

PROPUESTA DE TESINA  
DE LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN  
Compilación de  $\lambda$ -cálculo con matrices densidad  
en la máquina cuántica IBM-Q

Martín Villagra  
V-2719/7  
mvillagra0@gmail.com

11 de diciembre de 2020

Director: Alejandro Díaz-Caro  
Investigador Adjunto en el Instituto de Ciencias de la Computación (UBA/CONICET)  
Profesor Adjunto en la Universidad Nacional de Quilmes  
adiazcaro@icc.fcen.uba.ar

Codirector: Pablo E. Martínez López  
Profesor Titular en la Universidad Nacional de Quilmes  
fidel@unq.edu.ar

## Introducción y estado del arte

La computación cuántica se puede entender como un modelo de cómputo novedoso, basado en las propiedades de la física, o como un formalismo matemático para describir la física cuántica. Desde el punto de vista de la tecnología, las computadoras cuánticas son dispositivos programables, y por tanto, existen diversos lenguajes de programación para las mismas. El más común, se basa en circuitos cuánticos (el análogo a los circuitos electrónicos). Desde el punto de vista más teórico, el estudio de fundamentos de lenguajes de programación para las computadoras cuánticas, no sólo tiene por objetivo el desarrollo de dichos lenguajes, sino también el estudio de propiedades de la física cuántica, puramente desde el formalismo matemático, en particular dada la conexión entre los fundamentos de lenguajes (y la teoría de tipos), con la lógica.

En esta línea, existen diversas extensiones al cálculo lambda, y diversos sistemas de tipos, para lidiar con diferentes características de la computación cuántica. En particular, existen dos paradigmas muy diferenciados.

Por un lado, está el paradigma de “control clásico” [Sel04], el cual introduce en el lenguaje la posibilidad de describir circuitos cuánticos. Dichas descripciones son clásicas: las compuertas cuánticas se describen como cajas negras, y el flujo de control que dice qué compuerta aplicar a qué qubit, es clásico, en el sentido de que no existe superposición de flujos de control, o de programas. Dicho paradigma está justificado por el hecho de que las computadoras cuánticas se

visualizan como dispositivos anexos a las computadoras clásicas, y son estas últimas las que instruyen a las primeras sobre las operaciones a realizar. Así, para programar, por ejemplo, la computadora cuántica IBM-Q<sup>1</sup>, con su SDK Qiskit<sup>2</sup> (un conjunto de librerías de Python), se desarrolla un programa clásico en Python, con algunas instrucciones extra para instruir a la computadora cuántica qué circuito cuántico debe realizar. El código Qiskit es compilado internamente a QAsm [MAB<sup>+</sup>18] que puede correr tanto en la computadora cuántica, como en un simulador en una computadora clásica.

Un paradigma alternativo es el de “control cuántico” [AG05, AD17, DCGMV19]. La idea es que si queremos superar la etapa de circuitos (de la misma manera que lenguajes de alto nivel superaron la etapa de los circuitos electrónicos), debemos poder describir operaciones que, al ser compiladas, se transformen en circuitos cuánticos. Lo que se pretende es abrir la “caja negra” que representa a las operaciones cuánticas, y que la misma pueda ser descripta en el lenguaje. Para ello el lenguaje tiene que tener características cuánticas, como permitir la superposición de programas, o el enredo de los mismos. Sin embargo, el lenguaje deberá proveer una herramienta que asegure que lo que se escriba en el mismo, tenga una traducción (una compilación) directa a la computadora cuántica.

Un paradigma intermedio, llamado “control probabilístico” [DC17], consiste en utilizar el formalismo de las matrices densidad, que permite representar mezclas probabilísticas de estados cuánticos puros, generalizándolo a mezclas probabilísticas de programas. Las matrices densidad no son utilizadas usualmente para describir algoritmos, donde el output esperado es el resultado clásico de medir un estado cuántico. Sin embargo, el formalismo es utilizado ampliamente en teorías como la decoherencia cuántica. Desde el punto de vista de la computación cuántica como una descripción de la física, estudiar el paradigma del control probabilístico, y posibles simulaciones del mismo, es una herramienta más en el estudio de la mecánica cuántica.

En [DC17] se presentan dos extensiones al lambda cálculo para computación cuántica:  $\lambda_\rho$  y  $\lambda_\rho^\circ$ . Ambas utilizan matrices densidad, sólo que en la primera,  $\lambda_\rho$ , si la matriz densidad representa a un estado puro al inicio del cálculo, terminará en un estado puro al finalizar, y, por lo tanto,  $\lambda_\rho$  está en el paradigma del control clásico. En cambio, la segunda extensión,  $\lambda_\rho^\circ$ , introduce la noción de matriz densidad generalizada, dando posibilidad de llevar la evolución de estados mixtos. Por lo tanto,  $\lambda_\rho^\circ$  está en el paradigma de control probabilístico.

## Objetivos y actividades

El objetivo de esta tesina es definir e implementar un compilador de  $\lambda_\rho$  en código Qiskit, que corra tanto en el simulador como en la computadora cuántica IBM-Q, y definir e implementar un compilador de  $\lambda_\rho^\circ$  junto con un simulador del mismo que utilice matrices de densidad generalizadas.

---

<sup>1</sup><https://www.ibm.com/quantum-computing/>

<sup>2</sup><https://qiskit.org>

Para esto, se definen las siguientes actividades.

1. Definir una traducción desde  $\lambda_\rho$  a Python que utilice la librería Qiskit. Parte del trabajo es explorar las técnicas disponibles y escoger las convenientes.
2. Programar un compilador en Haskell que siga la traducción especificada en el paso anterior. El objetivo es que el código generado pueda ser simulado/ejecutado utilizando la librería Qiskit.
3. Desarrollar una herramienta que permita simular los programas en  $\lambda_\rho^\circ$  siguiendo la semántica especificada en [DC17], mediante el uso de matrices de densidad. Esto involucra en primer medida la implementación de un compilador que procese el código de  $\lambda_\rho^\circ$ .

#### Cronograma de actividades

Actividad Mes	1	2	3	4	5	6	7	8	9	10	11	12
Tarea 1			✓									
Tarea 2						✓						
Tarea 3											✓	
Redacción												✓

## Referencias

- [AD17] Pablo Arrighi and Gilles Dowek. Lineal: A linear-algebraic lambda-calculus. *Logical Methods in Computer Science*, 13(1:8), 2017.
- [AG05] Thorsten Altenkirch and Jonathan J. Grattage. A functional quantum programming language. *Proceedings of LICS*, 2005:249–258, 2005.
- [DC17] Alejandro Díaz-Caro. A lambda calculus for density matrices with classical and probabilistic controls. *Lecture Notes in Computer Science*, 10695:448–467, 2017. Proceedings of APLAS 2017.
- [DCGMV19] Alejandro Díaz-Caro, Mauricio Guillermo, Alexandre Miquel, and Benoît Valiron. Realizability in the unitary sphere. *Proceedings of LICS*, 2019:1–13, 2019.
- [MAB<sup>+</sup>18] David C. McKay, Thomas Alexander, Luciano Bello, Michael J. Biercuk, Lev Bishop, Jiayin Chen, Jerry M. Chow, Antonio D. Córcoles, Daniel Egger, Stefan Filipp, and et al. Qiskit backend specifications for openqasm and openpulse experiments. *arXiv.org*, Sep 2018.
- [Sel04] Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, 2004.