$ en este caso.

Conclusión: Si se obtiene $|0\rangle^{\otimes n}$ a la salida de la medición, la función es constante, en otro caso la función es balanceada.

2.3. Algoritmo de Búsqueda de Grover

Antes de analizar este algoritmo, son necesarias algunas compuertas extras: la compuerta *Oráculo* (Sección 2.3.1), y la compuerta de *inversión sobre el promedio* (Sección 2.3.2).

2.3.1. Oráculo

Dada una función de un bit en un bit f, la compuerta U_f definida en la Sección 1.7, es $U_f|x,y\rangle = |x, y \oplus f(x)\rangle$.
Si se elije $y = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, entonces

$$\begin{aligned} U_f|x,y\rangle &= U_f\left(|x\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right) \\ &= \frac{1}{\sqrt{2}}\left(U_f|x,0\rangle - U_f|x,1\rangle\right) \\ &= \frac{1}{\sqrt{2}}(|x, f(x)\rangle - |x, 1 \oplus f(x)\rangle) \\ &= |x\rangle\frac{1}{\sqrt{2}}(|f(x)\rangle - |1 \oplus f(x)\rangle) \\ &= (-1)^{f(x)}|x,y\rangle \end{aligned}$$

Dado que U_f no modifica el estado y, es posible omitirlo y tomarlo como parte de la definición de la compuerta. Entonces, definimos la compuerta

$$U|x\rangle = (-1)^{f(x)}|x\rangle$$

a la cual se le llama *Oráculo*.

2.3.2. Inversión sobre el promedio

Sea el estado $|\phi\rangle = {}^1/\sqrt{2^n}\sum_{\overline{x}\in\{0,1\}^n}|\overline{x}\rangle$. Definimos la compuerta de *Inversión sobre el promedio* como $G = 2|\phi\rangle\langle\phi| - I$. Es decir

$$\begin{aligned} G &= 2|\phi\rangle\langle\phi| - I \\ &= 2\begin{pmatrix} \frac{1}{\sqrt{2^n}} \\ \vdots \\ \frac{1}{\sqrt{2^n}} \end{pmatrix}_{2^n}\left(\frac{1}{\sqrt{2^n}}\cdots\frac{1}{\sqrt{2^n}}\right)_{2^n} - I \\ &= \begin{pmatrix} \frac{2}{2^n}-1 & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \frac{2}{2^n} & \frac{2}{2^n}-1 & \cdots & \frac{2}{2^n} \\ \vdots & \vdots & & \vdots \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n}-1 \end{pmatrix}_{2^n\times 2^n} \end{aligned}$$

La aplicación de G sobre un estado cualquiera $|\psi\rangle = \sum_{\overline{x}\in\{0,1\}^n} a_{\overline{x}}|\overline{x}\rangle$ es la siguiente

$$\begin{array}{c|c} G|\psi\rangle & \begin{pmatrix} a_0 \\ \vdots \\ a_{2^n-1} \end{pmatrix} \\ \hline \begin{pmatrix} \frac{2}{2^n}-1 & \cdots & \frac{2}{2^n} \\ \vdots & & \vdots \\ \frac{2}{2^n} & \cdots & \frac{2}{2^n}-1 \end{pmatrix} & \begin{pmatrix} \left(\sum_{\overline{x}\in\{0,1\}^n} \frac{2a_{\overline{x}}}{2^n}\right) - a_0 \\ \vdots \\ \left(\sum_{\overline{x}\in\{0,1\}^n} \frac{2a_{\overline{x}}}{2^n}\right) - a_{2^n-1} \end{pmatrix} \end{array}$$

Es decir:

$$G\left(\sum_{\overline{x}\in\{0,1\}^n} a_{\overline{x}}|\overline{x}\rangle\right) = \sum_{\overline{x}\in\{0,1\}^n}\left[\left(\sum_{\overline{y}\in\{0,1\}^n}\frac{2a_{\overline{y}}}{2^n}\right) - a_{\overline{x}}\right]|\overline{x}\rangle = \sum_{\overline{x}\in\{0,1\}^n}(2A - a_{\overline{x}})|\overline{x}\rangle$$

donde A es el promedio de los $a_{\overline{x}}$.

2.3.3. El algoritmo

El algoritmo de Grover es un algoritmo de búsqueda sobre una lista desordenada. Suponemos una lista de tamaño N, con $N = 2^n$ (observar que siempre es posible aumentar la lista con datos irrelevantes para cumplir la condición sobre N). Los índices de la lista son $\overline{x} \in 0, 1^n$, es decir $\overline{x} = 0 \ldots 2^n - 1$.
El objetivo del algoritmo es localizar el $\overline{x}_0$ tal que $f(\overline{x}_0) = 1$, para una función booleana f dada.
El input del circuito es $|0\rangle^{\otimes n}$.

2.3.3.1. Paso 1: Se aplica Hadamard ($H^{\otimes n}$)

El primer paso es generar una superposición en todos los qubits.

$$|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}}\sum_{\overline{x}\in\{0,1\}^n}|\overline{x}\rangle \tag{2.6}$$

Este estado es una superposición de todos los elementos de la lista. La idea del algoritmo es subir la probabilidad de que al medir este estado obtengamos el elemento $\overline{x}_0$.

2.3.3.2. Paso 2: Se aplica el oráculo (U)

Aplicar el oráculo es el equivalente a aplicar la función booleana f sobre la superposición.

$$(2.6) \xrightarrow{U} \frac{1}{\sqrt{2^n}}\sum_{\overline{x}\in\{0,1\}^n}(-1)^{f(\overline{x})}|\overline{x}\rangle \tag{2.7}$$

2.3.3.3. Paso 3: Se aplica la inversión sobre el promedio (G)

$$
\begin{aligned}
(2.7) &= \sum_{\overline{x}\in\{0,1\}^n} \underbrace{\left[\frac{(-1)^{f(\overline{x})}}{\sqrt{2^n}}\right]}_{a_{\overline{x}}} |\overline{x}\rangle \\
&\xrightarrow{G} \sum_{\overline{x}\in\{0,1\}^n} (2A - a_{\overline{x}})|\overline{x}\rangle \\
&= \sum_{\overline{x}\in\{0,1\}^n} \left[\left(2\sum_{\overline{y}\in\{0,1\}^n} \frac{(-1)^{f(\overline{y})}}{2^n\sqrt{2^n}}\right) - \frac{(-1)^{f(\overline{x})}}{\sqrt{2^n}}\right] |\overline{x}\rangle \\
&= \sum_{\overline{x}\in\{0,1\}^n} \left[\left(2\sum_{\substack{\overline{y}\in\{0,1\}^n \\ \overline{y}\neq\overline{x}}} \frac{(-1)^{f(\overline{y})}}{2^n\sqrt{2^n}}\right) + \frac{2(-1)^{f(\overline{x})}}{2^n\sqrt{2^n}} - \frac{(-1)^{f(\overline{x})}}{\sqrt{2^n}}\right] |\overline{x}\rangle \\
&= \sum_{\overline{x}\in\{0,1\}^n} \left[\left(2\sum_{\substack{\overline{y}\in\{0,1\}^n \\ \overline{y}\neq\overline{x}}} \frac{(-1)^{f(\overline{y})}}{2^n\sqrt{2^n}}\right) + \frac{2-2^n}{2^n\sqrt{2^n}}(-1)^{f(\overline{x})}\right] |\overline{x}\rangle
\end{aligned}
\tag{2.8}
$$

En el estado (2.8), el término $\overline{x} = \overline{x}_0$, con $f(\overline{x}_0) = 1$, el cual estamos buscando es el siguiente:

$$
\begin{aligned}
\left[\left(2\sum_{\substack{\overline{y}\in\{0,1\}^n \\ \overline{y}\neq\overline{x}_0}} \frac{1}{2^n\sqrt{2^n}}\right) + \frac{2^n-2}{2^n\sqrt{2^n}}\right] |\overline{x}_0\rangle &= \left[\frac{2}{2^n\sqrt{2^n}}(2^n-1) + \frac{2^n-2}{2^n\sqrt{2^n}}\right] |\overline{x}_0\rangle \\
&= \left[\frac{2^{n+1}+2^n-4}{2^n\sqrt{2^n}}\right] |\overline{x}_0\rangle
\end{aligned}
$$

mientras que los otros términos, donde $\overline{x} \neq \overline{x}_0$, son

$$
\left[\left(2\sum_{\substack{\overline{y}\in\{0,1\}^n \\ \overline{y}\neq\overline{x}_0 \\ \overline{y}\neq\overline{x}}} \frac{1}{2^n\sqrt{2^n}}\right) + \frac{2(-1)}{2^n\sqrt{2^n}} + \frac{2-2^n}{2^n\sqrt{2^n}}\right] |\overline{x}\rangle = \left[\frac{2^{n+1}-2^n-4}{2^n\sqrt{2^n}}\right] |\overline{x}\rangle
$$

El algoritmo ha cambiado las amplitudes del estado, aumentando la amplitud del estado $\overline{x}_0$ y disminuyendo las otras.
Repitiendo este proceso (pasos 2 y 3) se va subiendo la amplitud del estado que se quiere encontrar y disminuyendo las otras. Sin embargo es cíclico: pasado cierto número de repeticiones, esa amplitud vuelve a decrecer. En la Sección 2.3.4 se calcula el número optimo de repeticiones para obtener la amplitud máxima. Cuando la amplitud es máxima, se realiza una medición, obteniendo el estado $\overline{x}_0$ con la máxima probabilidad. En la Sección 2.3.4 se muestra que la probabilidad de error tiene cota máxima en $1/2^n$.

Ejemplo

Sea una lista de $2^4 = 16$ elementos, de los que sólo uno, $\overline{x}_0$, verifica la propiedad $f(\overline{x}_0) = 1$. El algoritmo comienza por tomar el estado $|0\rangle^{\otimes 4}$ y aplicar $H^{\otimes 4}$ obteniendo,

$$\frac{1}{4}\sum_{\overline{x}\in\{0,1\}^4} |\overline{x}\rangle$$

Inicialmente todas las amplitudes son iguales a $1/4$. Se aplica el oráculo y se obtiene

$$\frac{1}{4}\sum_{\overline{x}\in\{0,1\}^4} (-1)^{f(\overline{x})}|\overline{x}\rangle$$

Luego se aplica la inversión sobre el promedio, y la nueva amplitud del estado $\overline{x}_0$ será

$$\frac{2^5 + 2^4 - 4}{2^4\sqrt{2^4}} = \frac{11}{16} = 0.6875$$

y para el resto de los $\overline{x}$ la amplitud será

$$\frac{2^5 - 2^4 - 4}{2^4\sqrt{2^4}} = \frac{3}{16} = 0.1875$$

Con las sucesivas repeticiones de la aplicación del oráculo y la inversión sobre el promedio, se obtienen las siguientes amplitudes:

Repetición	Amplitud de $\overline{x}_0$	Amplitud de $\overline{x} \neq \overline{x}_0$	Probabilidad de error
1	0.6875	0.1875	0.527
2	0.953125	0.078125	0.092
3	0.98046875	-0.05078125	0.039

A partir de la iteración 4 la probabilidad de error comienza a subir, por lo tanto el número óptimo de iteraciones es 3, con una probabilidad de error de 0.039.

2.3.4. Cálculo del número óptimo de iteraciones

Luego de k iteraciones $\overline{x}_0$ tendrá una amplitud b_k y el resto tendrán todos una amplitud m_k. Es decir, el estado será

$$b_k|\overline{x}_0\rangle + m_k \sum_{\substack{\overline{x}\in\{0,1\}^n \\ \overline{x}\neq\overline{x}_0}} |\overline{x}\rangle$$

En cada iteración se aplica el oráculo U, el cual cambia el signo de b_k, y luego G. Es posible definir recursivamente las amplitudes en la repetición k:

$$\begin{aligned} m_0 = b_0 &= \frac{1}{\sqrt{2^n}} \\ m_{k+1} &= 2A_k - m_k \\ b_{k+1} &= 2A_k + b_k \end{aligned} \qquad \text{donde} \qquad A_k = \frac{(2^n - 1)m_k - b_k}{2^n}$$

Las fórmulas cerradas para estas recursiones son

$$m_k = \frac{1}{\sqrt{2^n - 1}} \cos((2k+1)\gamma)$$
$$b_k = \text{sen}((2k+1)\gamma)$$

donde $\cos(\gamma) = \sqrt{\frac{2^n - 1}{2^n}}$ y $\text{sen}(\gamma) = \sqrt{\frac{1}{2^n}}$.
Para conseguir la mínima probabilidad de error, se debe minimizar $|m_k|$. Notar que $m_k = 0$ si y sólo si $(2k+1)\gamma = \frac{\pi}{2}$, es decir, si $k = \frac{\pi}{4\gamma} - \frac{1}{2}$.
Sin embargo, dado que k es el número de repeticiones, debe ser entero, por lo tanto, el número óptimo de iteraciones es

$$\tilde{k} = \left\lfloor \frac{\pi}{4\gamma} \right\rfloor$$

Para calcular una cota de la probabilidad de error, observar primero que que $|k - \tilde{k}| \leq \frac{1}{2}$, entonces

$$|\frac{\pi}{2} - (2\tilde{k}+1)\gamma| = |(2k+1)\gamma - (2\tilde{k}+1)\gamma| = |2\gamma(k - \tilde{k})| \leq \gamma$$

Con esto, podemos determinar que la probabilidad de error luego de $\tilde{k}$ iteraciones es

$$(2^n - 1)(m_k)^2 = cos^2((2\tilde{k}+1)\gamma) = sen^2(\frac{\pi}{2} - (2\tilde{k}+1)\gamma) \leq sen^2(\gamma) = \frac{1}{2^n}$$

En el ejemplo anterior

$$\tilde{k} = \left\lfloor \frac{\pi}{4asen(\sqrt{\frac{1}{16}})} \right\rfloor = 3$$

y la probabilidad de error es $0.039 \leq \frac{1}{2^4} = 0.0625$.

2.4. Aplicación criptográfica

2.4.1. One-time pad

Este es un método de criptografía clásica [Vernam, 1926] que consiste en compartir una secuencia de bits (clave) del largo del mensaje a transmitir y aplicar la operación reversible XOR para cifrar y descifrar. (Ver Figura 2.1). Las claves deben ser secretas y no deben ser reutilizadas.
Este método es 100 % seguro: un 0 en el mensaje encriptado puede significar un 0 en el mensaje original y un 0 en la clave, o un 1 en el mensaje y un 1 en la clave. Lo mismo sucede con un 1 en el mensaje encriptado. Es decir que adivinar la clave tiene la misma probabilidad que adivinar el mensaje original. La única debilidad de este método es la predistribución de claves, ya que el canal que se use para distribuirla podría ser vulnerado.
El método cuántico que se describe a continuación, QKD-BB84 (por *Quantum Key Distribution* de Bennett y Brassard [1984]), es justamente un método para la distribución segura de claves.

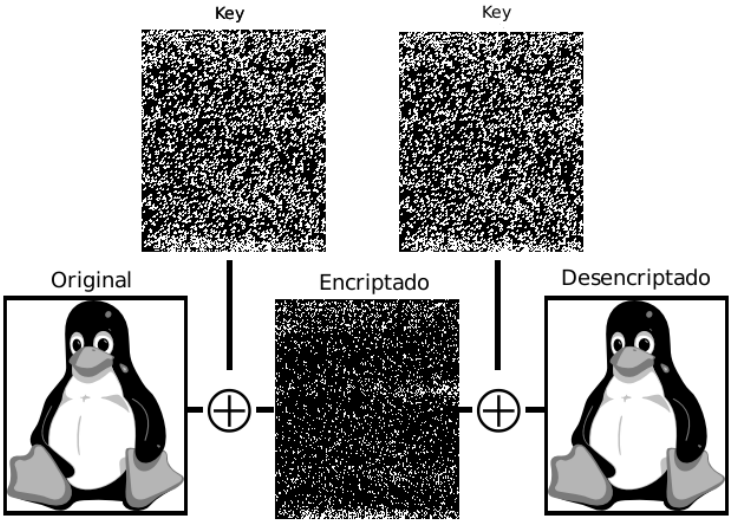


Figura 2.1: One-Time pad

2.4.2. Criptosistema Cuántico QKD-BB84

La idea es transmitir una clave binaria por un canal inseguro.
Para transmitir el bit 0, Alice (el emisor) puede elegir, al azar, la base $\{|0\rangle, |1\rangle\}$ (a la que llamaremos esquema $+$) y considerar $0 \equiv |0\rangle$, o la base $\{|-\rangle, |+\rangle\}$ (a la que llamaremos esquema $\times$) y considerar $0 \equiv |-\rangle$. Análogamente al bit 1 lo codificamos como $|1\rangle$ en el esquema $+$ o como $|+\rangle$ en el esquema $\times$.
Bob realizará una medición sobre el estado recibido eligiendo al azar entre el esquema $+$ y el esquema $\times$. Ver ejemplo en Figura 2.2. El paso final es intercambiar información (por un canal abierto) de los esquemas utilizados, y sólo conservar los bits producidos usando el mismo esquema.

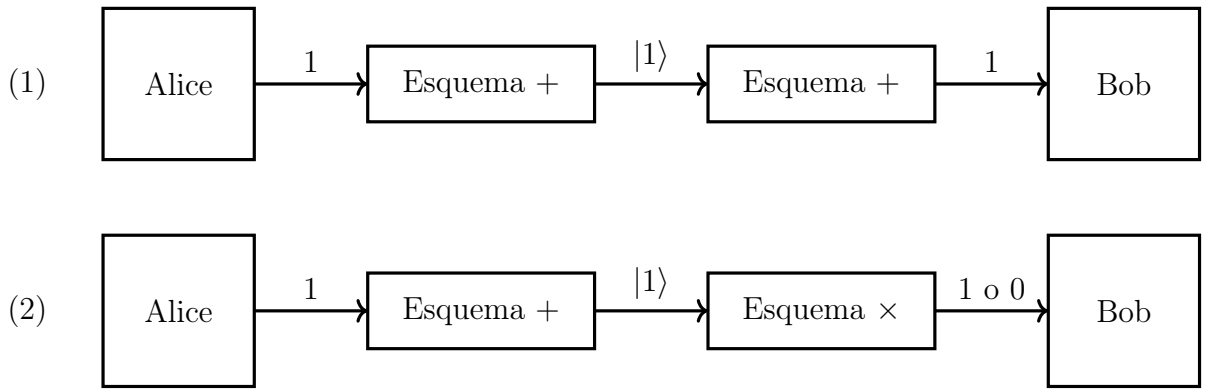


Figura 2.2: Ejemplo: (1) Alice transmite un 1 codificado mediante el esquema $+$ y Bob elije al azar el esquema $+$ obteniendo un 1 (2) si Bob elige el esquema $\times$ obtiene 0 ó 1 con probabilidad $1/2$.

El algoritmo paso a paso:

1. Alice comienza a transmitir una secuencia de 0 y 1, elegidos aleatoriamente, alternando los esquemas $+$ y $\times$ también aleatoriamente.

2. Bob recibe la secuencia y va alternando las mediciones entre los esquemas + y × aleatoriamente.

3. Alice le transmite a Bob la sucesión de esquemas empleada.

4. Bob le informa a Alice en qué casos utilizó el mismo esquema.

5. Usando solamente los bits de los esquemas idénticos a dos puntas, ambos han definido una sucesión aleatoria de bits que servirá como one-time pad de encriptación para transmisiones futuras por cualquier canal.

Esquemas de Alice	×	+	+	×	×	+	×	+
Valores de Alice	$\|-\rangle$	$\|0\rangle$	$\|0\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$	$\|-\rangle$	$\|1\rangle$
Esquemas de Bob	+	×	+	×	+	+	×	×
Valores de Bob	$\|0\rangle$	$\|+\rangle$	$\|0\rangle$	$\|+\rangle$	$\|1\rangle$	$\|0\rangle$	$\|-\rangle$	$\|-\rangle$
Coincidencias			√	√		√	√	
Clave			0	1		0	0	

6. Alice y Bob intercambian hashes de las claves (en bloques) para aceptarla o descartarla.

Inviolabilidad Este protocolo es, en teoría, inviolable. Supongamos que Cliff espía el canal de comunicación entre Alice y Bob e intenta recuperar la clave. Cliff está en la misma situación que Bob y no conoce cuál esquema es el correcto, + o ×. Por lo tanto elige al azar y se equivocará, en promedio, la mitad de las veces.
En el paso 5 Alice y Bob se ponen de acuerdo en cuáles valores tomar en cuenta (las coincidencias de la secuencia de esquemas). Esta información no le es útil a Cliff porque sólo en la mitad de las veces habrá usado el detector correcto, de manera que mal interpretará sus valores finales.
Además el QKD brinda el método para que Alice y Bob puedan detectar el potencial espionaje de Cliff:
Imaginemos que Alice envió un 0 con el esquema × (es decir, el qubit $|-\rangle$). Si Cliff usa el esquema +, colapsará el qubit a $|0\rangle$ o $|1\rangle$. Si Bob usa el esquema × y mide $|-\rangle$ coincide con lo enviado por Alice, pero si mide $|+\rangle$ Alice y Bob descubrirán esa discrepancia durante el intercambio de hashes, por lo tanto descartarán el bloque.

Capítulo 3

Introducción a la mecánica cuántica

3.1. Postulados de la mecánica cuántica

En el Capítulo 1 vimos los postulados de la mecánica cuántica sin mencionarlo, desde un punto de vista matemático formal. Revisitaremos los mismos postulados, nombrándolos como tales. Estos cuatro postulados definen el entorno matemático conocido como mecánica cuántica.

> **Postulado 1** (Espacio de estados)**.** Todo sistema físico cuántico aislado tiene asociado un espacio vectorial complejo con producto escalar conocido como el *espacio de estados* del sistema. El sistema se describe completamente por un *vector de estado*, el cual es un vector unidad en el espacio de estados.

> **Postulado 2** (Evolución)**.** La evolución de un sistema físico cuántico aislado se describe por una *transformación unitaria*. Es decir, el estado $|\psi\rangle$ del sistema en el tiempo t_1 se relaciona con el estado $|\psi'\rangle$ del sistema en el tiempo t_2 a través del operador unitario U, el cual sólo depende de los tiempos t_1 y t_2.
>
> $$|\psi'\rangle = U|\psi\rangle$$

El postulado anterior se puede tomar con tiempo continuo, para lo cual hace falta una ecuación diferencial, y el postulado se transforma en el siguiente:

> **Postulado 2′.** La evolución del estado de un sistema físico cuántico aislado es descripta por la *ecuación de Shcrödinger*,
>
> $$i\hbar\frac{d|\psi\rangle}{dt} = H|\psi\rangle$$
>
> En esta ecuación, $\hbar$ es una constante física conocida como *constante de Planck* cuyo valor debe ser determinado experimentalmente. El valor exacto no es importante, en la práctica es común absorber el valor $\hbar$ en H tomando $\hbar = 1$. El operador H no es la compuerta Hadamard vista anteriormente sino un operador hermítico fijo conocido como el *Hamiltoniano* del sistema.

Postulado 3 (Medición cuántica)**.** La medición cuántica se describe por una colección $\{M_m\}$ de *matrices de medición*. Dichas matrices actúan en el espacio de estados del sistema que se mide. El índice m refiere a los resultados posibles de la medición. Si el estado del sistema es $|\psi\rangle$, inmediatamente antes de la medición, entonces la probabilidad de que el resultado m ocurra viene dado por

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

y el estado del sistema luego de la medición es

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

Las matrices satisfacen la ecuación de completitud,

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

La ecuación de completitud expresa el hecho de que las probabilidades suman uno:

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

Postulado 4 (Sistema compuesto)**.** El espacio de estados de un sistema físico compuesto es el producto tensorial de los espacios de estados de los componentes. Más aún, si tenemos sistemas enumerados de 1 a n, donde el sistema i está en el estado $|\psi_i\rangle$, el estado conjunto del sistema total es

$$|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$$

3.1.1. Medición proyectiva

Un caso particular del postulado 3 es el conocido como *medición proyectiva*. De hecho, la medición general es equivalente a las mediciones proyectivas más operaciones unitarias. Por lo tanto, en general, usaremos sólo mediciones proyectivas.

3.1.1.1. Preliminares

Definición 3.1 (Autovectores y autovalores) Un *autovector* de un operador lineal A en un espacio vectorial dado es un vector no-nulo $|v\rangle$ tal que $A|v\rangle = v|v\rangle$, donde v es un número complejo conocido como *autovalor* de A correspondiente a $|v\rangle$.

Observación. En la definición anterior, notar que $v \neq |v\rangle$. De hecho, $v \in \mathbb{C}$ y $|v\rangle \in \mathbb{C}^2$. La "$v$" que aparece en $|v\rangle$ es simplemente una etiqueta, un nombre, para el vector.

Ejemplo 3.2 Consideremos la matriz de Pauli iXZ

$$\begin{aligned} iXZ &= i(|0\rangle\langle 1| + |1\rangle\langle 0|)(|0\rangle\langle 0| - |1\rangle\langle 1|) \\ &= i(|0\rangle\langle 1|0\rangle\langle 0| - |0\rangle\langle 1|1\rangle\langle 1| + |1\rangle\langle 0|0\rangle\langle 0| - |1\rangle\langle 0|1\rangle\langle 1|) \\ &= i(|1\rangle\langle 0| - |0\rangle\langle 1|) \end{aligned}$$

O, en su notación matricial, $iXZ = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$

Queremos buscar un vector $|v\rangle = \alpha|0\rangle + \beta|1\rangle$ tal que $iXZ|v\rangle = v|v\rangle$, para algún v, y con $\||v\rangle\| = 1$. Es decir:

$$i(|1\rangle\langle 0| - |0\rangle\langle 1|)(\alpha|0\rangle + \beta|1\rangle) = i(\alpha|1\rangle - \beta|0\rangle) = -\beta i|0\rangle + \alpha i|1\rangle$$

debe ser igual a $v(\alpha|0\rangle + \beta|1\rangle)$, con $|\alpha|^2 + |\beta|^2 = 1$.
Por lo tanto,

$$\begin{cases} |\alpha|^2 + |\beta|^2 = 1 \\ v\alpha = -\beta i \\ v\beta = \alpha i \end{cases} \Rightarrow \begin{cases} \left|\frac{\beta i}{v}\right|^2 + |\beta|^2 = 1 \\ \alpha = -\frac{\beta i}{v} \\ v\beta = \left(-\frac{\beta i}{v}\right) i \end{cases} \Rightarrow \begin{cases} |\beta| = \sqrt{\frac{|v|}{|v|+1}} \\ \alpha = -\beta i \\ v^2 = 1 \end{cases}$$

$$\Rightarrow \begin{cases} |\beta| = \frac{1}{\sqrt{2}} \\ \alpha = -\beta i \\ v = 1 \end{cases} \text{ o } \begin{cases} |\beta| = \frac{1}{\sqrt{2}} \\ \alpha = -\beta i \\ v = -1 \end{cases}$$

Tomando, por ejemplo, $v = 1$ y $\beta = 1/\sqrt{2}$ tenemos $\alpha = -i/\sqrt{2}$, y por lo tanto 1 es un autovalor de iXZ con autovector $1/\sqrt{2}(|1\rangle - i|0\rangle)$.

Definición 3.3 (Función característica) La *función característica* de un operador lineal A es $c(x) = \det|A - xI|$.

Teorema 3.4 Las soluciones a la ecuación $c(x) = 0$ son los autovalores del operador A. □

Ejemplo 3.5 En el ejemplo anterior, podemos ver que

$$\begin{aligned} c(x) &= \det|iXZ - xI| \\ &= \det|i(|1\rangle\langle 0| - |0\rangle\langle 1|) - x(|0\rangle\langle 0| + |1\rangle\langle 1|)| \\ &= \det|-x|0\rangle\langle 0| - i|0\rangle\langle 1| + i|1\rangle\langle 0| - x|1\rangle\langle 1|| \\ &= (-x)^2 - (-i^2) \\ &= x^2 - 1 \end{aligned}$$

Y tenemos $c(x) = 0 \Rightarrow x^2 - 1 = 0 \Rightarrow x = \pm 1$.

Definición 3.6 (Autoespacio) El *autoespacio* correspondiente un autovalor v de un operador linear A es el conjunto de vectores que tienen a v como autovalor.

Ejemplo 3.7 Siguiendo con el ejemplo anterior, el autoespacio correspondiente al autovalor 1 del operador iXZ es $\{\beta|1\rangle - \beta i|0\rangle \mid \beta \in \mathbb{C}\}$.

Teorema 3.8 El autoespacio de un autovalor v de un operador lineal A en un espacio vectorial V es un subespacio vectorial de V. □

3.1.1.2. Medición proyectiva

Definición 3.9 Una medición proyectiva es descripta por un *observable*, M, el cual es un operador hermítico en el espacio de estados del sistema que es objeto de la observación. El observable tiene descomposición espectral (es decir, factorización a forma canónica) dada por:

$$M = \sum_m m P_m$$

donde P_m es el proyector al autoespacio de M con autovalor m. Los posibles resultados de la medición corresponden con los autovalores m del observable. Luego de medir $|\psi\rangle$, la probabilidad de obtener el resultado m viene dada por

$$p(m) = \langle\psi|P_m|\psi\rangle$$

Al obtener el resultado m, el estado del sistema inmediatamente luego de la medición es

$$\frac{P_m|\psi\rangle}{\sqrt{p(m)}}$$

Observación. La medición proyectiva se puede ver como un caso particular del Postulado 3. Si a las matrices que forman el operador medición del Postulado 3 le agregamos la condición que los M_m son hermíticos y ortogonales, es decir $M_m M_{m'} = \delta_{m,m'} M_m$, entonces el Postulado 3 reduce a las mediciones proyectivas que acabamos de definir.

Ejemplo 3.10 Consideremos la medición del observable Z.

$$c(x) = \det|Z - xI| = \det|(1-x)|0\rangle\langle 0| - (1+x)|1\rangle\langle 1|| = (1-x)(-1-x) = -1 + x^2$$

Por lo tanto, las soluciones a $c(x) = 0$ son 1 y -1, y esos son los autovalores de Z. Dichos autovalores corresponden a los autovectores $|0\rangle$ y $|1\rangle$ respectivamente.
Los proyectores P_0 y P_1 sobre los autoespacios $\{\alpha|0\rangle \mid \alpha \in \mathbb{C}\}$ y $\{\beta|1\rangle \mid \beta \in \mathbb{C}\}$ respectivamente son $P_0 = |0\rangle\langle 0|$ y $P_1 = |1\rangle\langle 1|$. Notar que

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1|$$

Entonces, la medición de Z sobre el estado $|-\rangle = {}^{(|0\rangle+|1\rangle)}/\sqrt{2}$ da como resultado 1 con probabilidad $\langle -|0\rangle\langle 0|-\rangle = {}^1/_2$. Similarmente, se obtiene resultado -1 con probabilidad ${}^1/_2$.

3.1.2. Fase

Consideremos por ejemplo el estado $e^{i\theta}|\psi\rangle$[①], donde $|\psi\rangle$ es un vector de estado, y θ es un número real. Decimos que el estado $e^{i\theta}|\psi\rangle$ es igual a $|\psi\rangle$, excepto por la fase global $e^{i\theta}$. La medición sobre ambos estados es la misma: Supongamos que M_m es una matriz de un operador de medición. Entonces las probabilidades de aplicar esa matriz vienen dadas por $\langle\psi|M_m^\dagger M_m|\psi\rangle$ y por $\langle\psi|e^{-i\theta}M_m^\dagger M_m e^{i\theta}|\psi\rangle = \langle\psi|M_m^\dagger M_m|\psi\rangle$[②]. Por lo tanto, desde un punto de vista observacional, ambos estados son idénticos.
Por esta razón solemos ignorar las fases globales ya que son irrelevantes a las propiedades observacionales de sistemas físicos.

[①] $e^{i\theta} = \cos\theta + i\sin\theta$ ($ae^{i\theta}$ es la llamada *notación exponencial* de un número complejo, donde a su módulo y θ su argumento).
[②] $e^{-i\theta}e^{i\theta} = e^{-i\theta+i\theta} = e^0 = 1$.

3.2. Operador densidad

Hasta ahora hemos visto la mecánica cuántica en términos de vectores de estados. Una formulación alternativa es usando el operador densidad (o matriz densidad). Esta presentación es equivalente matemáticamente, pero provee un lenguaje más conveniente para razonar en algunos escenarios comunes que se encuentran en la mecánica cuántica.

3.2.1. Preliminares

Definición 3.11 (Traza) La *traza* de una matriz es la suma de sus elementos diagonales. Así, si $A = \sum_i \sum_j \alpha_{ij} |u_i\rangle\langle u_j|$, la traza se define por

$$\mathsf{tr}(A) = \sum_i \alpha_{ii}$$

Teorema 3.12 Sea $A = |\psi\rangle\langle\varphi|$. Entonces, $\mathsf{tr}(A) = \langle\varphi|\psi\rangle$.

Demostración. Sean $|\psi\rangle = \sum_i a_i|u_i\rangle$ y $|\varphi\rangle = \sum_j b_j|u_j\rangle$. Entonces,

$$\mathsf{tr}(A) = \mathsf{tr}(|\psi\rangle\langle\varphi|) = \mathsf{tr}(\sum_i a_i|u_i\rangle \sum_j b_j^*\langle u_j|) \ = \mathsf{tr}(\sum_{ij} a_i b_j^* |u_i\rangle\langle u_j|) = \sum_i a_i b_i^* = \langle\varphi|\psi\rangle$$

□

Ejemplo 3.13 Sea $A = |0\rangle\langle -|$. Entonces, $A = 1/\sqrt{2}(|0\rangle\langle 0| - |0\rangle\langle 1|)$ y $\mathsf{tr}(A) = 1/\sqrt{2}$.
Por otro lado, siguiendo el teorema, $\mathsf{tr}(A) = \langle 0|-\rangle = 1/\sqrt{2}(\langle 0|0\rangle - \langle 0|1\rangle) = 1/\sqrt{2}$.

Ejemplo 3.14 Sea $A = |+\rangle\langle -| = 1/2(|0\rangle\langle 0| - |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|)$.
Entonces, $\mathsf{tr}(A) = 1/2 - 1/2 = 0 = \langle +|-\rangle$.

El siguiente corollario es muy útil para evaluar la traza de un operador.

Corolario 3.15 Sea $|\psi\rangle$ un vector normalizado y A un operador cuántico. Entonces

$$\mathsf{tr}(A|\psi\rangle\langle\psi|) = \langle\psi|A|\psi\rangle$$

Ejercicio 3.16. Probar el Corollario 3.15.

Ejemplo 3.17 $\mathsf{tr}(X|0\rangle\langle 0|) = \langle 0|X|0\rangle = \langle 0|0\rangle\langle 1|0\rangle + \langle 0|1\rangle\langle 0|0\rangle = 0$.

Propiedades (de la traza de una matriz) Sean A y B matrices de la misma dimensión, U un operador unitario y $\lambda \in \mathbb{C}$. Entonces

1. $\mathsf{tr}(AB) = \mathsf{tr}(BA)$
2. $\mathsf{tr}(A + B) = \mathsf{tr}(A) + \mathsf{tr}(B)$
3. $\mathsf{tr}(\lambda A) = \lambda \mathsf{tr}(A)$
4. $\mathsf{tr}(UAU^\dagger) = \mathsf{tr}(A)$

Demostración. Sólo mostramos la propiedad 4, las otras se dejan como ejercicio. De la propiedad 1 se tiene $\mathsf{tr}(UAU^\dagger) = \mathsf{tr}(U^\dagger UA)$, y como U es unitaria, $\mathsf{tr}(U^\dagger UA) = \mathsf{tr}(A)$. □

3.2.2. Conjuntos de estados cuánticos

El operador densidad provee una manera conveniente de describir un sistema cuántico en el cual el estado no se conoce del todo.

Definición 3.18 (Operador o matriz densidad) Supongamos que un sistema cuántico está en uno de un número de estados $|\psi_i\rangle$, donde la probabilidad de que el estado sea $|\psi_i\rangle$ viene dada por p_i.
Decimos que el conjunto $\{p_i, |\psi_i\rangle\}$ es el *conjunto de estados puros*. El *operador densidad* o *matriz densidad* para este estado viene dado por la ecuación

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

Ejemplo 3.19 El operador densidad del conjunto de estados puros $\{(1/4, |+\rangle); (3/4, |1\rangle)\}$ tiene operador densidad

$$\rho = 1/4|+\rangle\langle+| + 3/4|1\rangle\langle 1| = 1/8|0\rangle\langle 0| + 1/8|0\rangle\langle 1| + 1/8|1\rangle\langle 0| + 7/8|1\rangle\langle 1|$$

Es decir

$$\rho = \begin{pmatrix} 1/8 & 1/8 \\ 1/8 & 7/8 \end{pmatrix}$$

Observación. Todos los postulados de la mecánica cuántica se pueden reformular en términos del operador densidad, y haremos eso más adelante en esta sección.

Evolución Supongamos que la evolución de un sistema cuántico cerrado se describe por el operador unitario U. Si el sistema estaba inicialmente en el estado $|\psi_i\rangle$ con probabilidad p_i, entonces, luego de la evolución el sistema estará en estado $U|\psi_i\rangle$ con probabilidad p_i. Por lo tanto, la evolución del operador densidad se describe por

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \xrightarrow{U} \sum_i p_i U|\psi_i\rangle\langle\psi_i|U^\dagger = U\rho U^\dagger$$

Ejemplo 3.20 Siguiendo con el Ejemplo 3.19, tomemos $U = H$, entonces el conjunto de estados puros original $\{(1/4, |+\rangle); (3/4, |1\rangle)\}$ evolucionará a $\{(1/4, |0\rangle); 3/4, |-\rangle)\}$ y su matriz densidad puede calcularse de dos maneras equivalentes:

1. A partir del conjunto de estados puros:

$$\rho' = 1/4|0\rangle\langle 0| + 3/4|-\rangle\langle -| = 5/8|0\rangle\langle 0| - 3/8|0\rangle\langle 1| - 3/8|1\rangle\langle 0| + 3/8|1\rangle\langle 1| = \begin{pmatrix} 5/8 & -3/8 \\ -3/8 & 3/8 \end{pmatrix}$$

2. O utilizando la igualdad dada más arriba: $\rho' = H\rho H^\dagger = H\rho H$.

Medición Supongamos que realizamos una medición descripta por las matrices M_m. Si el estado inicial era $|\psi_i\rangle$, entonces la probabilidad de obtener el resultado m es

$$p(m|i) = \langle\psi_i|M_m^\dagger M_m|\psi_i\rangle \overset{\text{Cor.3.15}}{=} \mathsf{tr}(M_m^\dagger M_m|\psi_i\rangle\langle\psi_i|)$$

Usando la ley de probabilidades totales, la probabilidad de obtener el resultado m es

$$\begin{aligned} p(m) &= \sum_i p(m|i)p_i \\ &= \sum_i p_i \mathsf{tr}(M_m^\dagger M_m|\psi_i\rangle\langle\psi_i|) \\ &= \mathsf{tr}(\sum_i p_i M_m^\dagger M_m|\psi_i\rangle\langle\psi_i|) \\ &= \mathsf{tr}(M_m^\dagger M_m \sum_i p_i|\psi_i\rangle\langle\psi_i|) \\ &= \mathsf{tr}(M_m^\dagger M_m \rho) \end{aligned}$$

Si el estado inicial era $|\psi_i\rangle$, el estado luego de obtener el resultado m será

$$|\psi_i^m\rangle = \frac{M_m|\psi_i\rangle}{\sqrt{\langle\psi_i|M_m^\dagger M_m|\psi_i\rangle}}$$

Por lo tanto, luego de una medición que de resultado m tendremos el conjunto de estados $|\psi_i^m\rangle$, con probabilidades $p(i|m)$ respectivamente. Por lo tanto, el operador densidad ρ_m correspondiente es

$$\rho_m = \sum_i p(i|m)|\psi_i^m\rangle\langle\psi_i^m| = \sum_i p(i|m)\frac{M_m|\psi_i\rangle\langle\psi_i|M_m^\dagger}{\langle\psi_i|M_m^\dagger M_m|\psi_i\rangle} \tag{3.1}$$

Pero, usando teoría de probabilidad condicional,

$$p(i|m) = \frac{p(m\cap i)}{p(m)} = \frac{p(m|i)p_i}{p(m)} = p_i\frac{\mathsf{tr}(M_m^\dagger M_m|\psi_i\rangle\langle\psi_i|)}{\mathsf{tr}(M_m^\dagger M_m\rho)} = p_i\frac{\langle\psi_i|M_m^\dagger M_m|\psi_i\rangle}{\mathsf{tr}(M_m^\dagger M_m\rho)}$$

Substituyendo en (3.1), obtenemos

$$\rho_m = \sum_i p_i\frac{M_m|\psi_i\rangle\langle\psi_i|M_m^\dagger}{\mathsf{tr}(M_m^\dagger M_m\rho)} = \frac{M_m\rho M_m^\dagger}{\mathsf{tr}(M_m^\dagger M_m\rho)}$$

Ejemplo 3.21 Volviendo al conjunto de estados del Ejemplo 3.19, tenemos

$$\rho = {}^1\!/_8|0\rangle\langle 0| + {}^1\!/_8|0\rangle\langle 1| + {}^1\!/_8|1\rangle\langle 0| + {}^7\!/_8|1\rangle\langle 1|$$

que corresponde a la matriz densidad del conjunto de estados $\{(\frac{1}{4}, |+\rangle); (\frac{3}{4}, |1\rangle)\}$. Utilizaremos la medición proyectiva $\{P_0, P_1\}$ con $P_0 = |0\rangle\langle 0|$ y $P_1 = |1\rangle\langle 1|$.

Entonces, la probabilidad de medir 0 viene dada por

$$\begin{aligned}\mathsf{tr}(P_0^\dagger P_0 \rho) &= |0\rangle\langle 0|(1/8|0\rangle\langle 0| + 1/8|0\rangle\langle 1| + 1/8|1\rangle\langle 0| + 7/8|1\rangle\langle 1|) \\ &= \mathsf{tr}(1/8|0\rangle\langle 0| + 1/8|0\rangle\langle 1|) \\ &= 1/8\,\mathsf{tr}(|0\rangle\langle 0|) + 1/8\,\mathsf{tr}(|0\rangle\langle 0|) \\ &= 1/8\end{aligned}$$

Similarmente, la probabilidad de medir 1 por

$$\mathsf{tr}(|1\rangle\langle 1|\rho) = 1/8\,\mathsf{tr}(|1\rangle\langle 0|) + 7/8\,\mathsf{tr}(|1\rangle\langle 1|) = 7/8$$

Podemos ver que el conjunto de estados está en el estado $|1\rangle$ con probabilidad $3/4$. Si ese es efectivamente el estado inicial, la probabilidad de medir 1 sería 1. Por otro lado, en el estado $|+\rangle$, la probabilidad de medir 1 es $1/2$. De ahí que la probabilidad de medir 1 es

$$3/4 \times 1 + 1/4 \times 1/2 = 7/8$$

tal y como dedujimos con la traza.
Luego de realizar la medición, si se midió 1, el estado del sistema podrá ser descripto por el operador siguiente:

$$\rho_1 = \frac{P_1 \rho P_1^\dagger}{7/8} = \frac{7/8|1\rangle\langle 1|}{7/8} = |1\rangle\langle 1|$$

Efectivamente, si se midió 1 y el estado inicial era $|+\rangle$, el estado final será $|1\rangle$, pero lo mismo pasa si el estado inicial era $|1\rangle$, por lo que la matriz densidad es la matriz densidad del conjunto de estados $\{1, |1\rangle\}$.

Definición 3.22 Un sistema cuántico donde el estado $|\psi\rangle$ se conoce exactamente se dice que está en un *estado puro*. En este caso, el operador densidad es simplemente $\rho = |\psi\rangle\langle\psi|$. Si no es un estado puro, ρ está en un *estado mixto* (o mezcla), o que es una mezcla de diferentes estados puros.

Teorema 3.23 Para todo operador densidad ρ se tiene $\mathsf{tr}(\rho^2) \leq 1$.
Más aún, la igualdad se cumple si y sólo si ρ está en un estado puro

Ejercicio 3.24. Probar el Teorema 3.23

Teorema 3.25 Un estado cuántico que está en estado ρ_i con probabilidad p_i, puede ser descripto por la matriz densidad $\sum_i p_i \rho_i$.

Demostración. Supongamos que ρ_i viene de un conjunto $\{p_{ij}, |\psi_{ij}\rangle\}$ de estados puros (con i fijo). Por lo tanto, la probabilidad de estar en el estado $|\psi_{ij}\rangle$ viene dada por $p_i p_{ij}$. Es decir que la matriz densidad es $\rho = \sum_i \sum_j p_i p_{ij} |\psi_{ij}\rangle\langle\psi_{ij}| = \sum_i p_i \rho_i$. □

3.2.3. Propiedades generales del operador densidad

Definición 3.26 (Operador positivo) Un operador A se dice *positivo* si para todo vector $|\psi\rangle$, $\langle\psi|A|\psi\rangle \geq 0$. Si $\langle\psi|A|\psi\rangle > 0$ para todo $|\psi\rangle \neq 0$, decimos que A es *definido positivo*.

Teorema 3.27 Si A es un operador positivo, entonces existe una descomposición $A = \sum_j \lambda_j |j\rangle\langle j|$ donde los vectores $|j\rangle$ son ortonormales y $\lambda_j \in \mathbb{R}_0^+$ son autovalores de A. □

Teorema 3.28 (Caracterización de operadores densidad) Un operador ρ es el operador densidad de un conjunto $\{p_i, |\psi_i\rangle\}$ si y sólo si satisface las siguientes condiciones:

1. $\mathsf{tr}(\rho) = 1$
2. ρ es un operador positivo

Demostración.

⇒) Sea $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$ un operador densidad. Entonces,

1. $\mathsf{tr}(\rho) = \sum_i p_i \mathsf{tr}(|\psi_i\rangle\langle\psi_i|) = \sum_i p_i = 1$.
2. Sea $|\varphi\rangle$ un vector arbitrario en el espacio de estados. Entonces,

$$\langle\varphi|\rho|\varphi\rangle = \langle\varphi| \left(\sum_i p_i |\psi_i\rangle\langle\psi_i| \right) |\varphi\rangle = \sum_i p_i \langle\varphi|\psi_i\rangle\langle\psi_i|\varphi\rangle = \sum_i p_i |\langle\varphi|\psi_i\rangle|^2 \geq 0$$

⇐) Sea ρ cualquier operador positivo con traza igual a 1. Como ρ es positivo, usando el Teorema 3.27 tenemos $\rho = \sum_j \lambda_j |j\rangle\langle j|$, donde los vectores $|j\rangle$ son ortogonales y $\lambda_j \in \mathbb{R}_0^+$ son autovalores de ρ. Por la condición de traza 1, tenemos $\sum_j \lambda_j = 1$. Por lo tanto, un sistema en el estado $|j\rangle$ con probabilidad λ_j tendrá un operador de densidad ρ. □

El Teorema 3.28 nos permite reformular el Postulado 1 para no depender de vectores, y podemos entonces escribir todos los postulados en términos del operador densidad.

Postulado 1. Todo sistema físico cuántico aislado tiene asociado un espacio vectorial complejo con producto escalar conocido como el *espacio de estados* del sistema. El sistema se describe completamente por su *operador densidad*, el cual es un operador positivo ρ con traza 1, que actúa en el espacio de estados del sistema. Si un sistema cuántico está en estado ρ_i con probabilidad p_i, entonces el operador densidad del sistema es $\sum_i p_i \rho_i$.

Postulado 2. La evolución de un sistema físico cuántico aislado se describe por una *transformación unitaria*. Es decir, el estado ρ del sistema en el tiempo t_1 se relaciona con el estado ρ' del sistema en el tiempo t_2 a través del operador unitario U, el cual sólo depende de los tiempos t_1 y t_2.

$$\rho' = U\rho U^\dagger$$

Postulado 3. La medición cuántica se describe por una colección $\{M_m\}$ de *matrices de medición*. Dichas matrices actúan en el espacio de estados del sistema que se mide. El índice m refiere a los resultados posibles de la medición. Si el estado del sistema es ρ, inmediatamente antes de la medición, entonces la probabilidad de que el resultado m ocurra viene dado por

$$p(m) = \mathsf{tr}(M_m^\dagger M_m \rho)$$

y el estado del sistema luego de la medición es

$$\frac{M_m \rho M_m^\dagger}{\mathsf{tr}(M_m^\dagger M_m \rho)}$$

Las matrices satisfacen la ecuación de completitud,

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

Postulado 4. El espacio de estados de un sistema físico compuesto es el producto tensorial de los espacios de estados de los componentes. Más aún, si tenemos sistemas enumerados de 1 a n, donde el sistema i está en el estado ρ_i, el estado conjunto del sistema total es $\rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_n$.

3.2.4. El operador densidad reducido

Uno de los usos más interesantes del operador densidad es para describir subsistemas de un sistema cuántico compuesto. Tal descripción viene dada por el *operador densidad reducido*.

Definición 3.29 Sean A y B dos sistemas físicos tales que su estado es descripto por el operador densidad ρ^{AB}. El operador densidad reducido para A se define por

$$\rho^A = \mathsf{tr}_B(\rho^{AB})$$

donde tr_B es la *traza parcial sobre el sistema* B, y es un operador lineal definido por

$$\mathsf{tr}_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) = |a_1\rangle\langle a_2|\mathsf{tr}(|b_1\rangle\langle b_2|) = \langle b_2|b_1\rangle|a_1\rangle\langle a_2|$$

para todo $|a_1\rangle, |a_2\rangle$ en el espacio de estados de A y $|b_1\rangle$, $|b_2\rangle$ en el espacio de estados de B.

Ejemplo 3.30 Supongamos que tenemos un sistema cuántico en el estado $\rho^{AB} = \rho \otimes \sigma$, donde ρ es el operador densidad del sistema A y σ el del sistema B. Entonces,

$$\rho^A = \mathsf{tr}_B(\rho \otimes \sigma) = \rho\mathsf{tr}(\sigma) = \rho$$

Similarmente, $\rho^B = \sigma$.

Ejemplo 3.31 Un ejemplo menos trivial es el estado de Bell $\beta_{00} = 1/\sqrt{2}(|00\rangle + |11\rangle)$, que tiene el siguiente operador densidad

$$\rho = \left(\frac{|00\rangle + |11\rangle}{\sqrt{2}}\right)\left(\frac{\langle 00| + \langle 11|}{\sqrt{2}}\right) = \frac{|00\rangle\langle 00| + |11\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 11|}{2}$$

Haciendo la traza sobre el segundo qubit obtenemos el operador densidad del primer qubit:

$$\begin{aligned}
\rho^1 &= \mathsf{tr}_2(\rho) \\
&= \frac{\mathsf{tr}_2(|00\rangle\langle 00|) + \mathsf{tr}_2(|11\rangle\langle 00|) + \mathsf{tr}_2(|00\rangle\langle 11|) + \mathsf{tr}_2(|11\rangle\langle 11|)}{2} \\
&= \frac{\mathsf{tr}_2(|0\rangle\langle 0| \otimes |0\rangle\langle 0|) + \mathsf{tr}_2(|1\rangle\langle 0| \otimes |1\rangle\langle 0|) + \mathsf{tr}_2(|0\rangle\langle 1| \otimes |0\rangle\langle 1|) + \mathsf{tr}_2(|1\rangle\langle 1| \otimes |1\rangle\langle 1|)}{2} \\
&= \frac{\langle 0|0\rangle|0\rangle\langle 0| + \langle 0|1\rangle|1\rangle\langle 0| + \langle 1|0\rangle|0\rangle\langle 1| + \langle 1|1\rangle|1\rangle\langle 1|}{2} \\
&= \frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2} \\
&= \frac{I}{2}
\end{aligned}$$

Notar que este es un estado mixto, ya que $\mathsf{tr}((I/2)^2) = 1/2 < 1$. Es decir que al estar enredados, por más que el estado de dos qubits sea un estado puro, el primer qubit sólo está en un estado mixto: es decir, un estado que no conocemos completamente.

3.2.4.1. Teleportación cuántica y el operador densidad reducido

Podemos usar el operador densidad reducido para analizar el algoritmo de teleportación. Cuando presentamos el algoritmo de teleportación (Sección 1.6.2) dijimos que no contradice la teoría de la relatividad (que entre otras cosas determina que nada puede viajar a mayor velocidad que la luz, ni siquiera la información), ya que no hay trasmisión de información hasta que Alice no le envía (usando un canal clásico) el resultado de la medición a Bob. Podemos hacer esta afirmación de manera más rigurosa utilizando el operador densidad reducido.
Antes de que Alice haga la medición, el estado del sistema es

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

por lo que al medir los dos primeros qubits, se obtendrá

$$\begin{array}{ll}
|00\rangle(\alpha|0\rangle + \beta|1\rangle) & \text{con probabilidad } 1/4 \\
|01\rangle(\alpha|1\rangle + \beta|0\rangle) & \text{con probabilidad } 1/4 \\
|10\rangle(\alpha|0\rangle - \beta|1\rangle) & \text{con probabilidad } 1/4 \\
|11\rangle(\alpha|1\rangle - \beta|0\rangle) & \text{con probabilidad } 1/4
\end{array}$$

Por lo tanto, el operador densidad del sistema es

$$\begin{aligned}\rho = \frac{1}{4}&(|00\rangle\langle 00|(\alpha|0\rangle + \beta|1\rangle)(\alpha^*\langle 0| + \beta^*\langle 1|)\\ &+|01\rangle\langle 01|(\alpha|1\rangle + \beta|0\rangle)(\alpha^*\langle 1| + \beta^*\langle 0|)\\ &+|10\rangle\langle 10|(\alpha|0\rangle - \beta|1\rangle)(\alpha^*\langle 0| - \beta^*\langle 1|)\\ &+|11\rangle\langle 11|(\alpha|1\rangle - \beta|0\rangle)(\alpha^*\langle 1| - \beta^*\langle 0|))\end{aligned}$$

Por lo tanto, si hacemos la traza parcial sobre el sistema de Alice, obtenemos que operador densidad del sistema de Bob es

$$\begin{aligned}\rho^B &= \frac{1}{4}((\alpha|0\rangle + \beta|1\rangle)(\alpha^*\langle 0| + \beta^*\langle 1|) + (\alpha|1\rangle + \beta|0\rangle)(\alpha^*\langle 1| + \beta^*\langle 0|)\\ &\quad + (\alpha|0\rangle - \beta|1\rangle)(\alpha^*\langle 0| - \beta^*\langle 1|) + (\alpha|1\rangle - \beta|0\rangle)(\alpha^*\langle 1| - \beta^*\langle 0|))\\ &= \frac{2(|\alpha|^2 + |\beta|^2)|0\rangle\langle 0| + 2(|\alpha|^2 + |\beta|^2)|1\rangle\langle 1|}{4}\\ &= \frac{|0\rangle\langle 0| + |1\rangle\langle 0|}{2}\\ &= \frac{I}{2}\end{aligned}$$

Por lo tanto, el estado de Bob *después* de que Alice hizo la medición, pero *antes* de que Bob obtuvo el resultado de esa medición es $I/2$. Este estado no depende del estado $|\psi\rangle$ que se transmitió, y por lo tanto, cualquier medición que haga Bob no contendrá información sobre $|\psi\rangle$, lo que previene que Alice use la teleportación para enviar información a mayor velocidad que la luz.

3.3. Descomposición de Schmidt

Teorema 3.32 (Descomposición de Schmidt) Sea $|\psi\rangle$ un estado puro de un sistema compuesto AB. Entonces existen estados ortonormales $|i_A\rangle$ en el sistema A y estados ortogonales $|i_B\rangle$ en el sistema B tal que

$$|\psi\rangle = \sum_i \lambda_i |i_A\rangle|i_B\rangle$$

donde para todo i, $\lambda_i \in \mathbb{R}_0^+$ tales que $\sum_i \lambda_i^2 = 1$. A los λ_i se los conoce como coeficientes de Schmidt. □

Corolario 3.33 Sea $|\psi\rangle$ un estado puro de un sistema compuesto AB. Entonces

$$\rho^A = \sum_i \lambda_i^2 |i_A\rangle\langle i_A| \qquad \text{y} \qquad \rho^B = \sum_i \lambda_i^2 |i_B\rangle\langle i_B|$$

Demostración. Por el Teorema 3.32, $|\psi\rangle = \sum_i \lambda_i |i_A\rangle|i_B\rangle$. Por lo tanto,

$$\begin{aligned}\rho^{AB} &= \left(\sum_i \lambda_i |i_A\rangle|i_B\rangle\right)\left(\sum_j \lambda_j \langle j_A|\langle j_B|\right) \\ &= \sum_{ij} \lambda_i\lambda_j |i_A\rangle|i_B\rangle\langle j_A|\langle j_B| \\ &= \sum_{ij} \lambda_i\lambda_j (|i_A\rangle\langle j_A| \otimes |i_B\rangle\langle j_B|)\end{aligned}$$

Entonces

$$\rho^A = \mathsf{tr}_B(\rho^{AB}) = \sum_{ij} \lambda_i\lambda_j \mathsf{tr}_B(|i_A\rangle\langle j_A| \otimes |i_B\rangle\langle j_B|) = \sum_{ij} \lambda_i\lambda_j \langle i_B|j_B\rangle |i_A\rangle\langle j_A| = \sum_i \lambda_i^2 |i_A\rangle\langle i_A|$$

Análogamente, $\rho^B = \sum_i \lambda_i^2 |i_B\rangle\langle i_B|$. □

Ejemplo 3.34 Sea $|\psi\rangle = \dfrac{|00\rangle + |01\rangle + |11\rangle}{\sqrt{3}}$.
Primero debemos hallar la descomposición de Schmidt de este estado, para luego obtener los operadores densidad reducidos.
Llamamos A a la matriz de coeficientes de $|\psi\rangle$:

$$A = \frac{1}{\sqrt{3}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 1|)$$

Buscamos los autovalores de $A^\dagger A$

$$\begin{aligned}A^\dagger A &= \frac{1}{\sqrt{3}}(|0\rangle\langle 0| + |1\rangle\langle 0| + |1\rangle\langle 1|)\frac{1}{\sqrt{3}}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 1|) \\ &= \frac{1}{3}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + 2|1\rangle\langle 1|)\end{aligned}$$

Vemos que $\det(A^\dagger A - Ix) = x^2 - x + \frac{1}{9}$, por lo tanto, los autovalores de $A^\dagger A$ son

$$\lambda_1 = \frac{3+\sqrt{5}}{6} \qquad \text{y} \qquad \lambda_2 = \frac{3-\sqrt{5}}{6}$$

Sean $v_1 = \alpha_1|0\rangle + \beta_1|1\rangle$ un autovalor de norma 1 asosiado a λ_1. Entonces

$$A^\dagger A v_1 = \frac{\alpha_1 + \beta_1}{3}|0\rangle + \frac{\alpha_1 + 2\beta_1}{3}|1\rangle \tag{3.2}$$

y por otro lado

$$\lambda_1 v_1 = \frac{3+\sqrt{5}}{6}\alpha_1|0\rangle + \frac{3+\sqrt{5}}{6}\beta_1|1\rangle \tag{3.3}$$

Tomando (3.2) = (3.3), y $\|v_1\| = 1$ tenemos

$$\begin{cases} \dfrac{\alpha_1+\beta_1}{3} &= \dfrac{3+\sqrt{5}}{6}\alpha_1 \\ \dfrac{\alpha_1+2\beta_1}{3} &= \dfrac{3+\sqrt{5}}{6}\beta_1 \\ |\alpha|^2 + |\beta|^2 &= 1 \end{cases} \implies \begin{cases} |\alpha_1|^2 &= \dfrac{2}{5+\sqrt{5}} \\ |\beta_1|^2 &= \dfrac{5+\sqrt{5}}{10} \end{cases}$$

Tomamos, por ejemplo,

$$\alpha_1 = \sqrt{\frac{2}{5+\sqrt{5}}} \qquad \beta_1 = \sqrt{\frac{5+\sqrt{5}}{10}}$$

Análogamente, sea $v_2 = \alpha_2|0\rangle + \beta_2|1\rangle$ un autovector de norma 1 asociado a λ_2, entonces tomamos

$$\alpha_2 = \sqrt{\frac{2}{5-\sqrt{5}}} \qquad \beta_2 = -\sqrt{\frac{5-\sqrt{5}}{10}}$$

Sean $u_1 = \frac{Av_1}{\sqrt{\lambda_1}}$ y $u_2 = \frac{Av_2}{\sqrt{\lambda_2}}$. Por la descomposición de valores singulares de A, tenemos que $\{u_1, u_2\}$ y $\{v_1, v_2\}$ son bases ortonormales de $\mathbb{C}^2$ y $A = UDV^\dagger$, donde $U = u_1|0\rangle + u_2|1\rangle$, $D = \sqrt{\lambda_1}|00\rangle + \sqrt{\lambda_2}|11\rangle$ y $V = v_1|0\rangle + v_2|1\rangle$.
Luego, para la descomposición de Schmidt tomamos $|1_A\rangle = u_1$, $|2_A\rangle = u_2$, $|1_B\rangle = v_1$ y $|2_B\rangle = v_2$, y como coeficientes de Schmidt $\theta_1 = \sqrt{\lambda_1}$ y $\theta_2 = \sqrt{\lambda_2}$. Es decir,

$$|\psi\rangle = \theta_1|1_A\rangle|1_B\rangle + \theta_2|2_A\rangle|2_B\rangle$$

Por lo tanto, por el Corolario 3.33,

$$\rho^A = \theta_1|1_A\rangle\langle 1_A| + \theta_2|2_A\rangle\langle 2_A|$$

Por lo tanto, $\mathsf{tr}((\rho^A)^2) = \mathsf{tr}((\theta_1|1_A\rangle\langle 1_A| + \theta_2|2_A\rangle\langle 2_A|)^2) = \theta_1^4 + \theta_2^4 = \frac{7}{9}$.

Parte II

Fundamentos de lenguajes de programación

Capítulo 4

El isomorfismo de Curry-Howard

4.1. Lógica Proposicional Intuicionista en Deducción Natural

4.1.1. Gramática y pruebas

Vamos a considerar proposiciones cuyo valor de verdad no depende de cómo se intepretan. Es decir, sólo consideramos tautologías de la lógica proposicional (aquellas proposiciones cuyas tablas de verdad tienen T en todas las filas).
Con Deducción Natural daremos una caracterización sintáctica, es decir, un conjunto de fórmulas que se puedan probar en un sistema deductivo.

Definición 4.1 (Reglas de prueba) Una regla de prueba (*proof rule*), es una regla que permite deducir una proposición (conclusión) a partir de otras (premisas). Si $A_1, A_2, \ldots, A_n$ son las premisas y B es la conclusión, la regla de prueba "r" se escribe

$$\frac{A_1 \quad A_2 \quad \ldots \quad A_n}{B}\ r$$

Definición 4.2 (Prueba) Una prueba de una proposicion lógica en deducción natural se construye aplicando sucesivamente reglas de prueba.

Ejemplo 4.3 A partir de asumir A, B y C como verdaderas, podemos derivar $(A \wedge B) \wedge C$ como sigue, utilizando las reglas Hip y $\wedge_i$.

$$\dfrac{\dfrac{\dfrac{}{A}\ \text{Hip} \quad \dfrac{}{B}\ \text{Hip}}{A \wedge B}\ \wedge_i \quad \dfrac{}{C}\ \text{Hip}}{(A \wedge B) \wedge C}\ \wedge_i$$

Definición 4.4 (Secuente) La notación $A_1, \ldots, A_n \vdash B$ denota que a partir del conjunto de proposiciones $\{A_1, \ldots, A_n\}$ podemos obtener una prueba de B.
Un secuente es válido si podemos construir una prueba.

Ejemplo 4.5 A partir de la regla de hipótesis (Hip), vemos que el secuente $A \vdash A$ es válido para cualquier proposición A.

Ejemplo 4.6 Continuando con el Ejemplo 4.3, el secuente $A, B, C \vdash (A \wedge B) \wedge C$ es un secuente válido.

Ejemplo 4.7 $(A \wedge B) \Rightarrow C, A, \neg C \vdash \neg B$, es válido (¡pero aún no dimos las reglas de prueba para mostrarlo!).

$A, B \vdash A \wedge \neg B$, es un secuente inválido.

Las reglas de prueba deben permitir solo proposiciones válidas, impidiendo probar secuentes como el del Ejemplo 4.7.
A partir del Ejemplo 4.5, podemos cambiar las reglas de pruebas para que en lugar de tener proposiciones como premisas y conclusión, tengan secuentes.

Ejemplo 4.8 El Ejemplo 4.3 puede ser reescrito con reglas de prueba de secuentes como sigue.

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}\,\text{Hip} \quad \dfrac{}{B \vdash B}\,\text{Hip}}{A, B \vdash A \wedge B}\,\wedge_i \quad \dfrac{}{C \vdash C}\,\text{Hip}}{A, B, C \vdash (A \wedge B) \wedge C}\,\wedge_i \tag{4.1}$$

Más generalmente, podemos cambiar la regla Hip por una regla que nos permita probar $A_1, \ldots, A_n \vdash A_i$. En efecto, si asumimos que $A_1, \ldots, A_n$ son válidas, es posible derivar A_i simplemente con

$$\dfrac{}{A_i}\,\text{Hip}$$

A esta nueva regla que permite derivar $A_1, \ldots, A_n \vdash A_i$ la llamamos "ax", por axioma:

$$\dfrac{}{A_1, \ldots, A_n \vdash A_i}\,\text{ax}$$

Así, la derivación (4.1), podemos generalizarla como

$$\dfrac{\dfrac{\dfrac{}{A, B, C \vdash A}\,\text{ax} \quad \dfrac{}{A, B, C \vdash B}\,\text{ax}}{A, B, C \vdash A \wedge B}\,\wedge_i \quad \dfrac{}{A, B, C \vdash C}\,\text{ax}}{A, B, C \vdash (A \wedge B) \wedge C}\,\wedge_i$$

con la ventaja de que cada secuente en la regla carga con el conjunto completo de hipótesis.

La regla que introduce la implicación es la siguiente: Si a partir de la hipótesis A se puede derivar B, entonces $A \Rightarrow B$.

$$\dfrac{\begin{array}{c}\overline{A}\,\text{Hip}\\ \vdots\\ B\end{array}}{A \Rightarrow B}\,\Rightarrow_i$$

Aquí también se ve la ventaja de cargar con las hipótesis en cada premisa y conclusión, ya que esta regla, con secuentes, se puede notar como sigue

$$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\,\Rightarrow_i$$

Donde Γ es un conjunto de hipótesis extra.

Definición 4.9 (Gramática de la lógica proposicional) El lenguaje de la lógica proposicional que consideramos en este curso es el que se obtiene a partir de la siguiente gramática:

$$A ::= \top \mid \bot \mid A \Rightarrow A \mid A \wedge A \mid A \vee A$$

Donde $\top$ denota "Verdadero" y $\bot$ denota "Falso".

Observación. Si bien no consideramos $\neg A$ directamente en la gramática, es fácil ver que $\neg A$ es equivalente a $A \Rightarrow \bot$.

Definición 4.10 (Reglas de prueba) Las reglas de prueba de la lógica proposicional intuicionista son las siguientes. Notar que para cada constructor de la gramática hay reglas de introducción $_i$ y de eliminación $_e$, excepto para $\bot$, donde sólo hay una regla de eliminación.

$$\frac{}{\Gamma, A \vdash A}\ \text{ax} \qquad \frac{}{\Gamma \vdash \top}\ \top_i \qquad \frac{\Gamma \vdash \top \quad \Gamma \vdash A}{\Gamma \vdash A}\ \top_e \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash C}\ \bot_e$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}\ \Rightarrow_i \qquad \frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}\ \Rightarrow_e$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ \wedge_i \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ \wedge_{e_1} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ \wedge_{e_2}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ \vee_{i_1} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ \vee_{i_2} \qquad \frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\ \vee_e$$

Descripción de las reglas:

- La regla ax es la regla ya introducida de la hipótesis. Si en mi conjunto de hipótesis asumo que una proposición A es verdadera, entonces puedo derivar que A es verdadera.
- La regla $\top_i$ nos dice que bajo cualquier conjunto de hipótesis, $\top$ es verdadero.
- La regla $\top_e$ nos dice básicamente que podemos descartar la prueba $\Gamma \vdash \top$, o, dicho de otra manera, que en esa prueba no hay información relevante.
- La regla $\bot_e$ nos dice que si a partir de un conjunto de hipótesis podemos derivar que $\bot$ es verdadero, entonces podemos derivar cualquier cosa, ya que $\bot$ representa al falso, y esto constituiría una contradicción.
- La regla $\Rightarrow_i$ nos dice que si asumiendo A como hipótesis podemos derivar B, entonces quitando esa hipótesis podemos derivar que A implica B.
- La regla $\Rightarrow_e$ es el *modus ponens*: Si a partir de un conjunto de hipótesis podemos derivar $A \Rightarrow B$ y también A, entonces podemos derivar B.
- La regla $\wedge_i$ nos dice que si a partir de un conjunto de hipótesis podemos derivar A y también B, entonces podemos derivar $A \wedge B$.
- Las reglas $\wedge_{e_1}$ y $\wedge_{e_2}$ nos dicen que a si a partir de un conjunto de hipótesis podemos derivar $A \wedge B$, entonces podemos derivar tanto A como B.

- La regla $\vee_{i_1}$ nos dice que si a partir de un conjunto de hipótesis podemos derivar A, entonces podemos derivar $A \vee B$. La regla $\vee_{i_2}$ nos dice lo mismo si podemos derivar B en lugar de A.
- La regla $\vee_e$ nos dice que si a partir de un conjunto de hipótesis podemos derivar $A \vee B$, y además agregando la hipótesis A o la hipótesis B, podemos derivar C, entonces podemos derivar C.

Ejemplo 4.11 El secuente $\vdash A \Rightarrow (B \Rightarrow A)$ es válido. Es decir, la proposición $A \Rightarrow (B \Rightarrow A)$ es siempre válida sin asumir ninguna hipótesis para A, B y C. La prueba es la siguiente.

$$\dfrac{\dfrac{\dfrac{}{A, B \vdash A}\,\text{ax}}{A \vdash B \Rightarrow A}\Rightarrow_i}{\vdash A \Rightarrow (B \Rightarrow A)}\Rightarrow_i$$

Ejemplo 4.12 En el Ejemlo 4.7 dijimos que el secuente $(A \wedge B) \Rightarrow C, A, \neg C \vdash \neg B$, es válido. Ahora podemos probarlo.
Primero, debemos reescribir la negación con la implicación a botom. Por lo tanto, el secuente a probar es $(A \wedge B) \Rightarrow C, A, C \Rightarrow \bot \vdash B \Rightarrow \bot$. Una prueba es la siguiente.
Escribimos Γ en lugar de $(A \wedge B) \Rightarrow C, A, C \Rightarrow \bot$ por una cuestión de espacio.

$$\dfrac{\dfrac{\dfrac{}{\Gamma, B \vdash C \Rightarrow \bot}\,\text{ax} \quad \dfrac{\dfrac{}{\Gamma, B \vdash (A \wedge B) \Rightarrow C}\,\text{ax} \quad \dfrac{\dfrac{}{\Gamma, B \vdash A}\,\text{ax} \quad \dfrac{}{\Gamma, B \vdash B}\,\text{ax}}{\Gamma, B \vdash A \wedge B}\wedge_i}{\Gamma, B \vdash C}\Rightarrow_e}{\Gamma, B \vdash \bot}\Rightarrow_e}{\Gamma \vdash B \Rightarrow \bot}\Rightarrow_i$$

4.1.2. Reducción de pruebas: cut-elimination

Naturalmente, existen muchas pruebas para una misma proposicion. Por ejemplo, una prueba de $A, B \vdash A$ puede ser derivarse simplemente a partir de la regla de axioma, o con una derivación más compleja como

$$\dfrac{\dfrac{\dfrac{}{A, B, \vdash A}\,\text{ax} \quad \dfrac{}{A, B, \vdash B}\,\text{ax}}{A, B \vdash A \wedge B}\wedge_i}{A, B \vdash A}\wedge_{e_1}$$

En este ejemplo se ve que al árbol de derivación más simple para obtener $A, B \vdash A$, se le introduce una conjunción y luego se elimina la misma conjunción.
Otro ejemplo, es una prueba de $A, B \vdash A \wedge A$ la que puede ser simplemente la aplicación de la regla axioma, seguida de la introducción de la conjunción,

$$\dfrac{\dfrac{}{A, B \vdash A}\,\text{ax} \quad \dfrac{}{A, B \vdash B}\,\text{ax}}{A, B \vdash A \wedge B}\wedge_i$$

o algo más complejo como

$$\dfrac{\dfrac{\dfrac{}{A, B, A \wedge B \vdash A \wedge B}\,\text{ax}}{A, B \vdash (A \wedge B) \Rightarrow (A \wedge B)}\Rightarrow_i \quad \dfrac{\dfrac{}{A, B \vdash A}\,\text{ax} \quad \dfrac{}{A, B \vdash B}\,\text{ax}}{A, B \vdash A \wedge B}\wedge_i}{A, B \vdash A \wedge B}\Rightarrow_e$$

Aquí también se puede ver que se introduce un conectivo, $\Rightarrow$ en este caso, e inmediatamente después se elimina.
En general, se llama "cut" a la introducción seguida de la eliminación de cualquier conectivo. El proceso de "cut-elimination" es el proceso de eliminar los cuts, mediante un sistema de reescritura de derivaciones.

Definición 4.13 (Reglas de cut-elimination) Vamos a nombrar las derivaciones con $\pi_1, \ldots, \pi_n$, de manera que $\dfrac{\overset{\pi}{}}{\Gamma \vdash C}$ es la derivación llamada π que termina en la conclusión $\Gamma \vdash C$.
También definiremos informalmente la substitución de derivaciones. Notamos $(\pi_2/A)\pi_1$ al proceso de substituir en la derivación de π_1 todas las utilizaciones de la proposición A por la derivación π_2.
Las reglas, entonces, son las siguientes (una regla para los cuts possibles para cada conectivo):

$$\frac{\dfrac{}{\Gamma \vdash \top}\,\top_i \quad \dfrac{\overset{\pi}{}}{\Gamma \vdash A}}{\Gamma \vdash A}\,\top_e \longrightarrow \pi \qquad (\top)$$

$$\frac{\dfrac{\overset{\pi_1}{}}{\Gamma, A \vdash B}}{\dfrac{\Gamma \vdash A \Rightarrow B}{\Gamma \vdash B}\,\Rightarrow_i \quad \dfrac{\overset{\pi_2}{}}{\Gamma \vdash A}}\,\Rightarrow_e \longrightarrow (\pi_2/A)\pi_1 \qquad (\Rightarrow)$$

$$\frac{\dfrac{\overset{\pi_1}{}}{\Gamma \vdash A} \quad \dfrac{\overset{\pi_2}{}}{\Gamma \vdash B}}{\dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\,\wedge_{e_1}}\,\wedge_i \longrightarrow \pi_1 \qquad (\wedge_1)$$

$$\frac{\dfrac{\overset{\pi_1}{}}{\Gamma \vdash A} \quad \dfrac{\overset{\pi_2}{}}{\Gamma \vdash B}}{\dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\,\wedge_{e_2}}\,\wedge_i \longrightarrow \pi_2 \qquad (\wedge_2)$$

$$\frac{\dfrac{\dfrac{\overset{\pi_1}{}}{\Gamma \vdash A}}{\Gamma \vdash A \vee B}\,\vee_{i_1} \quad \dfrac{\overset{\pi_2}{}}{\Gamma, A \vdash C} \quad \dfrac{\overset{\pi_3}{}}{\Gamma, B \vdash C}}{\Gamma \vdash C}\,\vee_e \longrightarrow (\pi_1/A)\pi_2 \qquad (\vee_1)$$

$$\frac{\dfrac{\dfrac{\overset{\pi_1}{}}{\Gamma \vdash B}}{\Gamma \vdash A \vee B}\,\vee_{i_2} \quad \dfrac{\overset{\pi_2}{}}{\Gamma, A \vdash C} \quad \dfrac{\overset{\pi_3}{}}{\Gamma, B \vdash C}}{\Gamma \vdash C}\,\vee_e \longrightarrow (\pi_1/A)\pi_3 \qquad (\vee_2)$$

Ejemplo 4.14 En la derivación

$$\frac{\dfrac{\dfrac{}{A, B, \vdash A}\,\text{ax} \quad \dfrac{}{A, B, \vdash B}\,\text{ax}}{A, B \vdash A \wedge B}\,\wedge_i}{A, B \vdash A}\,\wedge_{e_1}$$

usando la regla $(\wedge_1)$ se obtiene la derivación

$$\pi_1 = \frac{}{A, B, \vdash A}\,\text{ax}$$

Ejemplo 4.15 En la derivación

$$\dfrac{\dfrac{\dfrac{}{A, B, A \wedge B \vdash A \wedge B}\,\text{ax}}{A, B \vdash (A \wedge B) \Rightarrow (A \wedge B)}\Rightarrow_i \quad \dfrac{\dfrac{}{A, B \vdash A}\,\text{ax} \quad \dfrac{}{A, B \vdash B}\,\text{ax}}{A, B \vdash A \wedge B}\wedge_i}{A, B \vdash A \wedge B}\Rightarrow_e$$

usando la regla $(\Rightarrow)$, donde

$$\pi_1 = \dfrac{}{A, B, A \wedge B \vdash A \wedge B}\,\text{ax} \qquad \text{y} \qquad \pi_2 = \dfrac{\dfrac{}{A, B \vdash A}\,\text{ax} \quad \dfrac{}{A, B \vdash B}\,\text{ax}}{A, B \vdash A \wedge B}\wedge_i$$

se obtiene

$$(\pi_2/A \wedge B)\pi_1 = \pi_2 = \dfrac{\dfrac{}{A, B \vdash A}\,\text{ax} \quad \dfrac{}{A, B \vdash B}\,\text{ax}}{A, B \vdash A \Rightarrow B}\wedge_i$$

4.2. Cálculo lambda (extendido) simplemente tipado

4.2.1. Gramática

En esta materia veremos una extensión del cálculo lambda simplemente tipado. El cálculo lambda (que se asume ya visto en materias anteriores), se construye a partir de variables, abstracciones y aplicaciones, es decir, la siguiente gramática

$$t ::= x \mid \lambda x.t \mid tt$$

Aquí extenderemos el cálculo lambda con algunas construcciones, las cuales no son necesarias a priori en el cálculo lambda sin tipos, ya que se pueden codificar en el cálculo lambda, sin embargo, son prácticas para no tener que codificarlas, y son necesarias en el caso del cálculo lambda simplemente tipado.

Definición 4.16 (Gramática del cálculo lambda extendido) El lenguaje del cálculo lambda extendido que consideraremos en este curso es el que se obtiene a partir de la siguiente gramática:

$$\begin{aligned} t = \; & x \mid \star \mid t; t \mid \mathsf{err}(t) \\ & \mid \lambda x.t \mid tt \\ & \mid \langle t, t \rangle \mid \pi_1 t \mid \pi_2 t \\ & \mid \mathsf{inl}(t) \mid \mathsf{inr}(t) \mid \mathsf{match}(t, x.t, y.t) \end{aligned}$$

Llamamos términos a las palabras obtenidas mediante esta gramática.

Descripción informal de la gramática:

- A las variables las anotaremos con las letras x, y, z.
- El símbolo $\star$ denota el par vacío, el fin de la ejecución.
- La construcción $t; r$ denota la secuencia: primero t, luego r.

- La construcción $\mathsf{err}(t)$ denota que el término t produce un error.
- La construcción $\lambda x.t$ denota la función cuya variable es x y cuerpo es t.
- La construcción tr denota la aplicación de t al argumento r.
- La construcción $\langle t, r\rangle$ denota el par.
- La construcción $\pi_1 t$ denota la proyección de la primera componente de un par t.
- La construcción $\pi_2 t$ denota la proyección de la segunda componente de un par t.
- La construcción $\mathsf{inl}(t)$ denota que t es la componente izquieda de un tipo suma.
- La construcción $\mathsf{inr}(t)$ denota que t es la componente derecha de un tipo suma.
- La construcción $\mathsf{match}(t, x.r, y.s)$ denota el match, el cual si machea t con $\mathsf{inl}(t')$ devuelve el resultado de aplicar $\lambda x.r$ a t' y si machea t con $\mathsf{inr}(t')$ devuelve el resultado de aplicar $\lambda y.s$ a t'.

4.2.2. Semántica operacional

4.2.2.1. Reglas de reducción

La gramática nos dice qué términos podemos escribir sintácticamente. La semántica operacional nos da el significado de los términos, al definir como operan.

Definición 4.17 (Reglas de reducción) Definimos una relación entre términos $t \longrightarrow r$, llamada reducción, como la relación que satisface las siguientes reglas:

$$\begin{aligned}
\star; t &\longrightarrow t && (\mathsf{sec})\\
(\lambda x.t)r &\longrightarrow (r/x)t && (\beta)\\
\pi_1\langle t, r\rangle &\longrightarrow t && (\pi_1)\\
\pi_2\langle t, r\rangle &\longrightarrow r && (\pi_2)\\
\mathsf{match}(\mathsf{inl}(t), x.r, y.s) &\longrightarrow (t/x)r && (\mathsf{match}_l)\\
\mathsf{match}(\mathsf{inr}(t), x.r, y.s) &\longrightarrow (t/y)s && (\mathsf{match}_r)
\end{aligned}$$

así como las reglas de congruencia que permitirán reducir un subtérmino de un término:

$$\frac{t \longrightarrow r}{t; s \longrightarrow r; s} \qquad \frac{t \longrightarrow r}{s; t \longrightarrow s; r} \qquad \frac{t \longrightarrow r}{\mathsf{err}(t) \longrightarrow \mathsf{err}(r)} \qquad \frac{t \longrightarrow r}{\lambda x t \longrightarrow \lambda x r} \qquad \frac{t \longrightarrow r}{ts \longrightarrow rs} \qquad \frac{t \longrightarrow r}{st \longrightarrow sr}$$

Ejercicio 4.18. Escribir las reglas de congruencia que faltan para que se pueda reducir dentro de cualquier término.

Observación. La cuarta regla de congruencia, que permite reducir dentro de la función, corresponde a la posibilidad de optimizar programas.

4.2.2.2. Captura de variables

Ejercicio 4.19. Reducir los siguientes términos

1. $(\lambda x.\lambda x.x)\mathsf{inl}(\star)\mathsf{inr}(\star)$
2. $(\lambda x.(\lambda y.(\lambda x.y; x))x)\star$
3. $(\lambda x.(\lambda f.(\lambda x.f\star)\mathsf{inr}(\star))(\lambda y.y; x))\mathsf{inl}(\star)$

Tenemos que definir precisamente qué significa $(r/x)t$. Damos una definición inductiva:

$$
\begin{aligned}
(r/x)x &= r\\
(r/x)y &= y\\
(r/x)\star &= \star\\
(r/x)t; s &= (r/x)t; (r/x)s\\
(r/x)\mathsf{err}(t) &= \mathsf{err}((r/x)t)\\
(r/x)(\lambda x.t) &= \lambda x.t\\
(r/x)(\lambda y.t) &= \lambda y.(r/x)t \quad \text{Si } y \notin \mathsf{FV}(r)\\
(r/x)(\lambda y.t) &= \lambda z.(r/x)(z/y)t \quad \text{Si } y \in \mathsf{FV}(r)\\
(r/x)(ts) &= (r/x)t(r/x)s\\
(r/x)\langle t, s\rangle &= \langle (r/x)t, (r/x)s\rangle\\
(r/x)\pi_1 t &= \pi_1(r/x)t\\
(r/x)\pi_2 t &= \pi_2(r/x)t\\
(r/x)\mathsf{inl}(t) &= \mathsf{inl}((r/x)t)\\
(r/x)\mathsf{inr}(t) &= \mathsf{inr}((r/x)t)\\
(r/x)\mathsf{match}(t, x.s_1, z.s_2) &= \mathsf{match}((r/x)t, x.s_1, z.(r/x)s_2) \quad \text{Si } z \notin \mathsf{FV}(r)\\
(r/x)\mathsf{match}(t, x.s_1, z.s_2) &= \mathsf{match}((r/x)t, x.s_1, w.(r/x)(z/w)s_2) \quad \text{Si } z \in \mathsf{FV}(r)\\
(r/x)\mathsf{match}(t, y.s_1, x.s_2) &= \mathsf{match}((r/x)t, y.(r/x)s_1, x.s_2) \quad \text{Si } y \notin \mathsf{FV}(r)\\
(r/x)\mathsf{match}(t, y.s_1, x.s_2) &= \mathsf{match}((r/x)t, y.(r/x)(y/w)s_1, z.s_2) \quad \text{Si } y \in \mathsf{FV}(r)\\
(r/x)\mathsf{match}(t, y.s_1, z.s_2) &= \mathsf{match}((r/x)t, y.(r/x)s_1, z.(r/x)s_2) \quad \text{Si } \{y, z\} \cap \mathsf{FV}(r) = \emptyset\\
(r/x)\mathsf{match}(t, y.s_1, z.s_2) &= \mathsf{match}((r/x)t, w.(r/x)(w/y)s_1, z.(r/x)s_2) \quad \text{Si } \{y, z\} \cap \mathsf{FV}(r) = \{y\}\\
(r/x)\mathsf{match}(t, y.s_1, z.s_2) &= \mathsf{match}((r/x)t, y.(r/x)s_1, w.(r/x)(w/z)s_2) \quad \text{Si } \{y, z\} \cap \mathsf{FV}(r) = \{z\}\\
(r/x)\mathsf{match}(t, y.s_1, z.s_2) &= \mathsf{match}((r/x)t, w_1.(r/x)(w_1/y)s_1, w_1.(r/x)(w_2/z)s_2) \quad \text{Si } \{y, z\} \subseteq \mathsf{FV}(r)
\end{aligned}
$$

Ejercicio 4.20. Definir, por inducción sobre t, $\mathsf{FV}(t)$.

Ejercicio 4.21. Definir, por inducción sobre t, $BV(t)$, es decir, el conjunto de variables ligadas de t ("bounded variables").

4.2.2.3. Estrategias de reducción

Primeras definiciones

Definición 4.22 Notamos $\longrightarrow^*$ al cierre reflexivo y transitivo de $\longrightarrow$.
Es decir, si $t \longrightarrow^* r$, entonces, $t = s_0 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow \ldots \longrightarrow s_n = r$, con $n \geq 0$.
Notamos $\longrightarrow^+$ al cierre transitivo de $\longrightarrow$.
Es decir, si $t \longrightarrow^* r$, $t = s_0 \longrightarrow s_1 \longrightarrow \ldots \longrightarrow s_n = u$, con $n \geq 1$.

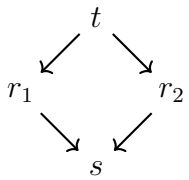
Ejemplo 4.23 $(\lambda x.x; \star)\star \longrightarrow^* \star$ porque $(\lambda x.x; \star)\star \longrightarrow \star; \star \longrightarrow \star$.
También, $(\lambda x.x; \star)\star \longrightarrow^+ \star$, ya que $(\lambda x.x; \star)\star \neq \star$.

Definición 4.24

1. Un término t está en forma normal si no existe r tal que $t \longrightarrow r$.
2. Un término t es normalizable (o tiene forma normal) si existe r en forma normal tal que $t \longrightarrow^* r$.
3. Un término t es fuertemente normalizable si no existe una secuencia infinita s_0, s_1, ... tal que $t \longrightarrow s_0 \longrightarrow s_1 \longrightarrow \ldots$. Es decir, toda secuencia de reducción comenzada en t debe ser finita y terminar en un término en forma normal.

Definición 4.25 Sea $\longrightarrow_R$ una relación binaria, y $\longrightarrow_R^*$ su cierre reflexivo y transitivo.

- $\longrightarrow_R$ satisface la *propiedad del diamante* si $t \longrightarrow_R r_1$ y $t \longrightarrow_R r_2$ implica que $r_1 \longrightarrow_R s$ y $r_2 \longrightarrow_R s$ para algún s.



- $\longrightarrow_R$ es *Church-Rosser* o *confluente* si $\longrightarrow_R^*$ satisface la propiedad del diamante. Es decir, si $t \longrightarrow_R^* r_1$ y $t \longrightarrow_R^* r_2$ implica que $r_1 \longrightarrow_R^* s$ y $r_2 \longrightarrow_R^* s$ para algún s.
- $\longrightarrow_R$ tiene *formas normales únicas* si $t \longrightarrow_R^* r_1$ y $t \longrightarrow_R^* r_2$, para términos en forma normal r_1 y r_2, implica $r_1 = r_2$.

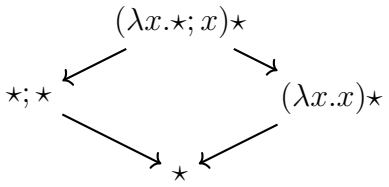
Lema 4.26

1. Si $\longrightarrow_R$ satisface la propiedad del diamante, entonces es Church-Rosser.
2. Si $\longrightarrow_R$ es Church-Rosser, entonces tiene formas normales únicas.

Ejercicio 4.27. Demostrar el Lemma 4.26

Teorema 4.28 La relación definida en la Sección 4.2.2 (semántica operacional) es Church-Rosser. □

Ejemplo 4.29



Pero esta propiedad, cuando hay términos que no terminan, no es suficiente, como veremos en los siguientes ejemplos.

Ejemplo 4.30 Sea

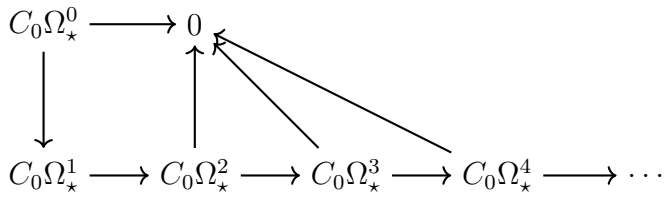
$$\Omega_\star = (\lambda x.xx\star)(\lambda x.xx\star)$$

Es facil ver que $\Omega_\star \longrightarrow \Omega_\star\star \longrightarrow \Omega_\star \star\star \longrightarrow \Omega_\star \star\star\star \longrightarrow \cdots$.
Entonces:

$$\begin{aligned}\mathsf{match}(\mathsf{inl}(\star), x.x, y.\Omega_\star) &= \mathsf{match}(\mathsf{inl}(\star), x.x, y.\Omega_\star\star)\\ &\longrightarrow \mathsf{match}(\mathsf{inl}(\star), x.x, y.\Omega_\star \star\star)\\ &\longrightarrow \mathsf{match}(\mathsf{inl}(\star), x.x, y.\Omega_\star \star\star\star)\\ &\longrightarrow \cdots \longrightarrow \infty\end{aligned}$$

$\mathsf{match}(\mathsf{inl}(\star), x.x, y.\Omega_\star)$ tiene un único resultado que es $\star$, pero no cualquier camino llega a él.
Solución (en este caso): cuando hay un match, reducir primero el match antes que sus ramas. Ésto, como veremos luego, es una *estrategia*.

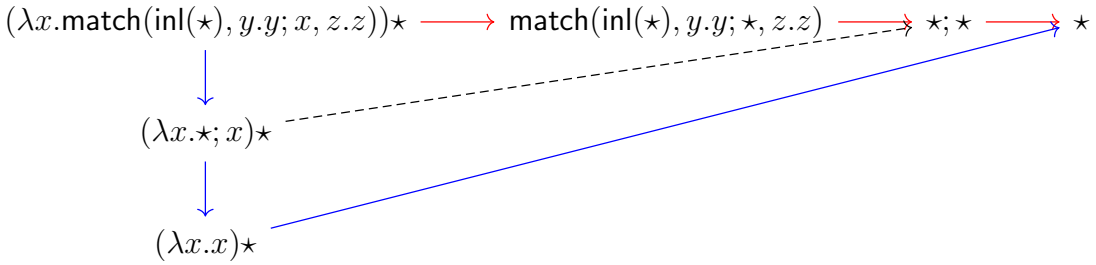
Ejemplo 4.31 Sea $C_0 = \lambda x.0$ y $\Omega^n_\star = \Omega_\star \underbrace{\star\cdots\star}_{n}$.



La noción de *estrategia de reducción* permite definir el orden en el cual se debe reducir un término.

Definición 4.32 Llamamos *redex* (por *Red*ucible *Ex*pression) a un subtérmino de un término que puede reducir.

Reducción débil Ejemplo motivador:



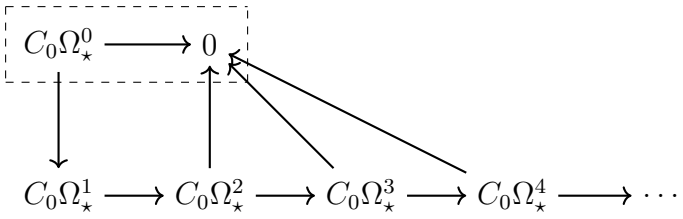
- La dirección $\rightarrow$ dice qué sucede cuando se ejecuta el programa.
- La dirección $\downarrow$ comienza a ejecutar el programa antes de recibir los argumentos, es decir, no ejecuta el programa sino que lo optimiza.

Definición 4.33 Una estrategia de reducción es *débil* si no reduce nunca el cuerpo de una función, es decir, si no reduce bajo λ.

Observación. La estrategia débil no optimiza programa, los ejecuta. Sólo hace falta para ésto eliminar la regla

$$\frac{t \longrightarrow u}{\lambda x.t \longrightarrow \lambda x.u}$$

Call-by-name



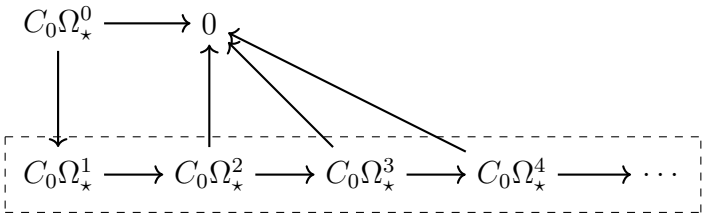
Definición 4.34 La estrategia *call-by-name* reduce siempre el redex de más a la izquierda. En caso de ser además débil, será el más a la izquierda que no esté debajo de un λ.

Teorema 4.35 (Estandarización) Si un término reduce a un término en forma normal, entonces la estrategia call-by-name termina. □

Una ventaja de ésta estrategia es el teorema de estandarización. Otra ventaja es que si tenemos, por ejemplo $(\lambda x.\star)(\mathit{Fact}\ 10)$ no necesitamos calcular el factorial de 10. Por otro lado, si tenemos $(\lambda x.\langle x, x\rangle)(\mathit{Fact}\ 10)$, tendremos que calcular el factorial de 10 dos veces. De todas maneras, la mayoría de los lenguajes que usan call-by-name usan alguna manera de "compartir" información (por ejemplo, con punteros que dicen que $(\lambda x.\langle x, x\rangle)(\mathit{Fact}\ 10)$ reduce a $\langle x, x\rangle$, donde x es un puntero a $\mathit{Fact}\ 10$). A eso se le llama *reducción lazy*.

Ejercicio 4.36. Escribir las reglas de reducción y congruencia que implementan call-by-name.

Call-by-value



Definición 4.37 A los términos t tales que $\mathsf{FV}(t) = \emptyset$ y que t esté en forma normal, se les llaman *valores*.

Definición 4.38 La estrategia *call-by-value* consiste en evaluar siempre los argumentos antes de pasarlos a la función. La idea es que

$$(\lambda x.t)v$$

reduce sólo cuando v esté en forma normal (si la estrategia es débil, y sólo reducimos términos cerrados, v es un valor).

En $(\lambda x.\langle x, x\rangle)(\mathit{Fact}\ 10)$ comenzamos por reducir el factorial, obtenemos 3628800 y recién ahí lo pasamos a la función. De esa manera el factorial es calculado una vez.

Ejercicio 4.39. Escribir las reglas que implementan call-by-value.

Observación. Un poco de pereza es necesaria: match *siempre* debe evaluar primero la condición, estemos en call-by-name o call-by-value.

4.2.3. Tipos simples

4.2.3.1. Introducción

Ejemplos motivadores:

$$\begin{aligned} (\lambda x.\pi_1 x)\lambda x.x &\longrightarrow \pi_1 \lambda x.x \\ \mathsf{match}(\lambda x.x, x.x, y.y) &\not\longrightarrow \\ (\lambda x.x) \star \lambda x.x &\longrightarrow \star \lambda x.x \end{aligned}$$

¡Todo es aplicable a todo! Sin restricciones. Proyectar una función no tiene sentido. Hacer un match sobre una función o pasarle un argumento a un $\star$, tampoco.
Idea: detectar este tipo de errores sintácticamente. Por ejemplo:

$$\frac{\lambda x.x \text{ recive un argumento y devuelve lo mismo} \quad \star \text{ es una constante}}{(\lambda x.x) \star \text{ es una constante}}$$

Es decir, deducimos que no tiene sentido pasarle un argumento a $(\lambda x.x)\star$, ya que es una constante, y lo dedujimos sin tener que *ejecutar* el programa.

En matemáticas:

$$\text{Función: Dominio} \to \text{Codominio}$$
$$\text{Cualquier conjunto}$$

Ejemplo:

$$\begin{aligned} f : \mathit{Pares} &\to \mathbb{N} \\ f(x) &\mapsto \frac{x}{2} \end{aligned}$$

¿Está bien definido $f(3+(4+1))$? Hay que determinar si $3+(4+1)$ pertenece al dominio, es decir, si es par.

En general, determinar si un objeto cualquiera pertenece a un conjunto cualquiera es un problema *indecidible*.

De todas maneras, $\frac{x}{2}$ lo podemos calcular si x es un número (y no, por ejemplo, una función), y poco importa si es par o no. Así que vamos a restringir las clases de conjuntos que se pueden utilizar como dominios. A estos conjuntos los llamamos **tipos**.

4.2.3.2. Gramática

Definición 4.40 (Gramática de tipos) El lenguaje de tipos simples que consideramos en este curso es el que se obtiene a partir de la siguiente gramática:

$$A ::= \top \mid \bot \mid A \Rightarrow A \mid A \wedge A \mid A \vee A$$

Donde $\top$ y $\bot$ son dos tipos de base.

4.2.3.3. La relación de tipado

Definición 4.41 (Contexto de tipado) Un contexto de tipado es un conjunto finito de pares de variables con tipos: $\{(x_1, A_1), \ldots, (x_n, A_n)\}$. Usualmente escribimos los contextos como $x_1 : A_1, \ldots, x_n : A_n$, y los notamos genéricamente con letras griegas mayúsculas como Γ, Δ, Ξ.

Definición 4.42 (Reglas de tipado) La relación de tipado es una relación entre un contexto de tipado Γ, un término t y un tipo A (notación $\Gamma \vdash t : A$), que se define por medio de las siguientes reglas de tipado. Notar que para cada constructor de la gramática hay reglas de introducción $_i$ y de eliminación $_e$, excepto para $\bot$, donde sólo hay una regla de eliminación.

$$\frac{}{\Gamma, x : A \vdash x : A}\ \text{ax} \qquad \frac{}{\Gamma \vdash \star : \top}\ \top_i \qquad \frac{\Gamma \vdash t : \top \quad \Gamma \vdash r : A}{\Gamma \vdash t; r : A}\ \top_e \qquad \frac{\Gamma \vdash t : \bot}{\Gamma \vdash \mathsf{err}(t) : C}\ \bot_e$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \Rightarrow B}\ \Rightarrow_i \qquad \frac{\Gamma \vdash t : A \Rightarrow B \quad \Gamma \vdash r : A}{\Gamma \vdash tr : B}\ \Rightarrow_e$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash r : B}{\Gamma \vdash \langle t, r \rangle : A \wedge B}\ \wedge_i \qquad \frac{\Gamma \vdash t : A \wedge B}{\Gamma \vdash \pi_1 t : A}\ \wedge_{e_1} \qquad \frac{\Gamma \vdash t : A \wedge B}{\Gamma \vdash \pi_2 t : B}\ \wedge_{e_2}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathsf{inl}(t) : A \vee B}\ \vee_{i_1} \qquad \frac{\Gamma \vdash t : B}{\Gamma \vdash \mathsf{inr}(t) : A \vee B}\ \vee_{i_2}$$

$$\frac{\Gamma \vdash t : A \vee B \quad \Gamma, x : A \vdash r : C \quad \Gamma, y : B \vdash s : C}{\Gamma \vdash \mathsf{match}(t, x.r, y.s) : C}\ \vee_e$$

Descripción de las reglas:

- La regla ax dice que si en el contexto de tipado se tiene $x : A$, entonces puedo derivar que $x : A$.
- La regla $\top_i$ nos dice que en cualquier contexto de tipado, $\star : \top$.
- La regla $\top_e$ nos dice que si en un contexto de tipado $t : \top$, entonces $t; r$ tendrá el tipo de A, la reducción de t terminará en $\star$, el cual será descartado luego.
- La regla $\bot_e$ nos dice que si en un contexto de tipado podemos derivar $t : \bot$, entonces t es un error y podemos tiparlo con cualquier tipo, márcandolo como error.
- La regla $\Rightarrow_i$ nos dice que si en un contexto de tipado tengo $x : A$ y podemos derivar $t : B$, entonces quitando esa variable del contexto, podemos derivar que $\lambda x.T : A \Rightarrow B$.

- La regla $\Rightarrow_e$ dice que si en un contexto de tipado podemos derivar que t tiene el tipo de función $A \Rightarrow B$, y que $r : A$, entonces la aplicación de t a r tendrá el tipo B.

- La regla $\wedge_i$ nos dice que si en un contexto de tipado podemos derivar $t : A$ y también $r : B$, entonces podemos derivar $\langle t, r\rangle : A \wedge B$.

- Las reglas $\wedge_{e_1}$ y $\wedge_{e_2}$ nos dicen que a si en un contexto de tipado podemos derivar $t : A \wedge B$, entonces podemos derivar tanto $\pi_1 t : A$ como $\pi_2 t : B$.

- La regla $\vee_{i_1}$ nos dice que si en un contexto de tipado podemos derivar $t : A$, entonces podemos derivar $\mathsf{inl}(t) : A \vee B$. La regla $\vee_{i_2}$ nos dice lo mismo si podemos derivar $t : B$ en lugar de $t : A$.

- La regla $\vee_e$ nos dice que si en un contexto de tipado podemos derivar $t : A \vee B$, y además agregando la variable $x : A$ o la $y : B$, podemos derivar $r : C$ y $s : C$ respectivamente, entonces podemos derivar $\mathsf{match}(t, x.y, s.\, :)C$.

Ejemplo 4.43 La derivación $\vdash \lambda x.\lambda y.x : A \Rightarrow (B \Rightarrow A)$ es válida. La prueba es la siguiente.

$$\dfrac{\dfrac{\dfrac{}{x : A, y : B \vdash x : A}\,\text{ax}}{x : A \vdash \lambda y.x : B \Rightarrow A}\,\Rightarrow_i}{\vdash \lambda x.\lambda y.x : A \Rightarrow (B \Rightarrow A)}\,\Rightarrow_i$$

Ejemplo 4.44 Sea $\Gamma = x : (A \wedge B) \Rightarrow C, y : A, z : C \Rightarrow \bot$ y $\Delta = \Gamma, w : B$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Delta \vdash z : C \Rightarrow \bot}\,\text{ax} \quad \dfrac{\dfrac{}{\Delta \vdash x : (A \wedge B) \Rightarrow C}\,\text{ax} \quad \dfrac{\dfrac{}{\Delta \vdash y : A}\,\text{ax} \quad \dfrac{}{\Delta \vdash w : B}\,\text{ax}}{\Delta \vdash \langle y, w\rangle : A \wedge B}\,\wedge_i}{\Delta \vdash x\langle y, w\rangle : C}\,\Rightarrow_e}{\Delta \vdash z(x\langle y, w\rangle) : \bot}\,\Rightarrow_e}{\Gamma \vdash \lambda w.z(x\langle y, w\rangle) : B \Rightarrow \bot}\,\Rightarrow_i$$

Ejemplo 4.45 Sean $\Delta = x : (\top \vee \top) \Rightarrow \top$, y $\Gamma = \Delta, y : \top$. Entonces,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Delta \vdash x : (\top \vee \top) \Rightarrow \top}\,\text{ax} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash y : \top}\,\text{ax}}{\Gamma \vdash y; \mathsf{inl}(\star) : \top \vee \top}\,\vee_i \quad \dfrac{}{\Gamma \vdash \star : \top}\,\top_i}{\Gamma \vdash y; \mathsf{inl}(\star) : \top \vee \top}\,\top_e}{\Delta \vdash \lambda y.y; \mathsf{inl}(\star) : \top \Rightarrow (\top \vee \top)}\,\Rightarrow_i \quad \dfrac{}{\Delta \vdash \star : \top}\,\top_i}{\Delta \vdash (\lambda y.y; \mathsf{inl}(\star))\star : \top \vee \top}\,\Rightarrow_e}{\Delta \vdash x((\lambda y.y; \mathsf{inl}(\star))\star) : \top}\,\Rightarrow_e}{\vdash \lambda x.x((\lambda y.y; \mathsf{inl}(\star))\star) : ((\top \vee \top) \Rightarrow \top) \Rightarrow \top}\,\Rightarrow_i$$

Ejercicio 4.46. Tipar $\lambda x.xx$.

Teorema 4.47 (Subject reduction) Si $\Gamma \vdash t : A$ y $t \longrightarrow r$ entonces $\Gamma \vdash r : A$. □

Es decir: si deducimos el tipo A para un término, con las reglas de tipado (sin "ejecutar" el programa), y luego ejecutamos el programa obteniendo r, entonces el término r tiene el mismo tipo. ¡Es exactamente lo que queríamos! La intención fue desde el principio saber qué *tipo* de resultado voy a tener al ejecutar un programa ($\top$, una función, etc), y este teorema nos dice que el sistema de tipos que definimos hace eso.

Teorema 4.48 (Normalización fuerte) Todo término tipado, termina. □

¿Qué sucede con $\Omega_\star = (\lambda x.xx\star)(\lambda x.xx\star)$? No es tipable. Es decir, no existe un tipo A tal que $\vdash \Omega_\star : A$.

Observación. En este lambda cálculo extendido podemos codificar $\mathsf{true} = \mathsf{inl}(\star)$, $\mathsf{false} = \mathsf{inr}(\star)$ y tendremos

$$\mathsf{if}\ t\ \mathsf{then}\ r\ \mathsf{else}\ s := \mathsf{match}(t, x.r, y.s)$$

En efecto,

$$\mathsf{if\ true\ then}\ r\ \mathsf{else}\ s = \mathsf{match}(\mathsf{inl}(\star), x.r, y.s) \longrightarrow r$$
$$\mathsf{if\ false\ then}\ r\ \mathsf{else}\ s = \mathsf{match}(\mathsf{inr}(\star), x.r, y.s) \longrightarrow r$$

Por lo tanto, podemos identificar el tipo Bool con $\top \vee \top$.
La compuerta Not que a true le asocia false y viceversa, la podemos codificar como

$$\mathsf{Not} := \lambda x.\mathsf{match}(x, y.\mathsf{inr}(\star), z.\mathsf{inl}(\star))$$

Ejercicio 4.49. Mostrar que $\vdash \mathsf{Not} : (\top \vee \top) \Rightarrow (\top \vee \top)$.

Ejercicio 4.50. Definir las compuertas And y Or.

Ejercicio 4.51. Dar el tipo de los términos del ejercicio anterior.

4.2.4. El isomorfismo

Tan sólo viendo las Definiciones 4.10 (reglas de prueba de la lógica proposicional) y 4.42 (reglas de tipado del cálculo lambda extendido), es evidente de que estamos hablando de lo mismo.

4.2.4.1. El cálculo lambda como un lenguaje de pruebas

Si consideramos el secuente $\vdash \top \Rightarrow \top$, podemos probarlo mediante la siguiente derivación de prueba:

$$\dfrac{\dfrac{}{\top \vdash \top}\ \mathrm{ax}}{\vdash \top \Rightarrow \top}\ \Rightarrow_i \tag{4.2}$$

pero también podemos derivarlo con

$$\dfrac{\dfrac{}{\top \Rightarrow \top \vdash \top \Rightarrow \top}\ \mathrm{ax} \quad \dfrac{\dfrac{}{\top \vdash \top}\ \mathrm{ax}}{\vdash \top \Rightarrow \top}\ \Rightarrow_i}{\dfrac{\vdash (\top \Rightarrow \top) \Rightarrow (\top \Rightarrow \top)}{\vdash \top \Rightarrow \top}\ \Rightarrow_e}\ \Rightarrow_i \tag{4.3}$$

La primer derivación, corresponde al término $\lambda x.x$, mientras que la segunda corresponde al término $(\lambda y.y)\lambda x.x$. Más aún, el secuente

$$\vdash \lambda x.x : \top \Rightarrow \top \tag{4.4}$$

está en correspondencia biunívoca con la prueba (4.2) y

$$\vdash (\lambda y.y)\lambda x.x : \top \Rightarrow \top \tag{4.5}$$

con la prueba (4.3). En general, tenemos que $\Gamma \vdash t : A$ está en correspondencia con una prueba y sólo una de $\Gamma \vdash A$, y por eso podemos acuñar el slogan

Los términos tipados del cálculo lambda son las pruebas de las proposiciones de la lógica proposicional

4.2.4.2. La semántica operacional y el cut-elimination

La Definición 4.13 (cut-elimination) también coincide con la Definición 4.17 (reglas de reducción). Por ejemplo, podemos ver que la derivación (4.3) reduce, mediante las reglas de cut-elimination, a la derivación (4.2), de la misma manera que el término (4.4) reduce al término (4.5) usando las reglas de reducción.

- La regla $(\top)$ coincide con la regla (sec): Se trata de la introducción de $\top$ $(\star)$ seguido de su eliminación (la secuencia).

$$\frac{\dfrac{}{\Gamma \vdash \top}\,\top_i \quad \dfrac{\pi}{\Gamma \vdash A}}{\Gamma \vdash A}\,\top_e \longrightarrow \pi \qquad\qquad \star; t \longrightarrow t$$

- La regla $(\Rightarrow)$ coincide con la regla (β): Se trata de la introducción de $\Rightarrow$ (una lambda abstracción) seguida de su eliminación (la aplicación).

$$\frac{\dfrac{\dfrac{\pi_1}{\Gamma, A \vdash B}}{\Gamma \vdash A \Rightarrow B}\,\Rightarrow_i \quad \dfrac{\pi_2}{\Gamma \vdash A}}{\Gamma \vdash B}\,\Rightarrow_e \longrightarrow (\pi_2/A)\pi_1 \qquad\qquad (\lambda x.t)r \longrightarrow (r/x)t$$

- Las reglas $(\wedge_1)$ y $(\wedge_2)$ coinciden con las reglas (π_1) y (π_2): Se trata de la introducción de $\wedge$ (un par) seguido de su eliminación (la proyección).

$$\frac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\,\wedge_i}{\Gamma \vdash A}\,\wedge_{e_1} \longrightarrow \pi_1 \qquad\qquad \pi_1\langle t, r\rangle \longrightarrow t$$

$$\frac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\,\wedge_i}{\Gamma \vdash B}\,\wedge_{e_2} \longrightarrow \pi_2 \qquad\qquad \pi_2\langle t, r\rangle \longrightarrow r$$

- Las reglas $(\vee_1)$ y $(\vee_2)$ coinciden con las reglas (match_l) y (match_r): Se trata de la introducción de $\vee$ (inl o inr) seguido de su eliminación (el match).

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A}}{\Gamma \vdash A \vee B}\vee_{i_1} \quad \dfrac{\pi_2}{\Gamma, A \vdash C} \quad \dfrac{\pi_3}{\Gamma, B \vdash C}}{\Gamma \vdash C}\vee_e \longrightarrow (\pi_1/A)\pi_2$$

$$\mathsf{match}(\mathsf{inl}(t), x.r, y.s) \longrightarrow (t/x)r$$

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash B}}{\Gamma \vdash A \vee B}\vee_{i_2} \quad \dfrac{\pi_2}{\Gamma, A \vdash C} \quad \dfrac{\pi_3}{\Gamma, B \vdash C}}{\Gamma \vdash C}\vee_e \longrightarrow (\pi_1/A)\pi_3$$

$$\mathsf{match}(\mathsf{inr}(t), x.r, y.s) \longrightarrow (t/y)s$$

Capítulo 5

Semántica denotacional

5.1. Introducción a la teoría de categorías

5.1.1. Primeras definiciones

Definición 5.1 Una categoría $\mathbf{C}$ se compone de:

1. Una colección de objetos $\mathbf{Obj}(\mathbf{C})$.
2. Una colección de flechas o morfismos $\mathbf{Arr}(\mathbf{C})$.
3. Operaciones que asignan a cada flecha f un objeto $dom\ f$ (su dominio) y un objeto $cod\ f$ (su codominio) (escribimos $f : A \to B$ o $A \xrightarrow{f} B$ para indicar que $dom\ f = A$ y $cod\ f = B$). A la colección de todas las flechas con dominio A y codominio B la escribimos $\mathsf{Hom}_{\mathbf{C}}(A, B)$.
4. Un operador de composición que asigna a cada par de flechas $A \xrightarrow{f} B$ y $B \xrightarrow{g} C$, la flecha composición $A \xrightarrow{f \circ g} C$ que satisface la siguiente ley de asociatividad:

   $$\text{Para todas las flechas } A \xrightarrow{f} B, B \xrightarrow{g} C, C \xrightarrow{h} D, \qquad h \circ (g \circ f) = (h \circ g) \circ f$$

5. Para cada objeto A existe una flecha identidad $A \xrightarrow{\mathsf{Id}_A} A$ que satisface la ley de indentidad:

   $$\text{Para toda flecha } A \xrightarrow{f} B, \qquad \mathsf{Id}_B \circ f = f \text{ y } f \circ \mathsf{Id}_A = f$$

Ejemplo 5.2 La categoría **Set** tiene como objetos a los conjuntos y como flechas a las funciones totales entre conjuntos. La composición es la composición de funciones, y las flechas identidad son las funciones identidad.

Ejemplo 5.3 Un orden parcial $\leq_P$ en sobre un conjunto P es una relación reflexiva, transitiva y antisimétrica en los elementos de P, es decir, es una relación para la cual, para todo $p, p', p'' \in P$,

1. $p \leq_P p$,

2. Si $p \leq_P p'$ y $p' \leq_P p''$, entonces $p \leq_P p''$,

3. Si $p \leq_P p'$ y $p' \leq_P p$, entonces $p = p'$

Si existe un orden parcial $\leq_P$ para un conjunto P, decimos que $(P, \leq_P)$ es un conjunto parcialmente ordenado.
Sean $(P, \leq_P)$ a $(Q, \leq_Q)$ dos conjuntos parcialmente ordenados. Una función $f : P \to Q$ preserva el orden (o es monónona) si $p \leq_P p'$ implica $f(p) \leq_Q f(p')$.
La categoría **Poset** tiene como objetos a los conjuntos parcialmente ordenados y como flechas a las funciones totales que preservan el orden.
Verificamos que esto satisface cada punto de la Definición 5.1:

1. **Obj**(**Poset**) es la colección de los conjuntos parcialmente ordenados.

2. **Arr**(**C**) es la colección de funciones totales $(P, \leq_P) \xrightarrow{f} (Q, \leq_Q)$ que preservan el orden.

3. Para cada función total que preserva el orden f con dominio P y codominio Q, tenemos $dom\ f = (P, \leq_P)$, $cod\ f = (Q, \leq_Q)$, y $f \in \mathsf{Hom}_{\mathbf{Poset}}((P, \leq_P), (Q, \leq_Q))$.

4. La composición de dos funciones totales $P \xrightarrow{f} Q$ y $Q \xrightarrow{g} R$ es una función total $g \circ f$ de P a R. Más aún, si $p \leq_P p'$, como f preserva el orden, tenemos $f(p) \leq_Q f(p')$, y, como g preserva el orden, tenemos $g(f(p)) \leq_R g(f(p'))$, por lo tanto, $g \circ f$ preserva el orden y entonces $g \circ f \in$ **Arr**(**Poset**). Finalmente, la composición de funciones es asociativa.

5. Para cada orden parcial $(P, \leq_P)$, tenemos que la función identidad Id_P preserva el órden y satisface la ley de indentidad.

Ejemplo 5.4 Un monoide $(M, *, e)$ es un conjunto M equipado con una operación binaria $*$ de un par de elementos de de M en M tal que $(x*y)*z = x*(y*z)$ para todo $x, y, z \in M$, y un elemento distinguido e tal que $e * x = x = x * e$ para todo $x \in M$.
Un homomorfismo de monoides de $(M, *_M, e_M)$ a $(N, *_N, e_N)$ es una función $f : M \to N$ tal que $f(e_M) = e_N$ y $f(x *_M y) = f(x) *_N f(y)$. La composición de dos homomorfismos de monoides es la misma que la composición de funciones en conjuntos.
La categoría **Mon** tiene a los monoides como objetos y a los homomorfismos de monoides como flechas.

Ejemplo 5.5 La categoría **Vec** es la categoría cuyos objetos son espacios vectoriales y cuyas flechas son las transformaciones lineales.

Ejemplo 5.6 La categoría **0** es la categoría tal que **Obj**(**0**) $= \emptyset$ y **Arr**(**0**) $= \emptyset$.

5.1.2. Diagramas

Definición 5.7 Un diagrama en una categoría **C** es una colección de vértices y aristas dirigidas, etiquetados consistentemente con objetos y flechas de la categoría **C**.

Un diagrama en una categoría **C** se dice conmutativo si, para cualquier par de vértices X e Y, todos los caminos en el diagrama desde X hasta Y son iguales, en el sentido de que cada camino del diagrama determina una flecha y esas flechas son iguales en **C**.
Por ejemplo, en lugar de decir $f \circ g' = g \circ f'$, puedo decir que el siguiente diagrama conmuta:

$$\begin{array}{ccc} X & \xrightarrow{f'} & Z \\ \downarrow{\scriptstyle g'} & & \downarrow{\scriptstyle g} \\ W & \xrightarrow{f} & Y \end{array}$$

Teorema 5.8 Si los dos cuadros internos del siguiente diagrama conmutan, entonces el rectángulo exterior también conmuta.

$$\begin{array}{ccccc} A & \xrightarrow{f} & B & \xrightarrow{f'} & C \\ \downarrow{\scriptstyle a} & & \downarrow{\scriptstyle b} & & \downarrow{\scriptstyle c} \\ A' & \xrightarrow{g} & B' & \xrightarrow{g'} & C' \end{array}$$

Demostración.

$$\begin{aligned} & (g' \circ g) \circ a \\ (\text{asociatividad}) &= g' \circ (g \circ a) \\ (\text{conmutatividad del primer cuadrado}) &= g' \circ (b \circ f) \\ (\text{asociatividad}) &= (g' \circ b) \circ f \\ (\text{conmutatividad del segundo cuadrado}) &= (c \circ f') \circ f \\ (\text{asociatividad}) &= c \circ (f' \circ f) \end{aligned}$$

□

5.1.3. Monomorfismos, epimorfismos e isomorfismos

Definición 5.9 Una flecha $B \xrightarrow{f} C$ en una categoría **C** es un monomorfismo si, para cualquier par de flechas $A \xrightarrow{g} B$ y $A \xrightarrow{h} B$ de la categoría, la igualdad $f \circ g = f \circ h$ implica que $g = h$.

Teorema 5.10 En **Set**, los monomorfismos son exactamente las funciones inyectivas (las funciones para las cuales $f(x) = f(y)$ implica $x = y$).

Demostración. Sea $B \xrightarrow{f} C$ una función inyectiva, y sean $A \xrightarrow{g} B$, $A \xrightarrow{h} B$ tales que $f \circ g = f \circ h$, pero $g \neq h$. Entonces hay algún elemento $a \in A$ para el cual $g(a) \neq h(a)$. Pero como f es inyectiva, $f(g(a)) \neq f(h(a))$, lo que contradice la suposición de que $f \circ g = f \circ h$. Esto muestra que una función inyectiva es un monomorfismo.
Sea $B \xrightarrow{f} C$ un monomorfismo. Si f no es inyectiva, entonces existen dos elementos $b, b' \in B$ tales que $f(b) = f(b')$. Sea $A = \{a\}$ un conjunto de un solo elemento, y

$A \xrightarrow{g} B$ la función $g(a) = b$, mientras que $A \xrightarrow{h} B$ es la función $h(a) = b'$. Entonces $f(g(a)) = f(b) = f(b') = f(h(a))$, lo cual contradice la supocisión de que f es un monomorfismo. □

Definición 5.11 Una flecha $A \xrightarrow{f} B$ en una categoría **C** es un epimorfismo si, para cualquier par de flechas $B \xrightarrow{g} C$ y $B \xrightarrow{h} C$, la igualdad $g \circ f = h \circ f$.

Teorema 5.12 En **Set**, los epimorfismos son exactamente las funciones sobreyectivas (las funciones $f : A \to B$ para las cuales para cada $b \in B$ existe un $a \in A$ tal que $f(a) = b$).

Demostración. Ejercicio □

Definición 5.13 Una flecha $A \xrightarrow{f} B$ es un isomorfismo si existe una flecha $B \xrightarrow{f^{-1}} A$, llamada inversa de f, tal que $f^{-1} \circ f = \mathsf{Id}_A$ y $f \circ f^{-1} = \mathsf{Id}_B$. Los objetos A y B se dicen isomorfos si hay un isomorfismo entre ellos.

Teorema 5.14 En **Set**, los isomorfismos son exactamente las funciones biyectivas (eso es, inyectivas y sobreyectivas a la vez). □

5.1.4. Algunas construcciones universales a todas las categorías

5.1.4.1. Objetos iniciales y terminales

Definición 5.15 Un objeto 0 es llamado objeto inicial si, para todo objeto A, existe exactamente una flecha desde 0 a A.

Definición 5.16 De forma dual, un objeto 1 es llamado objeto terminal si, para todo objeto A, existe exactamente una flecha desde A en 1.

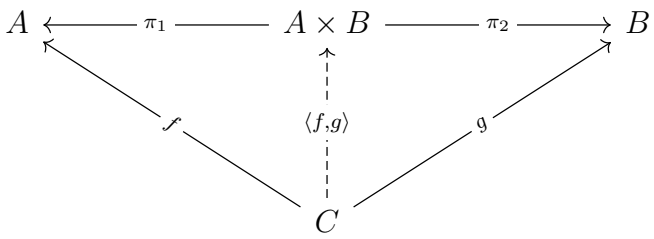
Ejemplo 5.17 En **Set** el objeto inicial es el conjunto vacío $\emptyset$. Para todo conjunto A, la función vacía es la única función de $\emptyset$ en A. Todo conjunto de un sólo elemento es un objeto terminal, ya que para todo conjunto A existe una única función de A a $\{x\}$ que mapea todos los elementos de A en x.

Ejemplo 5.18 Los objetos terminales pueden ser usados para proveer un análogo en categorías al concepto de elementos de conjuntos. La observación que motiva esto es que, en la categoría **Set**, las funciones desde un conjunto de un sólo elemento a un conjunto A están en correspondencia uno-a-uno con los elementos de A. Más aún, si $x \in A$, considerada como una flecha $1 \xrightarrow{x} A$ desde algún conjunto de un sólo elemento 1, y f es una función de A a algún otro conjunto B, entonces el elemento $f(x)$ es el único elemento de B tal que es la imagen de la composición $f \circ x$.
En términos categóricos, una flecha desde un objeto terminal a un objeto A se llama elemento global o constante de A.

5.1.4.2. Productos

Definición 5.19 Un producto de dos objetos A y B es un objeto $A \times B$, que viene con dos flechas proyección $A \times B \xrightarrow{\pi_1} A$ y $A \times B \xrightarrow{\pi_2} B$ tales que para todo objeto C y par

de flechas $C \xrightarrow{f} A$ y $C \xrightarrow{g} B$, existe exactamente una flecha $C \xrightarrow{\langle f,g\rangle} A \times B$ que hace que el siguiente diagrama conmute.



Es decir, $\pi_1 \circ \langle f, g\rangle = f$ y $\pi_2 \circ \langle f, g\rangle = g$.

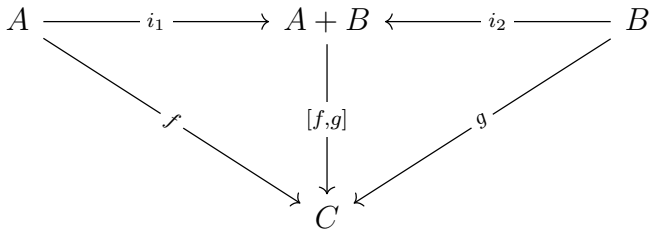
Las aristas punteadas en un diagrama se usan para representar flechas que se afirma que existen cuando el resto del diagrama se completa apropiadamente.

Definición 5.20 Si $A \times C$ y $B \times D$ son productos, entonces para todo par de flechas $A \xrightarrow{f} B$ y $C \xrightarrow{g} D$, el mapa producto $A \times C \xrightarrow{f\times g} B \times D$ es la flecha $\langle f \circ \pi_1, g \circ \pi_2\rangle$.

Observación. Notar que hay dos productos en juego en la definición anterior y sin embargo hablamos de π_1 y π_2 sin decir a cuál nos referíamos. Esto se puede hacer ya que la teoría de categorías es un formalismo "fuertemente tipado", en el sentido de que sólo un par de proyecciones puede ir allí, para que la flecha tenga sentido.

La noción dual al producto, el coproducto, corresponde, en teoría de conjuntos, a la unión disjunta.

Definición 5.21 Un coproducto de dos objetos A y B es un objeto $A+B$, que viene con dos flechas de inyección $A \xrightarrow{i_1} A+B$ y $B \xrightarrow{i_2} A+B$ tales que para todo objeto C y par de flechas $A \xrightarrow{f} C$ y $B \xrightarrow{g} C$, existe exactamente una flecha $A+B \xrightarrow{[f,g]} C$ que hace que el siguiente diagrama conmute.

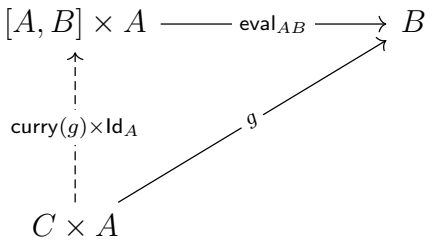


Es decir, $[f, g] \circ i_1 = f$ y $[f, g] \circ i_2 = g$.

5.1.4.3. Curryficación

Dado que si A y B son conjuntos, $\mathsf{Hom}_{\mathsf{Set}}(A, B)$ es un conjunto, tenemos que $\mathsf{Hom}_{\mathsf{Set}}(A, B) \in \mathbf{Obj}(\mathbf{Set})$. Sin embargo, no sucede en toda categoría $\mathbb{C}$ que dados dos objetos A y B en la categoría, $\mathsf{Hom}_{\mathbb{C}}(A, B)$ sea un objeto de la categoría. Por ejemplo, eso no sucede en la categoría **Mon**.
En las categorías que sí admiten considerar a $\mathsf{Hom}_{\mathbb{C}}(A, B)$ como un objeto de la categoría, vamos a identificar ase objeto como $[A, B]$, y vamos a dar una caracterización categórica del objeto (lo cual permitirá probar que el objeto pertenece a la categoría).

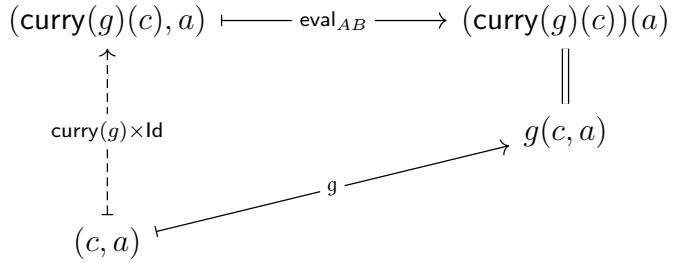
Definición 5.22 (Objeto exponencial) Sea $\mathbb{C}$ una categoría con productos y sean A y B objetos de la categoría. Un objeto $[A, B]$ es un objeto exponencial si existe una flecha $[A, B] \times A \xrightarrow{\mathsf{eval}_{AB}} B$ tal que para todo objeto C y flecha $C \times A \xrightarrow{g} B$ hay una única flecha $\mathsf{curry}(g) : C \to [A, B]$ que hace que el siguiente diagrama conmute.



Es decir, una única flecha $\mathsf{curry}(g)$ tal que

$$\mathsf{eval}_{AB} \circ (\mathsf{curry}(g) \times \mathsf{Id}_A) = g$$

Una forma intuitiva de entender la definición anterior, es evaluando los mapas. En este caso quedaría de la siguiente manera:



Definición 5.23 (CCC) Una categoría cartesiana cerrada (CCC) es una categoría con objeto terminal, productos y exponenciación.

5.1.5. Functores, transformaciones naturales y adjunciones

5.1.5.1. Functores

Definición 5.24 Sean $\mathbf{C}$ y $\mathbf{D}$ dos categorías. Un functor $F : \mathbf{C} \to \mathbf{D}$ es un mapa que lleva cada objeto $A \in \mathbf{Obj}(\mathbf{C})$ a un objeto $F(A) \in \mathbf{Obj}(\mathbf{D})$, y cada flecha $A \xrightarrow{f} B \in \mathbf{Arr}(\mathbf{C})$ a una flecha $F(A) \xrightarrow{F(f)} F(B) \in \mathbf{Arr}(\mathbf{D})$ de manera tal que para todos los objetos $A \in \mathbf{Obj}(\mathbf{C})$ y todas las flechas componibles $f, g \in \mathbf{Arr}(\mathbf{C})$ se tiene

1. $F(\mathsf{Id}_A) = \mathsf{Id}_{F(A)}$
2. $F(g \circ f) = F(g) \circ F(f)$

Ejemplo 5.25 Dado un conjunto S podemos armar el conjunto $\mathsf{List}(S)$ de listas finitas de elementos de S.
Esto define el mapa List que es la parte que actúa sobre objetos de un functor de **Set** en **Set**.

La parte que actúa sobre las flechas lleva la flecha $S \xrightarrow{f} S'$ a una función $\mathsf{List}(S) \xrightarrow{\mathsf{List}(f)} \mathsf{List}(S')$ que, dada una lista $L = [s_1, \ldots, s_n]$ mapea a la f sobre cada uno de los elementos:

$$\mathsf{List}(f)(L) = [f(s_1), \ldots, f(s_n)]$$

Ejemplo 5.26 Es facil ver que para cualquier conjunto S, $(\mathsf{List}(S), +\!\!+, [])$ es un monoide, donde $+\!\!+$ es la concatenación de listas y $[]$ es la lista vacía. En efecto, $+\!\!+$ es cerrada en listas, es asociativa, y $[]$ es su elemento neutro.
Por lo tanto, podemos considerar a List como un fucntor de **Set** en **Mon**. La parte que actúa sobre los elementos lleva cada conjunto S al monoide de listas con elementos de S. La parte que actúa sobre las flechas lleva una función f a un homomorfismo de monoides $\mathsf{List}(f) = \mathit{maplist}(f)$, cuya definición recursiva es

$$\begin{aligned} \mathit{maplist}(f)([]) &= [] \\ \mathit{maplist}(f)(L +\!\!+ L') &= \mathit{maplist}(f)(L) +\!\!+ \mathit{maplist}(f)(L') \\ \mathit{maplist}(f)([s]) &= [f(s)] \end{aligned}$$

Notar que las dos primeras líneas de esta definición, prueban que $\mathit{maplist}(f)$ es un homomorfismo de monoides.
$\mathsf{List}(S)$ es usualmente llamado el monoide libre generado por S.

Ejemplo 5.27 Sea $\mathbf{C}$ una categoría con producto $X \times Y$ para cada par de objetos X e Y. Entonces, cada objeto $A \in \mathbf{Obj}(\mathbf{C})$ determina un functor $(— \times A) : \mathbf{C} \to \mathbf{C}$ que lleva cada objeto B a $B \times A$ y cada flecha $B \xrightarrow{f} C$ a $f \times \mathsf{Id}_A$.
El — se usa para decir a dónde se coloca el argumento. Es lo que en cálculo lambda escribiríamos $\lambda x.x \times A$.

5.1.5.2. Transformaciones naturales

En la sección anterior definimos mapas de una categoría a otra (functores). Ahora vamos a definir mapas que preservan estructuras, llamadas transformaciones naturales, de un functor a otro.
Intuitivamente, la idea es la siguiente. Dados dos functores $F : \mathbf{C} \to \mathbf{D}$ y $G : \mathbf{C} \to \mathbf{D}$, podemos pensarlos como que cada uno de ellos proyecta un retrato de $\mathbf{C}$ en $\mathbf{D}$. Las transformaciones naturales surgen cuando imaginamos desplazar el retrato definido por F al retrato definido por G. Para cada objeto $A \in \mathbf{Obj}(\mathbf{C})$, definimos una flecha η_A de la imagen-F en la imagen-G de A. Para asegurar que la estructura de F es preservada por esta transformación, requerimos que para cada flecha $A \xrightarrow{f} B$ en $\mathbf{C}$, las transformaciones η_A y η_B lleven los puntos de la imagen-F de f a los puntos de la imagen-G de f. Luego de esta intuición, damos la definición formal:

Definición 5.28 Sean $\mathbf{C}$ y $\mathbf{D}$ dos categorías y sean F y G functores de $\mathbf{C}$ a $\mathbf{D}$. Una transformación natural η de F a G, notada $\eta : F \xrightarrow{\cdot} G$, es una función que asigna a cada elemento $A \in \mathbf{Obj}(\mathbf{C})$ una flecha $F(A) \xrightarrow{\eta_A} G(A) \in \mathbf{Arr}(\mathbf{D})$ tal que para cada

flecha $A \xrightarrow{f} B \in \mathbf{Arr}(\mathbf{C})$, el siguiente diagrama conmute en $\mathbf{D}$:

$$\begin{array}{ccc} F(A) & \xrightarrow{\eta_A} & G(A) \\ \Big\downarrow{\scriptstyle F(f)} & & \Big\downarrow{\scriptstyle G(f)} \\ F(B) & \xrightarrow{\eta_B} & G(B) \end{array}$$

Si cada componente η_A de η es un isomorfismo en $\mathbf{D}$, entonces a η se le llama isomorfismo natural.

Ejemplo 5.29 Para cada functor F, las componentes de la transformación natural identidad $\iota_F : F \xrightarrow{\cdot} F$ son las flechas identidad de los objetos en la imagen de F, es decir, $\iota_F = \mathsf{Id}_{F(A)}$. Más aún, ι_F es un isomorfismo natural.

Ejemplo 5.30 Sea *rev* la función que da vuelta listas, es decir, $rev_S : \mathsf{List}(S) \to \mathsf{List}(S)$ toma cualquier lista de S y la da vuelta. Por ejemplo,

$$rev_{\mathbb{N}}[5, 6, 7] = [7, 6, 5]$$

Este es un ejemplo de función polimórfica: opera exactamente igual, sin importar los elementos que componen la lista. Si le damos otros tres números, $rev_{\mathbb{N}}$ hace exactamente lo mismo que antes:

$$rev_{\mathbb{N}}[6, 7, 8] = [8, 7, 6]$$

En efecto, podemos aplicar cualquier mapeo a los elementos individuales del argumento de *rev*, incluso uno que cambie su tipo. Si $\mathbb{N} \xrightarrow{f} S$, entonces

$$maplist(f)(rev_{\mathbb{N}}[5, 6, 7]) = [f(7), f(6), f(5)]$$

En general, si $S \xrightarrow{f} T$, entonces

$$rev_T \circ maplist(f) = maplist(f) \circ rev_S$$

Y esto es precisamente lo que dice que *rev* es una transformación natural.

Ejemplo 5.31 Dado un conjunto fijo A, el mapa que lleva B a $[A, B] \times A$ puede ser extendido a un functor $F_A : \mathbf{Set} \to \mathbf{Set}$ como sigue:

$$\begin{aligned} F_A : \mathbf{Set} &\to \mathbf{Set} \\ B &\mapsto [A, B] \times A \\ (B \xrightarrow{f} C) &\mapsto (f \circ —) \times \mathsf{Id}_A \end{aligned}$$

Para que se entienda el $F_A(f)$ consideremos el siguiente diagrama

$$\begin{array}{ccc} B & \xrightarrow{F_A} & [A, B] \times A \\ \Big\downarrow{\scriptstyle f} & & \Big\downarrow{\scriptstyle F_A(f)=(f\circ —)\times \mathsf{Id}_A} \\ C & \xrightarrow{F_A} & [A, C] \times A \end{array}$$

A nivel de elementos el mapeo queda como sigue:

$$\begin{array}{ccc} b & \overset{F_A}{\longmapsto} & (a \mapsto b, a) \\ \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle F_A(f)=(f\circ\_)\times \mathsf{Id}_A} \\ f(b) & \overset{F_A}{\longmapsto} & (a \mapsto f(b), a) \end{array}$$

El hecho de que $\mathsf{eval} : F_A \dot{\longrightarrow} I_{\mathbf{Set}}$ es una transformación natural se demuestra con el siguiente diagrama conmutativo.

$$\begin{array}{ccc} F_A(C) = [A, C] \times A & \overset{\mathsf{eval}_{AC}}{\longrightarrow} & C = I_{\mathbf{Set}}(C) \\ \downarrow{\scriptstyle F_A(g)=(g\circ\_)\times \mathsf{Id}_A} & & \downarrow{\scriptstyle g=I_{\mathbf{Set}}(g)} \\ F_A(B) = [A, B] \times A & \overset{\mathsf{eval}_{AB}}{\longrightarrow} & B = I_{\mathbf{Set}}(B) \end{array}$$

Teorema 5.32 La composición de dos transformaciones naturales es una transformación natural.

Demostración. Sean $\mathbf{C}$ y $\mathbf{D}$ dos categorías, y sean F, G y H functores de $\mathbf{C}$ a $\mathbf{D}$. Sea $\sigma : F \dot{\longrightarrow} G$ y $\tau : G \dot{\longrightarrow} H$ dos transformaciones naturales.
Entonces, para cada flecha $A \xrightarrow{f} B \in \mathbf{Arr}(\mathbf{C})$ podemos hacer el siguiente diagrama

$$\begin{array}{ccccc} F(A) & \overset{\sigma_A}{\longrightarrow} & G(A) & \overset{\tau_A}{\longrightarrow} & H(A) \\ \downarrow{\scriptstyle F(f)} & & \downarrow{\scriptstyle G(f)} & & \downarrow{\scriptstyle H(f)} \\ F(B) & \overset{\sigma_B}{\longrightarrow} & G(B) & \overset{\tau_B}{\longrightarrow} & H(B) \end{array}$$

Ambos diagramas conmutan, ya que σ y τ son transformaciones naturales, por lo cual el rectángulo exterior conmuta (Teorema 5.8). Esto muestra que la transformación $(\tau \circ \sigma) : F \dot{\longrightarrow} H$ definida como $(\tau \circ \sigma)_A = \tau_A \circ \sigma_A$ es natural. □

5.1.5.3. Adjunciones

Definición 5.33 (Adjunción) Una adjunción consiste en un par de categorías $\mathbf{C}$ y $\mathbf{D}$ y un par de functores $F : \mathbf{C} \to \mathbf{D}$ y $G : \mathbf{D} \to \mathbf{C}$, tales que para cada objeto $X \in \mathbf{Obj}(\mathbf{C})$ y para cada objeto $Y \in \mathbf{Obj}(\mathbf{D})$ existe un isomirfismo

$$\mathsf{Hom}_{\mathbf{C}}(F(X), Y) \simeq \mathsf{Hom}_{\mathbf{D}}(X, G(Y))$$

que es natural en X e Y. Eso es, una transformación natural de dos variables entre $\mathsf{Hom}_{\mathbf{C}}(F(—), —)$ y $\mathsf{Hom}_{\mathbf{D}}(—, G(—))$ que preserve la estructura, ya que X e Y varían y es una bijección para todo X e Y.
Decimos que (F, G) es un par adjunto de functores, F es el adjunto a izquierda y G el adjunto a derecha. Se suele notar $F \dashv G$ o $(F, \eta) \dashv (G, \varepsilon)$.

Ejemplo 5.34 Sea $B \in \mathbf{Obj}(\mathbf{Set})$ y sea $— \times B : \mathbf{Set} \to \mathbf{Set}$ el functor producto introducido en el Ejemplo 5.27 definido como

$$\begin{aligned} — \times B : \mathbf{Set} &\to \mathbf{Set} \\ A &\mapsto B \times A \\ (A \xrightarrow{f} C) &\mapsto f \times \mathsf{Id}_B \end{aligned}$$

y sea $[B, —] : \mathbf{Set} \to \mathbf{Set}$ el functor definido como

$$\begin{aligned} [B, —] : \mathbf{Set} &\to \mathbf{Set} \\ C &\mapsto [B, C] \\ (C \xrightarrow{f} D) &\mapsto f \circ — \end{aligned}$$

Entonces existe se tiene una adjunción $— \times B \dashv [B, —]$, es decir, existe un isomorfismo

$$\mathsf{Hom}_{\mathsf{Set}}(A \times B, C) \simeq \mathsf{Hom}_{\mathsf{Set}}(A, [B, C])$$

que es natural en A y C.
Para ver esto, primero mostramos que el mapa curry de la Definición 5.22 es una bijección:

- Supongamos que $\mathsf{curry}(g) = \mathsf{curry}(h)$ para fos funciones $g, h : (C \times A) \to B$. Entonces $g = \mathsf{eval}_{AB} \circ (\mathsf{curry}(g) \times \mathsf{Id}_A) = \mathsf{eval}_{AB} \circ (\mathsf{curry}(h) \times \mathsf{Id}_A) = h$. Por lo tanto curry es inyectiva.

- Sea $g' : C \to [A, B]$, y definamos $g = \mathsf{eval}_{AB} \circ (g' \times \mathsf{Id}_A)$. Por la unicidad de $\mathsf{curry}(g)$, tenemos que $\mathsf{curry}(g) = g'$, por lo tanto también es sobreyectiva.

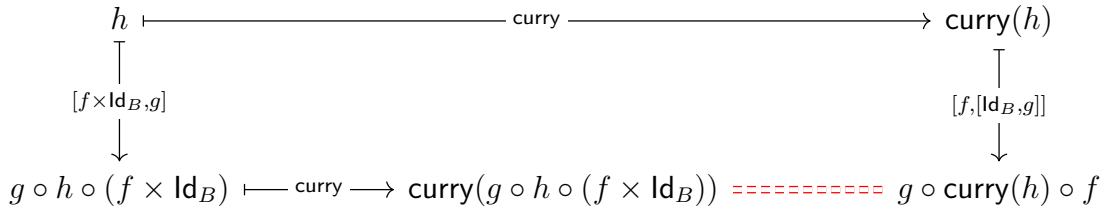
Esta biyección se suele marcar como

$$\frac{C \to [A, B]}{C \times A \to B}$$

Luego mostramos que curry es una transformación natural, es decir, que el siguiente mapa conmuta:

$$\begin{array}{ccc} [A \times B, C] & \xrightarrow{\ \mathsf{curry}\ } & [A, [B, C]] \\ \Big\downarrow{\scriptstyle [f \times \mathsf{Id}_B, g]} & & \Big\downarrow{\scriptstyle [f, [\mathsf{Id}_B, g]]} \\ [A' \times B, C'] & \xrightarrow{\ \mathsf{curry}\ } & [A', [B, C']] \end{array}$$

Esto lo podemos comprobar evaluando los mapas:



Donde la igualdad en rojo se justifica como sigue. Sea $a \in A$ y $b \in B$,

$$\begin{aligned}((g \circ \mathsf{curry}(h) \circ f)(a))(b) &= g(\mathsf{curry}(h)(f(a)))(b)\\ &= g(h(f(a), b))\\ &= g(h(f \times \mathsf{Id}_B)(a, b))\\ &= (\mathsf{curry}(g \circ h \circ (f \times \mathsf{Id}_B))(a))(b)\\ &= (\mathsf{curry}(g \circ h \circ (f \times \mathsf{Id}_B))(a))(b)\end{aligned}$$

5.2. Semántica denotacional (categórica)

5.2.1. Primeras definiciones

Sintaxis (o gramática) : Cómo escribir los términos. Cuáles son válidos y cuáles no.

Semántica: Qué significan.

Definición 5.35 (Semántica) La semántica de un lenguaje es una relación $\hookrightarrow$ que a cada expresión le asocia *algo* que le da significado.

Semántica denotacional (en programas deterministas). Para cada programa p, la relación entre las entradas y las salidas de p es una función que escribimos $[\![p]\!]$. La relación se define entonces como

$$p, e \hookrightarrow s \iff [\![p]\!]\, e = s$$

La pregunta es, obviamente, cómo definir $[\![p]\!]$. (Y ese es el tópico de este capítulo).

Semántica operacional a grandes pasos También llamada semántica operacional a grandes pasos o semántica natural. Consiste en dar una *definición inductiva* de $\hookrightarrow$ que nos relacione un término con su valor. Por ejemplo,

$$\underbrace{(\lambda x.\mathsf{match}(\mathsf{inl}(\star), y.y; x, z.z))\star}_{\text{Significado de esta expresión: } \star} \hookrightarrow \star$$

En ese ejemplo damos semántica de acuerdo a lo que calcula. Una definición inductiva para esta relación es lo que se conoce como "Intérprete" (ver, por ejemplo, [Dowek y Lévy, 2011, Capítulo 3]). Así, si considero

$$(\lambda x.\mathsf{match}(\mathsf{inl}(\star), y.x; y, z.z))\star \quad \text{y} \quad (\lambda x.\mathsf{match}(\mathsf{inl}(\star), y.y, z.z))\star$$

puedo ver que los 3 programas tienen la misma semántica.

Semántica operacional a pequeños pasos También llamada semántica por reescritura. Consiste en definir $\hookrightarrow$ *a partir de otra relación* $\longrightarrow$ que describe las etapas elementales. Ejemplo:

$$(\lambda x.\mathsf{match}(\mathsf{inl}(\star), y.y; x, z.z))\star \longrightarrow \mathsf{match}(\mathsf{inl}(\star), y.y; \star, z.z) \longrightarrow \star; \star \longrightarrow \star$$

$$t \hookrightarrow r \iff t \longrightarrow^* r \qquad \text{y } r \text{ irreducible}$$

donde $\longrightarrow^*$ es la clausura reflexiva y transitiva de $\longrightarrow$.

La no terminación. Un programa puede dar un resultado, producir un error o no terminar. Los errores se pueden considerar como resultados particulares. Para expresar programas que no terminan hay varias formas de expresar su semántica: La primera consiste en considerar que si t no termina, entonces no existe r tal que $t \hookrightarrow r$. La segunda consiste en agregar un elemento particular $\bot$ a los valores de salida y considerar que si t no termina, entonces $t \hookrightarrow \bot$. En este curso no trataremos el tema de la no terminación, sin embargo, un tratamiento fácil de seguir del punto fijo en cálculo lambda tipado se puede leer en Dowek y Lévy [2011].

5.2.2. La semántica denotacional del lambda cálculo extendido

En general, en los lenguajes funcionales buscamos reducir la distancia que separa la noción de programa de la de función. Es decir, se busca reducir la distancia entre un programa y su semántica denotacional.
El primer paso es definir una categoría cartesiana cerrada de manera tal de dar una interpretación de los tipos en objetos de dicha categoría, y los términos tipados interpretarlos en flechas de la categoría. La idea es que si $\Gamma = x_1 : A_1, \ldots, x_n : A_n$, queremos definir

$$[\![\Gamma \vdash t : B]\!] = [\![A_1]\!] \times \cdots \times [\![A_n]\!] \xrightarrow{t} [\![B]\!]$$

Para interpretar el cálculo tipado visto en la Sección 4.2, utilizaremos la categoría Set vista en la sección anterior.

Interpretación de los tipos. A cada tipo le asociamos un objeto de Set:

$$\begin{aligned} [\![\top]\!] &= 1 \\ [\![\bot]\!] &= \emptyset \\ [\![A \Rightarrow B]\!] &= [[\![A]\!], [\![B]\!]] \\ [\![A \wedge B]\!] &= [\![A]\!] \times [\![B]\!] \\ [\![A \vee B]\!] &= [\![A]\!] + [\![B]\!] \end{aligned}$$

El mapa $[\![\cdot]\!]$ también lo definimos para contextos de la siguiente manera:

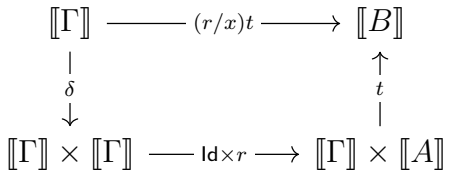
$$\begin{aligned} [\![\emptyset]\!] &:= 1 \\ [\![\Gamma, x : A]\!] &:= [\![\Gamma]\!] \times [\![A]\!] \end{aligned}$$

Interpretación de los términos. A cada secuente $\Gamma \vdash t : A$ le asociamos una flecha de la categoría Set. Dado que las reglas de tipado dadas en la Definición 4.42 son syntax directed (eso es, a cada secuente le corresponde una única derivación), definimos la relación para cada regla en lugar de para cada secuente. Algunas flechas utilizadas se encuentran definidas en la Práctica 3 (y sus propiedades son demostradas allí).

- $[\![\dfrac{}{\Gamma, x : A \vdash x : A}\ \mathrm{ax}]\!] = [\![\Gamma]\!] \times [\![A]\!] \xrightarrow{\pi_{[\![A]\!]}} [\![A]\!] \xrightarrow{\mathsf{Id}} [\![A]\!]$
- $[\![\dfrac{}{\Gamma \vdash \star : \top}\ \top_i]\!] = [\![\Gamma]\!] \xrightarrow{\lambda^{-1}} 1 \times [\![\Gamma]\!] \xrightarrow{\pi_1} 1$
- $[\![\dfrac{\Gamma \vdash t : \top \quad \Gamma \vdash r : A}{\Gamma \vdash t; r : A}\ \top_e]\!] = [\![\Gamma]\!] \xrightarrow{\delta} [\![\Gamma]\!] \times [\![\Gamma]\!] \xrightarrow{t \times r} 1 \times [\![A]\!] \xrightarrow{\lambda} [\![A]\!]$
- $[\![\dfrac{\Gamma \vdash t : \bot}{\Gamma \vdash \mathsf{err}(t) : C}\ \bot_e]\!] = [\![\Gamma]\!] \xrightarrow{t} \emptyset \xrightarrow{\emptyset} [\![C]\!]$
- $[\![\dfrac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \Rightarrow B}\ \Rightarrow_i]\!] = [\![\Gamma]\!] \xrightarrow{\eta^{[\![A]\!]}} [[\![A]\!], [\![\Gamma]\!] \times [\![A]\!]] \xrightarrow{[[\![A]\!], t]} [[\![A]\!], [\![B]\!]]$
- $[\![\dfrac{\Gamma \vdash t : A \Rightarrow B \quad \Gamma \vdash r : A}{\Gamma \vdash tr : B}\ \Rightarrow_e]\!] = [\![\Gamma]\!] \xrightarrow{\delta} [\![\Gamma]\!] \times [\![\Gamma]\!] \xrightarrow{t \times r} [[\![A]\!], [\![B]\!]] \times [\![A]\!] \xrightarrow{\varepsilon} [\![B]\!]$
- $[\![\dfrac{\Gamma \vdash t : A \quad \Gamma \vdash r : B}{\Gamma \vdash \langle t, r \rangle : A \wedge B}\ \wedge_i]\!] = [\![\Gamma]\!] \xrightarrow{\delta} [\![\Gamma]\!] \times [\![\Gamma]\!] \xrightarrow{t \times r} [\![A]\!] \times [\![B]\!]$
- $[\![\dfrac{\Gamma \vdash t : A \wedge B}{\Gamma \vdash \pi_1 t : A}\ \wedge_{e_1}]\!] = [\![\Gamma]\!] \xrightarrow{t} [\![A]\!] \times [\![B]\!] \xrightarrow{\pi_1} [\![A]\!]$
- $[\![\dfrac{\Gamma \vdash t : A \wedge B}{\Gamma \vdash \pi_2 t : B}\ \wedge_{e_2}]\!] = [\![\Gamma]\!] \xrightarrow{t} [\![A]\!] \times [\![B]\!] \xrightarrow{\pi_2} [\![B]\!]$
- $[\![\dfrac{\Gamma \vdash t : A}{\Gamma \vdash \mathsf{inl}(t) : A \vee B}\ \vee_{i_1}]\!] = [\![\Gamma]\!] \xrightarrow{t} [\![A]\!] \xrightarrow{i_1} [\![A]\!] + [\![B]\!]$
- $[\![\dfrac{\Gamma \vdash t : B}{\Gamma \vdash \mathsf{inr}(t) : A \vee B}\ \vee_{i_2}]\!] = [\![\Gamma]\!] \xrightarrow{t} [\![B]\!] \xrightarrow{i_2} [\![A]\!] + [\![B]\!]$
- $[\![\dfrac{\Gamma \vdash t : A \vee B \quad \Gamma, x : A \vdash r : C \quad \Gamma, y : B \vdash s : C}{\Gamma \vdash \mathsf{match}(t, x.r, y.s) : C}\ \vee_e]\!]$
  $= [\![\Gamma]\!] \xrightarrow{\delta} [\![\Gamma]\!] \times [\![\Gamma]\!] \xrightarrow{\mathsf{Id} \times t} [\![\Gamma]\!] \times ([\![A]\!] + [\![B]\!]) \xrightarrow{d} ([\![\Gamma]\!] \times [\![A]\!]) + ([\![\Gamma]\!] \times [\![B]\!]) \xrightarrow{[r,s]} [\![C]\!]$

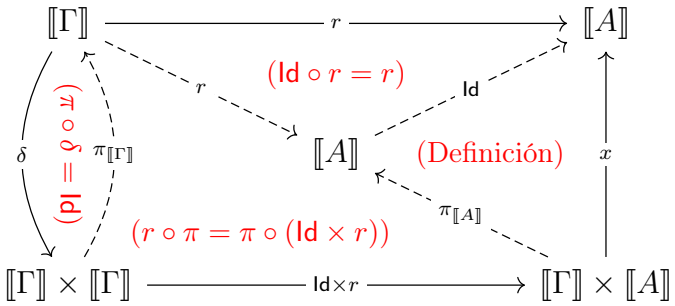
La correctitud de esta definición se establece por medio del teorema de "Soundness" (Teorema 5.37), que dice que si un término reduce a otro, ambos tienen la misma interpretación (es decir, denotan la misma flecha de la categoría). Para probar este teorema, primero debemos probar la propiedad de substitución (Lema 5.36).

Lema 5.36 (Substitución) Si $\Gamma, x : A \vdash t : B$ y $\Gamma \vdash r : A$, entonces el siguiente diagrama conmuta (modulo permutaciones).

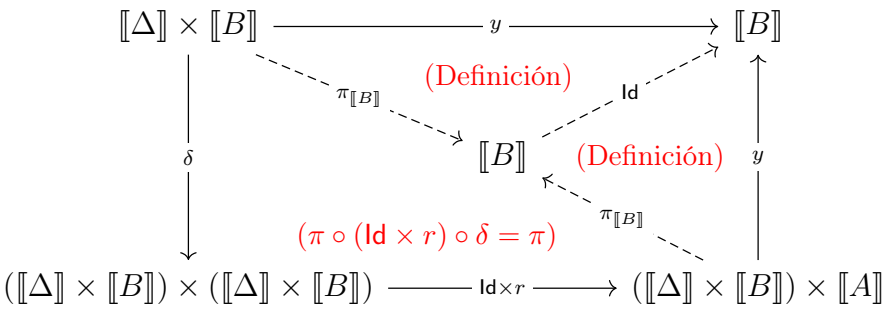


Demostración. Procedemos por inducción en t. Reescribiremos el diagrama para cada caso, agregando en líneas de puntos ponemos flechas extras que simplifiquen el diagrama, y en rojo la justificación de cada subdiagrama.

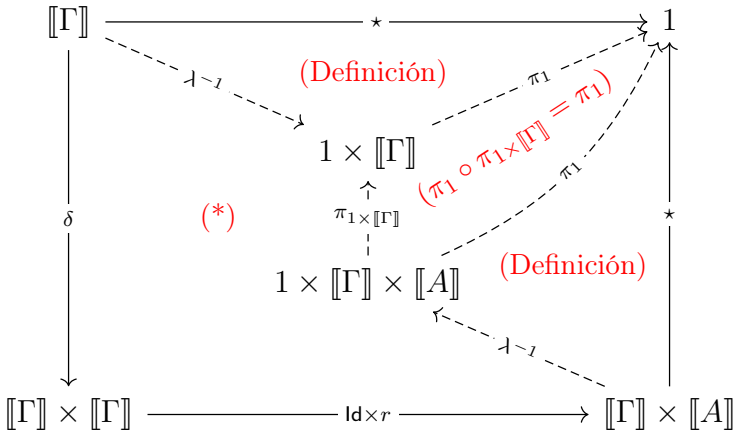
- Sea $t = x$, entonces $(r/x)t = r$, $A = B$ y $\llbracket \Gamma, x : A \vdash x : A \rrbracket = \mathsf{Id} \circ \pi_{\llbracket A \rrbracket}$. El diagrama queda como sigue



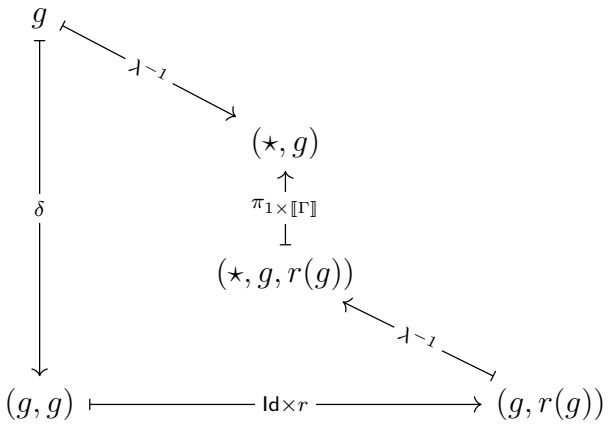
- Sea $t = y \neq x$, entonces $(r/x)t = y$, y $\Gamma = \Delta, y : B$. El diagrama queda como sigue.



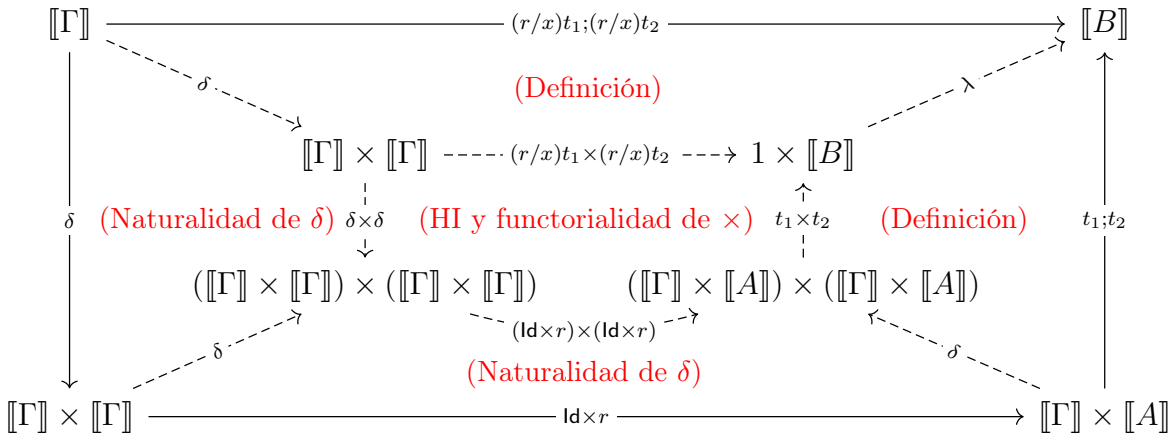
- Sea $t = \star$, entonces $(r/x)t = \star$ y $B = \top$ (y $\llbracket \top \rrbracket = 1$). El diagrama queda como sigue.



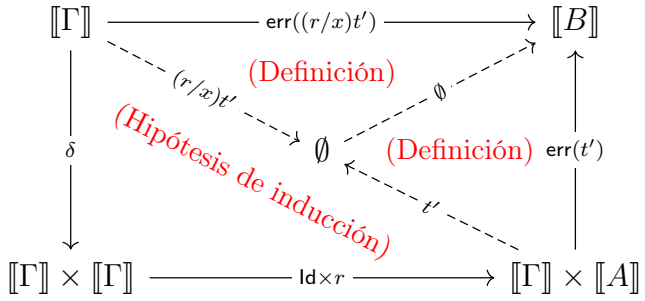
donde el diagrama (*) lo justificamos analizando los mapas. Sea $g \in [\![\Gamma]\!]$:



- Sea $t = t_1; t_2$, entonces $(r/x)t = (r/x)t_1; (r/x)t_2$ con $\Gamma, x : A \vdash t_1 : \top$ y $\Gamma, x : A \vdash t_2 : B$. El diagrama queda como sigue.



- Sea $t = \mathsf{err}(t')$, entonces $(r/x)t = \mathsf{err}((r/x)t')$, con $\Gamma, x : A \vdash t' : \bot$. El diagrama queda como sigue.

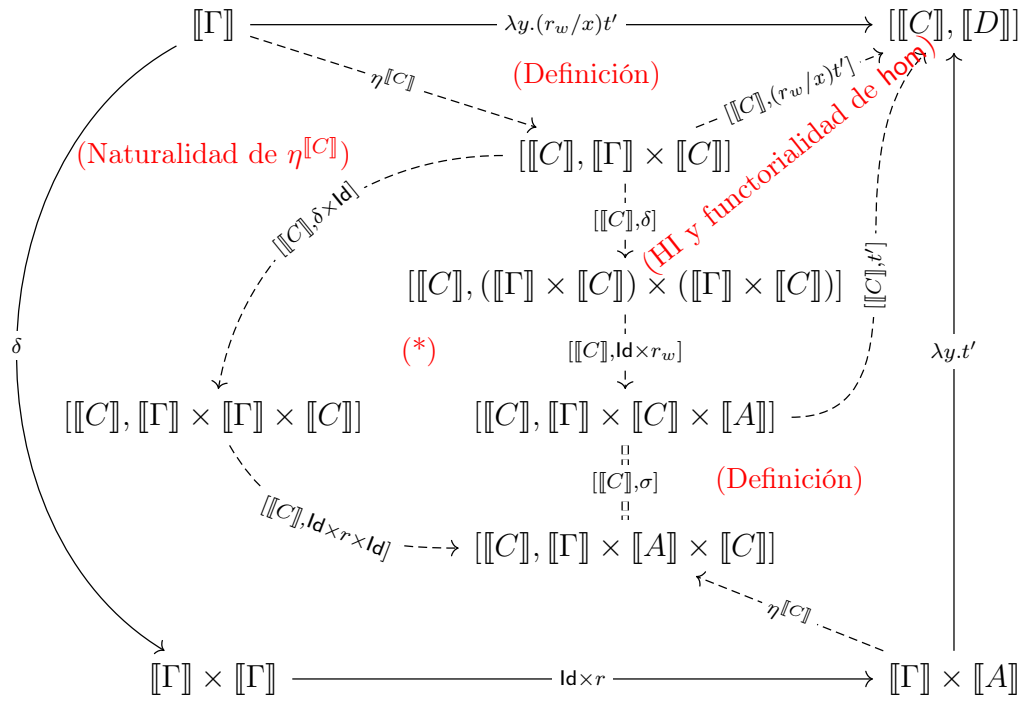


- Sea $t = \lambda y.t'$, entinces $(r/x)t = \lambda y.(r/x)t'$, con $B = C \Rightarrow D$, $\Gamma, x : A, y : C \vdash t' : D$. Para la hipótesis de inducción se debe utilizar la propiedad de weakening, mostrando que como $\Gamma \vdash r : A$, también tenemos $\Gamma, y : C \vdash r : A$, y entonces la hipótesis de

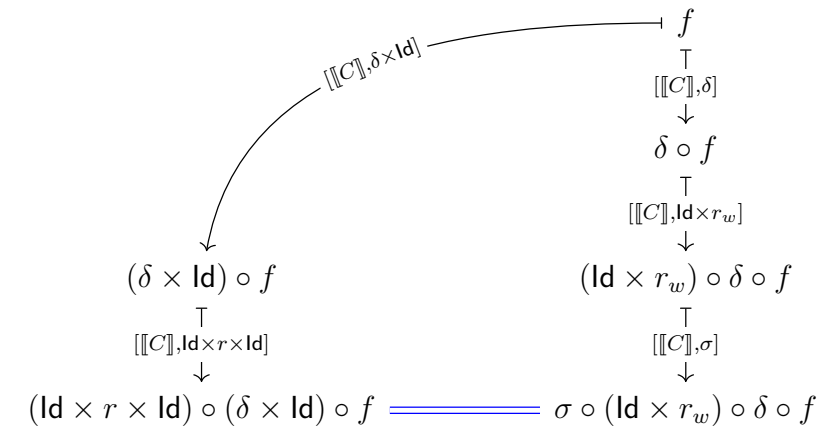
inducción dice lo siguiente, donde llamamos r_w a la flecha $[\![\Gamma, y : C \vdash r : A]\!]$ para diferenciarla de $r = [\![\Gamma \vdash r : A]\!]$.

$$\begin{array}{ccc} [\![\Gamma]\!] \times [\![C]\!] & \xrightarrow{\quad (r_w/x)t' \quad} & [\![D]\!] \\ \downarrow{\scriptstyle \delta} & & \uparrow{\scriptstyle t'} \\ ([\![\Gamma]\!] \times [\![C]\!]) \times ([\![\Gamma]\!] \times [\![C]\!]) & \xrightarrow{\mathsf{Id}\times r_w} & ([\![\Gamma]\!] \times [\![C]\!]) \times [\![A]\!] \end{array}$$

Por lo tanto, el diagrama queda como sigue.



donde el diagrama (*) lo justificamos analizando los mapas. Sea $f \in [[\![C]\!], [\![\Gamma]\!] \times [\![C]\!]]$.

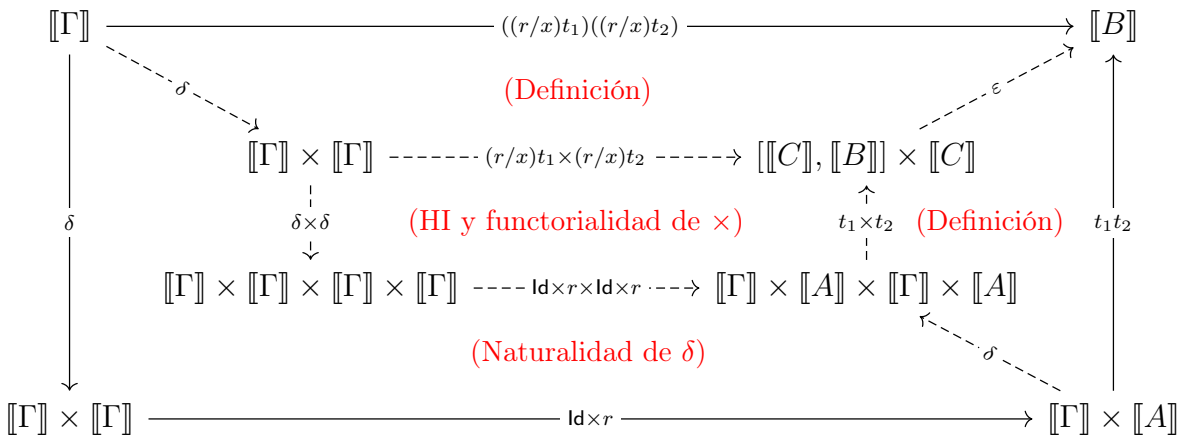


Donde la igualdad final es justificada como sigue. Sea $c \in [\![C]\!]$, $f(c) = (g, c')$ y

$r(g) = a$. Finalmente, notar que $r_w = r \circ \pi_{[\![\Gamma]\!]}$, entonces

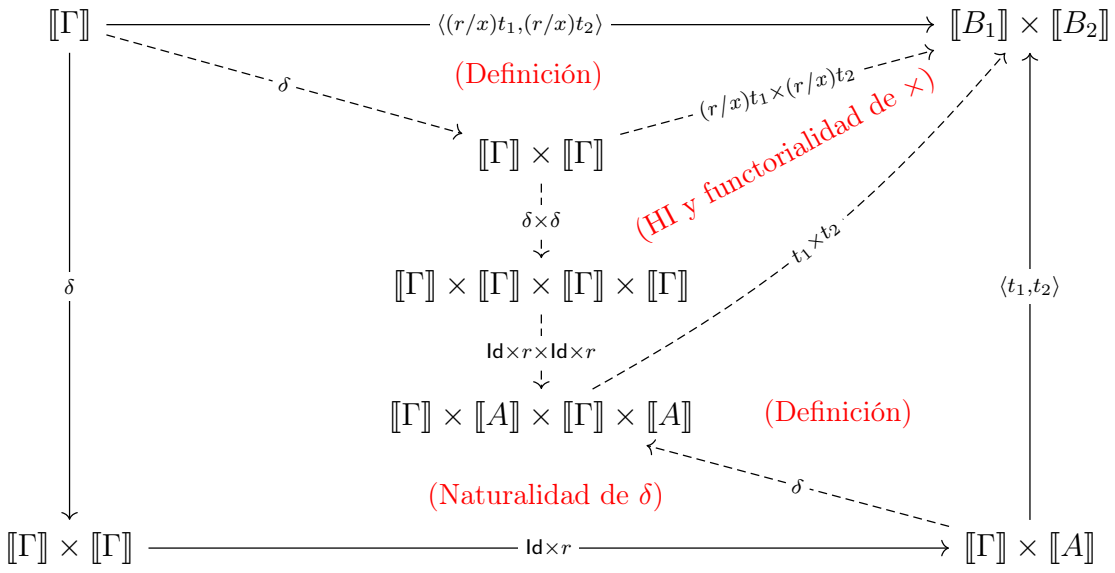
$$\begin{aligned}((\mathsf{Id} \times r \times \mathsf{Id}) \circ (\delta \times \mathsf{Id}) \circ f)(c) &= ((\mathsf{Id} \times r \times \mathsf{Id}) \circ (\delta \times \mathsf{Id}))(f(c)) \\ &= ((\mathsf{Id} \times r \times \mathsf{Id}) \circ (\delta \times \mathsf{Id}))(g, c') \\ &= (\mathsf{Id} \times r \times \mathsf{Id})(g, g, c') \\ &= (g, a, c') \\ &= \sigma(g, c', a) \\ &= (\sigma \circ (\mathsf{Id} \times r))(g, c', g) \\ &= (\sigma \circ (\mathsf{Id} \times (r \circ \pi_\Gamma)))(g, c', g, c') \\ &= (\sigma \circ (\mathsf{Id} \times r_w))(g, c', g, c') \\ &= (\sigma \circ (\mathsf{Id} \times r_w) \circ \delta)(g, c') \\ &= (\sigma \circ (\mathsf{Id} \times r_w) \circ \delta)(f(c)) \\ &= (\sigma \circ (\mathsf{Id} \times r_w) \circ \delta \circ f)(c)\end{aligned}$$

- Sea $t = t_1 t_2$, entonces $(r/x)t = ((r/x)t_1)((r/x)t_2)$, con $\Gamma, x : A \vdash t_1 : C \Rightarrow B$ y $\Gamma, x : A \vdash t_2 : C$. El diagrama queda como sigue.

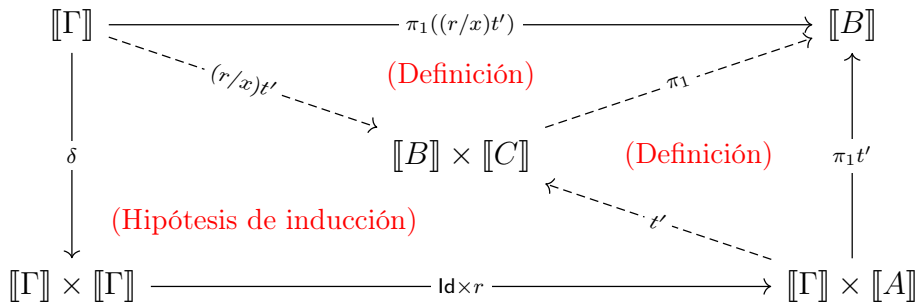


- Sea $t = \langle t_1, t_2 \rangle$, entonces $(r/x)t = \langle (r/x)t_1, (r/x)t_2 \rangle$ con $B = B_1 \wedge B_2$, $\Gamma, x : A \vdash$

$t_1 : B_1$ y $\Gamma, x : A \vdash t_2 : B_2$. El diagrama queda como sigue.

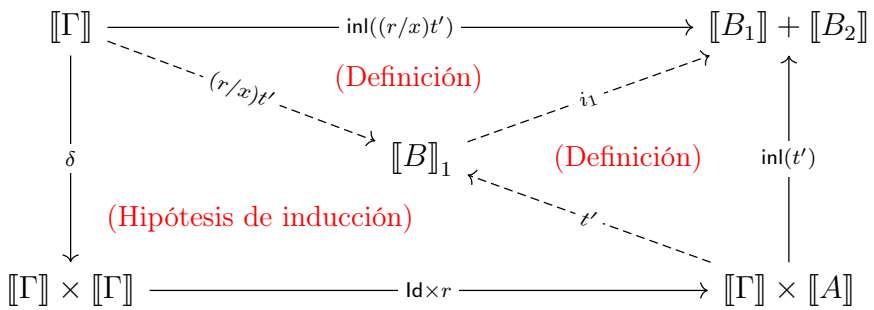


- Sea $t = \pi_1 t'$, entonces $(r/x)t = \pi_1((r/x)t')$, con $\Gamma, x : A \vdash t' : B \wedge C$. El diagrama queda como sigue.



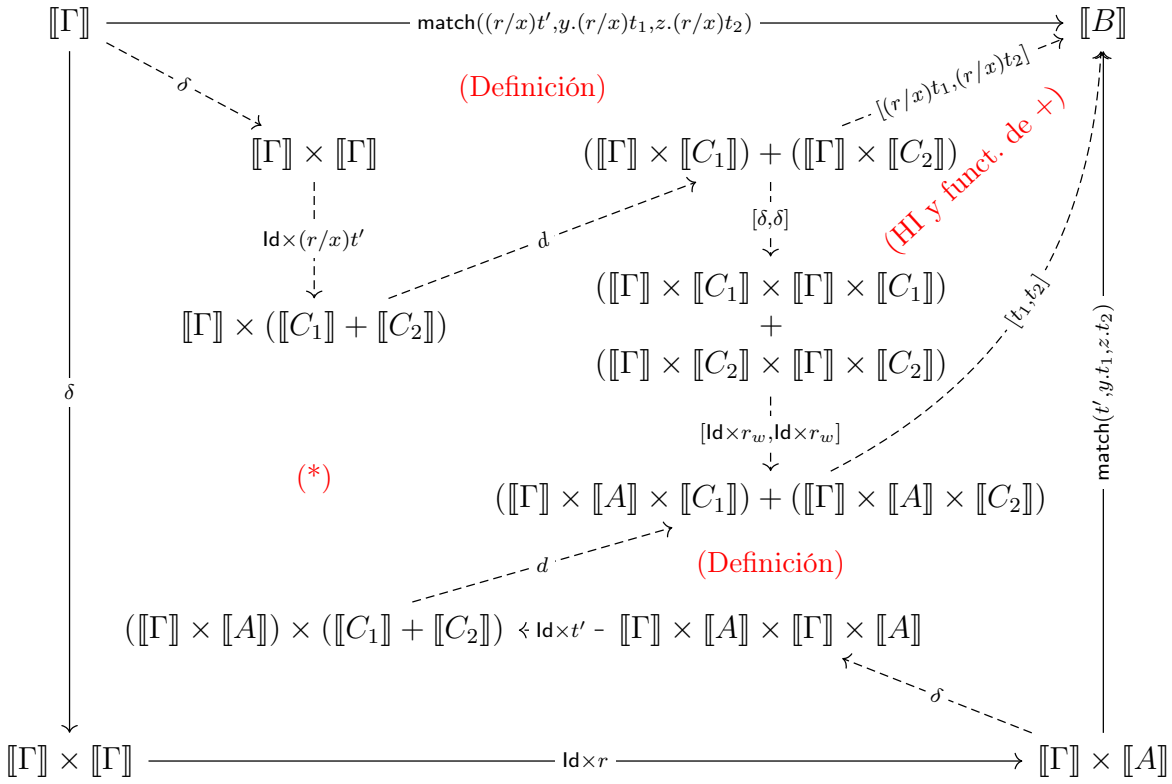
- Sea $t = \pi_2 t'$. Este caso es análogo al anterior.

- Sea $t = \mathsf{inl}(t')$, entonces $(r/x)t = \mathsf{inl}((r/x)t')$, con $B = B_1 \vee B_2$ y $\Gamma, x : A \vdash t' : B_1$. El diagrama queda como sigue.

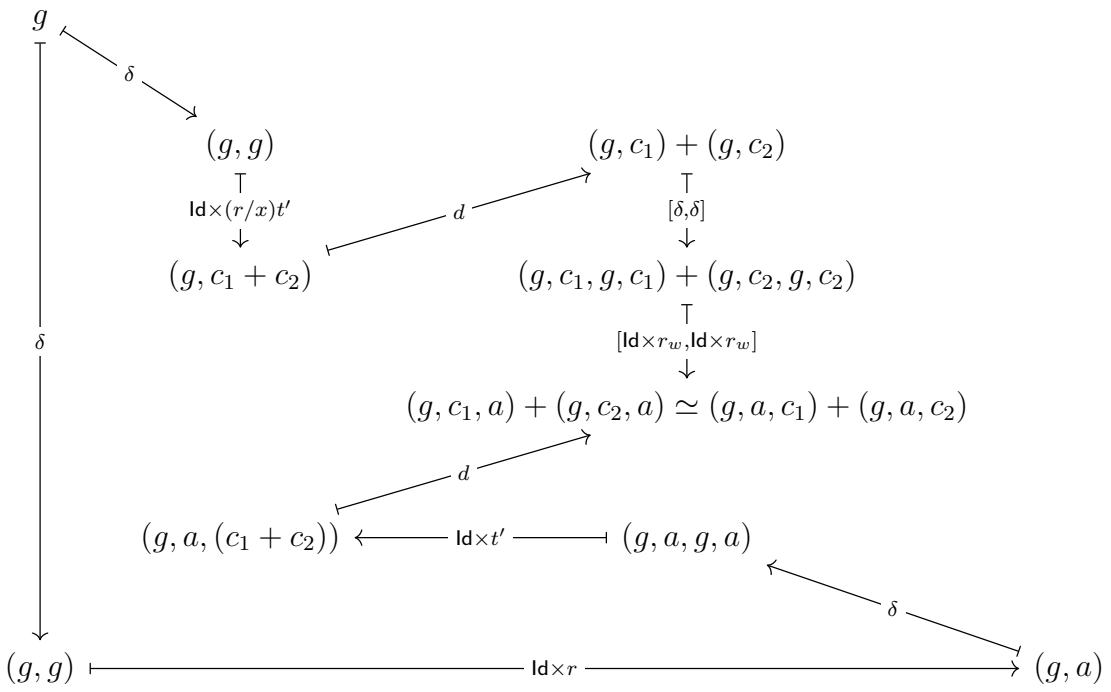


- Sea $t = \mathsf{inr}(t')$. Este caso es análogo al anterior.

- Sea $t = \mathsf{match}(t', y.t_1, z.t_2)$, entonces $(r/x)t = \mathsf{match}((r/x)t', y.(r/x)t_1, z.(r/x)t_2)$, con $\Gamma, x : A \vdash t' : C_1 \vee C_2$, $\Gamma, x : A, y : C_1 \vdash t_1 : B$, y $\Gamma, x : A, z : C_1 \vdash t_2 : B$. El diagrama queda como sigue.



donde el diagrama (*) se justifica de la siguiente manera. Sea $g \in \llbracket\Gamma\rrbracket$,$r(g) = a$, $((r/x)t')(g) = c_1 + c_2$, y, por hipótesis de inducción, $t'(g, a) = c_1 + c_2$. Entonces tenemos,



□

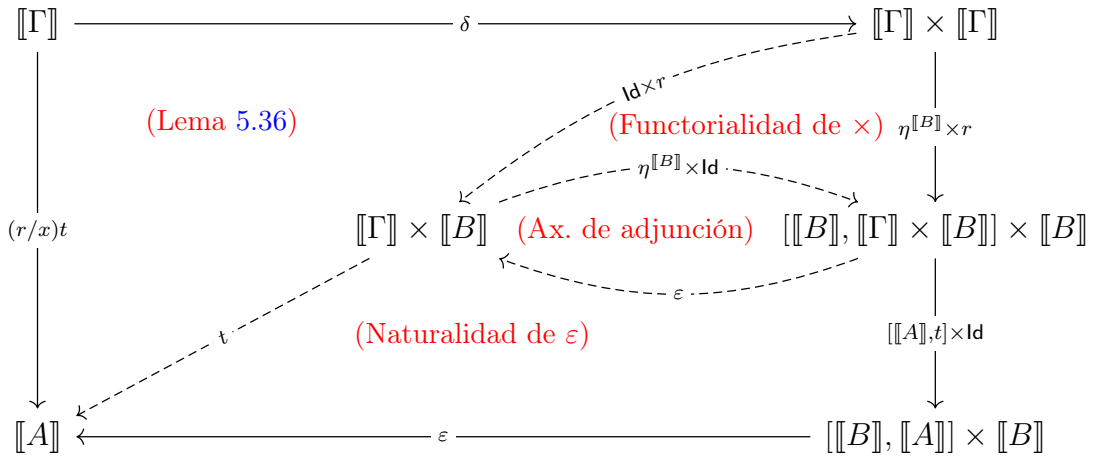
Teorema 5.37 (Soundness) Si $\Gamma \vdash t : A$ y $t \to r$, entonces $[\![\Gamma \vdash t : A]\!] = [\![\Gamma \vdash r : A]\!]$.

Demostración. Inducción sobre la relación $\longrightarrow$. Damos uno de los casos base y un caso inductivo a modo de ejemplo:

- Caso base $(\lambda x.t)r \longrightarrow (r/x)t$.

  Consideramos $\left[\!\!\left[\dfrac{\dfrac{\Gamma, x : B \vdash t : A}{\Gamma \vdash \lambda x.t : B \Rightarrow A} \quad \Gamma \vdash r : B}{\Gamma \vdash (\lambda x.t)r : A} \right]\!\!\right]$ y $[\![\Gamma \vdash (r/x)t : A]\!]$

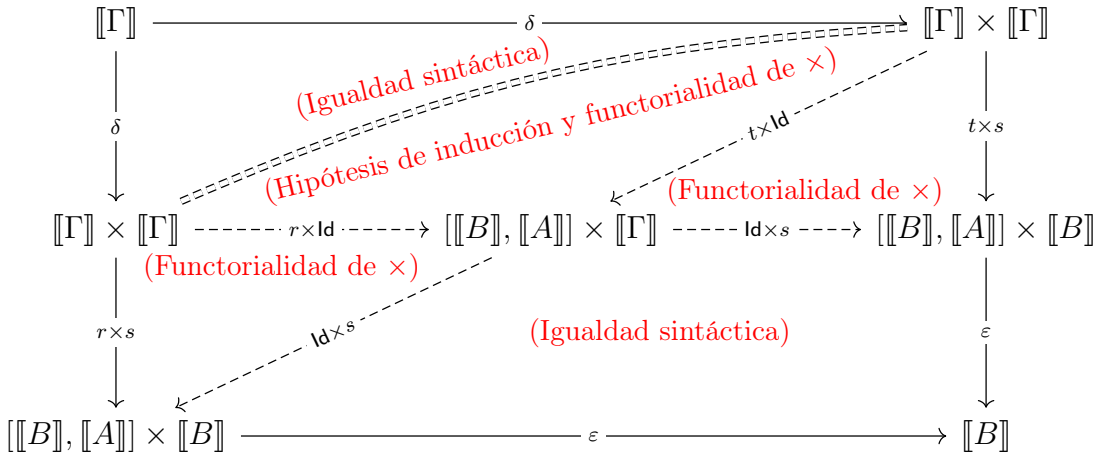
  El diagrama conmutativo correspondiente es el siguiente.



- Caso inductivo $\dfrac{t \longrightarrow r}{ts \longrightarrow rs}$

  Consideramos $\left[\!\!\left[\dfrac{\Gamma \vdash t : B \Rightarrow A \quad \Gamma \vdash s : B}{\Gamma \vdash ts : A} \right]\!\!\right]$ y $\left[\!\!\left[\dfrac{\Gamma \vdash r : B \Rightarrow A \quad \Gamma \vdash s : B}{\Gamma \vdash rs : A} \right]\!\!\right]$

  El diagrama conmutativo correspondiente es el siguiente.



Ejercicio 5.38. Resolver todos los casos restantes.

□

Capítulo 6

Rapid(ísim)a descripción de la lógica lineal (MALL)

Esta sección es una adaptación libre de la Sección 3.1 del artículo de Di Cosmo y Miller [2016]. Para más detalles, se sugiere recurrir a dicha fuente.

6.1. Introducción

La lógica lineal fue introducida por Girard [1987]. Aquí daremos una presentación por medio de cálculo de secuentes, ya que es muy similar a las reglas de tipado que hemos visto en las secciones anteriores.
La idea principal a retener es que la lógica lineal es una lógica de recursos: la fórmula $A \Rightarrow B$ normalmente se entiende como "Si me das A, te devuelvo B", pero, en la práctica, significa más bien "Si me das tantas A como necesite, te devuelvo B". Por ejemplo, el término

$$\lambda x.x; x$$

tiene tipo $\top \Rightarrow \top$, pero para calcular $x; x$, se necesitan dos copias de x. Es decir, el recurso (x), fue duplicado para poder calcular el resultado. Si, por ejemplo, el recurso fuese un programa complejo que devuelve $\star$, entonces duplicar el recurso tiene un costo. En lógica lineal no podemos duplicar recursos. Así, el tipo $\top \multimap \top$ significa: "Si me das un $\top$, te devuelvo un $\top$ usándolo exactamente una vez".
Ésta lógica nos será de utilidad para definir cálculos cuánticos, ya que el teorema de no clonado (Teorema 1.27) nos impide clonar recursos cuánticos.
De la misma manera, la función $\lambda x.\star$, que descarta su argumento, no podríamos decir que tiene tipo $\top \multimap \top$, ya que no utiliza una vez su argumento.

6.2. Cálculo de secuentes para MELL

Los conectivos de la lógica lineal se dividen en multiplicativos (que no permiten duplicar recursos) y aditivos (que lo permiten), y los conectivos clásicos tienen su paralelo en ambos:

Clásico		Multiplicativo		Aditivo	
$\wedge$	(conjunción)	$\otimes$	(tensor)	$\&$	(with)
$\top$	(verdadero)	$\mathbf{1}$	(uno)	$\top$	(top)
$\vee$	(disjunción)	$\parr$	(par)	$\oplus$	(oplus)
$\bot$	(falso)	$\bot$	(bottom)	$\mathbf{0}$	(cero)

Implicación lineal: $A \multimap B := \neg A \parr B$
A continuación se detallan las reglas en el formato $\Delta \vdash \Gamma$, que significa que la conjunción (multiplicativa) de las fórmulas en Δ, implican la disjunción (multiplicativa) de las fórmulas en Γ.

Gramática

$$A := p \mid \neg A \mid A \otimes A \mid A \parr A \mid A \,\&\, A \mid A \oplus A \mid \mathbf{1} \mid \bot \mid \top \mid \mathbf{0}$$

Donde p representa una fórmula atómica.

Reglas de indentidad y negación

$$\frac{}{A \vdash A}\ ax \qquad \frac{\Delta \vdash B, \Gamma \quad \Delta', B \vdash \Gamma'}{\Delta, \Delta' \vdash \Gamma, \Gamma'}\ cut \qquad \frac{\Delta \vdash A, \Gamma}{\Delta, \neg A \vdash \Gamma}\ \neg_l \qquad \frac{\Delta, A \vdash \Gamma}{\Delta \vdash \neg A, \Gamma}\ \neg_r$$

Reglas multiplicativas

$$\frac{\Delta \vdash \Gamma}{\Delta, \mathbf{1} \vdash \Gamma}\ \mathbf{1}_l \qquad \frac{}{\vdash \mathbf{1}}\ \mathbf{1}_r \qquad \frac{\Delta, A, B \vdash \Gamma}{\Delta, A \otimes B \vdash \Gamma}\ \otimes_l \qquad \frac{\Delta \vdash A, \Gamma \quad \Delta' \vdash B, \Gamma'}{\Delta, \Delta' \vdash A \otimes B, \Gamma, \Gamma'}\ \otimes_r$$

$$\frac{}{\bot \vdash}\ \bot_l \qquad \frac{\Delta \vdash \Gamma}{\Delta \vdash \bot, \Gamma}\ \bot_r \qquad \frac{\Delta, A \vdash \Gamma \quad \Delta', B \vdash \Gamma'}{\Delta, \Delta', A \parr B \vdash \Gamma, \Gamma'}\ \parr_l \qquad \frac{\Delta \vdash A, B, \Gamma}{\Delta \vdash A \parr B, \Gamma}\ \parr_r$$

Reglas aditivas

$$\frac{}{\Delta, \mathbf{0} \vdash \Gamma}\ \mathbf{0}_l \qquad \frac{\Delta, A \vdash \Gamma}{\Delta, A \,\&\, B \vdash \Gamma}\ \&_{l1} \qquad \frac{\Delta, B \vdash \Gamma}{\Delta, A \,\&\, B \vdash \Gamma}\ \&_{l2} \qquad \frac{\Delta \vdash A, \Gamma \quad \Delta \vdash B, \Gamma}{\Delta \vdash A \,\&\, B, \Gamma}\ \&_r$$

$$\frac{}{\Delta \vdash \top, \Gamma}\ \top_r \qquad \frac{\Delta, A \vdash \Gamma \quad \Delta, B \vdash \Gamma}{\Delta, A \oplus B \vdash \Gamma}\ \oplus_l \qquad \frac{\Delta \vdash A, \Gamma}{\Delta \vdash A \oplus B, \Gamma}\ \oplus_{r1} \qquad \frac{\Delta \vdash B, \Gamma}{\Delta \vdash A \oplus B, \Gamma}\ \oplus_{r2}$$

6.3. Un ejemplo simple de sistema de tipos lineal

Supongamos que queremos redefinir lambda cálculo extendido de la Sección 4.2, de manera que los tipos usen la lógica lineal. Aquí tenemos algunas deciciones a hacer: ¿usamos aditivos o multiplicativos? O una mezcla de ambos? Un resultado conocido es que en lógica intuicionista, no existe la disjunción multiplicativa ni el falso multiplicativo, y no existe la implicación es siempre multiplicativa, por lo que ahí no hay elección posible: la disjunción y el falso deben ser el aditivos ($\oplus$ y $\mathbf{0}$ respectivamente), y la implicación será $\multimap$. En cambio es posible tener una conjunción multiplicativa y una aditiva. Si queremos

seguir utilizando el mismo cálculo, deberemos elegir la aditiva, ya que las proyecciones π_1 y π_2 que tenemos en el cálculo no pueden ser multiplicativas. Por lo tanto, utilizaremos & para la conjunción. Finalmente, es posible tener tanto el verdadero aditivo como el multiplicativo. Elegiremos el multiplicativo.
Con todo esto, definimos el siguiente sistema de tipos (ver la diferencia con la relación de tipado de la Definición 4.42). Es posible comprobar (y un buen ejercicio), que cada una de las reglas es derivable en el cálculo de secuentes dado en la sección anterior.

$$\frac{}{x : A \vdash x : A}\ \text{ax} \qquad \frac{}{\vdash \star : \mathbf{1}}\ \mathbf{1}_i \qquad \frac{\Gamma \vdash t : \mathbf{1} \quad \Delta \vdash r : A}{\Gamma, \Delta \vdash t; r : A}\ \mathbf{1}_e \qquad \frac{\Gamma \vdash t : \mathbf{0}}{\Gamma \vdash \mathsf{err}(t) : C}\ \mathbf{0}_e$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \multimap B}\ \multimap_i \qquad \frac{\Gamma \vdash t : A \multimap B \quad \Delta \vdash r : A}{\Gamma, \Delta \vdash tr : B}\ \multimap_e$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash r : B}{\Gamma \vdash \langle t, r \rangle : A \& B}\ \&_i \qquad \frac{\Gamma \vdash t : A \& B}{\Gamma \vdash \pi_1 t : A}\ \&_{e_1} \qquad \frac{\Gamma \vdash t : A \& B}{\Gamma \vdash \pi_2 t : B}\ \&_{e_2}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathsf{inl}(t) : A \oplus B}\ \oplus_{i_1} \qquad \frac{\Gamma \vdash t : B}{\Gamma \vdash \mathsf{inr}(t) : A \oplus B}\ \oplus_{i_2}$$

$$\frac{\Gamma \vdash t : A \vee B \quad \Delta, x : A \vdash r : C \quad \Delta, y : B \vdash s : C}{\Gamma, \Delta \vdash \mathsf{match}(t, x.r, y.s) : C}\ \vee_e$$

Donde $\Gamma \cap \Delta = \emptyset$.
Podríamos agregar términos para los conectivos faltantes. Por ejemplo, podemos agregar $t \otimes r$ para la conjunción multiplicativa, y un término para la eliminación que podría ser $\mathsf{let}\ t = x \otimes y\ \mathsf{in}\ u$, con la regla de reducción siguiente:

$$\mathsf{let}\ t \otimes r = x \otimes y\ \mathsf{in}\ u \longrightarrow (t/x, r/y)u$$

y sus reglas de tipado:

$$\frac{\Gamma \vdash t : A \quad \Delta \vdash r : B}{\Gamma, \Delta \vdash t \otimes r : A \otimes B}\ \otimes_i \qquad \frac{\Gamma \vdash t : A \otimes B \quad \Delta, x : A, y : B \vdash u : C}{\Gamma, \Delta \vdash \mathsf{let}\ t = x \otimes y\ \mathsf{in}\ u : C}\ \otimes_e$$

Ejemplo 6.1 Dado que la conjunción & es aditiva, puede duplicar su argumento:

$$\frac{\dfrac{\dfrac{}{x : \mathbf{1} \vdash x : \mathbf{1}}\ \mathbf{1}_i \quad \dfrac{}{x : \mathbf{1} \vdash x : \mathbf{1}}\ \mathbf{1}_i}{x : \mathbf{1} \vdash \langle x, x \rangle : \mathbf{1} \& \mathbf{1}}\ \&_i}{\vdash \lambda x.\langle x, x \rangle : \mathbf{1} \multimap \mathbf{1} \& \mathbf{1}}\ \multimap_i$$

En cambio, con la conjunción multiplicativa $\otimes$, no es posible, $\lambda x.x \otimes x$ no tiene un tipo, ya que $x : \mathbf{1} \nvdash x \otimes x : \mathbf{1} \otimes \mathbf{1}$.

Ejercicio 6.2. Mostrar que las reglas dadas son lógicamente derivables en cálculo de secuentes.

Ejercicio 6.3. Completar el lenguaje con términos que se correspondan con el conectivo que dejamos afuera: el verdadero aditivo. Dar sus reglas de tipado.

Parte III

Hacia un Curry-Howard en Computación Cuántica

Capítulo 7

Extensiones cuánticas al lambda cálculo

Capítulo 8

Un nuevo conectivo de Deducción Natural

Bibliografía citada

Charles Bennett y Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. En *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, págs. 175–179. 1984.

Charles Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, y William Wootters. Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels. *Physical Review Letters*, 70(13):1895–1899, 1993.

Charles Bennett y Stephen Wiesner. Communication via one- and two-particle operators on Einstein–Podolsky–Rosen states. *Physical Review Letters*, 69(20):2881–2884, 1992.

Garret Birkhoff y John von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37(4):823–843, 1936.

Julian Brown. *The Quest for the Quantum Computer*. Touchstone, 2001.

David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

David Deutsch y Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 439(1907):553–558, 1992.

Roberto Di Cosmo y Dale Miller. Linear logic. *The Stanford Encyclopedia of Philosophy*, Winter Edition, 2016. Edward N. Zalta (ed.). https://plato.stanford.edu/archives/win2016/entries/logic-linear.

Paul A. M. Dirac. A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35(03):416–418, 1939.

Gilles Dowek y Jean-Jacques Lévy. *Introduction to the theory of programming languages*. Undergraduate topics in computer science. Springer, 2011.

Albert Einstein, Boris Podolsky, y Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, 44(10):777–780, 1935.

Jean-Yves Girard. Linear logic. *Theoretical Compututer Science*, 50:1–102, 1987.

Lov K. Grover. A fast quantum mechanical algorithm for database search. En *Proceedings of the 28th Annual ACM Symposium on Theory of computing*, STOC-96, págs. 212–219. ACM, 1996.

John Preskill. Quantum computing: pro and con. *Proceedings of the Royal Society of London A*, 454:469–486, 1998.

Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

Gilbert S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Transactions of the American Institute of Electrical Engineers*, XLV:295–301, 1926.

William K. Wootters y Wojciech .H. Zurek. A single quantum cannot be cloned. *Nature*, 299:802–803, 1982.