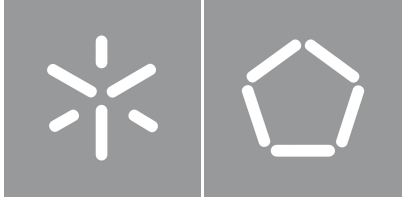
University of Minho
School of Engineering

Bruna Filipa Martins Salgado

A Metric Equational System for Quantum Computation

February 2024



University of Minho
School of Engineering

Bruna Filipa Martins Salgado

A Metric Equational System for Quantum Computation

Master's Dissertation
Master in Physics Engineering

Work carried out under the supervision of
Renato Jorge Araújo Neves

February 2024

Copyright and Terms of Use for Third Party Work

This dissertation reports on academic work that can be used by third parties as long as the internationally accepted standards and good practices are respected concerning copyright and related rights.
This work can thereafter be used under the terms established in the license below.
Readers needing authorization conditions not provided for in the indicated licensing should contact the author through the RepositóriUM of the University of Minho.

License granted to users of this work:

[Caso o autor pretenda usar uma das licenças Creative Commons, deve escolher e deixar apenas um dos seguintes ícones e respetivo lettering e URL, eliminando o texto em itálico que se lhe segue. Contudo, é possível optar por outro tipo de licença, devendo, nesse caso, ser incluída a informação necessária adaptando devidamente esta minuta]

CC BY

https://creativecommons.org/licenses/by/4.0/ *[Esta licença permite que outros distribuam, remixem, adaptem e criem a partir do seu trabalho, mesmo para fins comerciais, desde que lhe atribuam o devido crédito pela criação original. É a licença mais flexível de todas as licenças disponíveis. É recomendada para maximizar a disseminação e uso dos materiais licenciados.]*

CC BY-SA

https://creativecommons.org/licenses/by-sa/4.0/ *[Esta licença permite que outros remisturem, adaptem e criem a partir do seu trabalho, mesmo para fins comerciais, desde que lhe atribuam o devido crédito e que licenciem as novas criações ao abrigo de termos idênti-*

cos. Esta licença costuma ser comparada com as licenças de software livre e de código aberto «copyleft». Todos os trabalhos novos baseados no seu terão a mesma licença, portanto quaisquer trabalhos derivados também permitirão o uso comercial. Esta é a licença usada pela Wikipédia e é recomendada para materiais que seriam beneficiados com a incorporação de conteúdos da Wikipédia e de outros projetos com licenciamento semelhante.]

CC BY-ND

https://creativecommons.org/licenses/by-nd/4.0/ *[Esta licença permite que outras pessoas usem o seu trabalho para qualquer fim, incluindo para fins comerciais. Contudo, o trabalho, na forma adaptada, não poderá ser partilhado com outras pessoas e têm que lhe ser atribuídos os devidos créditos.]*

CC BY-NC

https://creativecommons.org/licenses/by-nc/4.0/ *[Esta licença permite que outros remisturem, adaptem e criem a partir do seu trabalho para fins não comerciais, e embora os novos trabalhos tenham de lhe atribuir o devido crédito e não possam ser usados para fins comerciais, eles não têm de licenciar esses trabalhos derivados ao abrigo dos mesmos termos.]*

CC BY-NC-SA

https://creativecommons.org/licenses/by-nc-sa/4.0/ *[Esta licença permite que outros remisturem, adaptem e criem a partir do seu trabalho para fins não comerciais, desde que lhe atribuam a si o devido crédito e que licenciem as novas criações ao abrigo de termos idênticos.]*

CC BY-NC-ND

https://creativecommons.org/licenses/by-nc-nd/4.0/ *[Esta é a mais restritiva das nossas seis licenças principais, só permitindo que outros façam download dos seus trabalhos e os compartilhem desde que lhe sejam atribuídos a si os devidos créditos, mas sem que possam alterá- los de nenhuma forma ou utilizá-los para fins comerciais.]*

Acknowledgements

Write your acknowledgements here. Do not forget to mention the projects and grants that you have benefited from while doing your research, if any. Ask your supervisor about the specific textual format to use. (Funding agencies are quite strict about this.)

Statement of Integrity

I hereby declare having conducted this academic work with integrity.
I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.
I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, Braga, February 2024

Bruna Filipa Martins Salgado

Abstract

Noisy intermediate-scale quantum (NISQ) computers are expected to operate with severely limited hardware resources. Precisely controlling qubits in these systems comes at a high cost, is susceptible to errors, and faces scarcity challenges. Therefore, error analysis is indispensable for the design, optimization, and assessment of NISQ computing. Nevertheless, the analysis of errors in quantum programs poses a significant challenge. The overarching goal of the M.Sc. project is to provide a fully-fledged quantum programming language on which to study metric program equivalence in various scenarios, such as in quantum algorithmics and quantum information theory.

Keywords approximate equivalence, λ-calculus, metric equations

Resumo

Escrever aqui o resumo (pt)

Palavras-chave palavras, chave, aqui, separadas, por, vírgulas

Contents

List of Figures

List of Tables

Part I

Introductory material

Chapter 1

Introduction

1.1 Motivation and Context

Quantum computing dates back to 1982 when Nobel laureate Richard Feynman proposed the idea that constructing computers founded on the principles of quantum mechanics could efficiently simulate quantum systems of interest to physicists, whereas this seemed to be very difficult with classical computers [Feynman (1982)].

This paradigm holds immense promise, as evidenced by several compelling results in computational complexity theory [Shor (1994); Grover (1996)]. While hardware advancements have brought the scientific community closer to realizing this potential, the ultimate goal the ultimate goal is yet to be accomplished. A NISQ quantum computer equipped with 50-100 qubits may surpass the capabilities of current classical computers, yet the impact of quantum noise, such as decoherence in entangled states, imposes limitations on the size of quantum circuits that can be executed reliably [Preskill (2018)]. Unfortunately, general-purpose error correction techniques [Calderbank and Shor (1996); Gottesman (1997); Steane (1996)] consume a substantial number of qubits, making it difficult for NISQ devices to make use of them in the near term. For instance, the implementation of a single logical qubit may require between 10^3 and 10^4 physical qubits [Fowler et al. (2012)].

To reconcile quantum computation with NISQ computers, quantum compilers perform transformations for error mitigation [Wallman and Emerson (2016)] and noise-adaptive optimization [Murali et al. (2019)]. Additionally, current quantum computers only support a restricted, albeit universal, set of quantum operations. As a result, nonnative operations must be decomposed into sequences of native operations before execution [Harrow et al. (2002), Burgholzer and Wille (2020)]. In general, perfect computational universality is not sought, but only the ability to approximate any quantum algorithm, with a preference for minimizing the use of

additional gates beyond the original requirements. The assessment of these compiler transformations necessitates a comparison of the error bounds between the source and compiled quantum programs. Furthermore, in quantum information theory, the concept of an $\epsilon -$ approximation channel is fundamental when studying quantum teleportation via noisy channels [Watrous (2018)]. This suggests the development of appropriate notions of approximate program equivalence, *in lieu* of the classical program equivalence and underlying theories that typically hinge on the idea that equivalence is binary, *i.e.* two programs are either equivalent or they are not [Winskel (1993)].

As previously noted, Shor's and Grover's algorithms have played a pivotal role in sparking heightened interest within the scientific community toward quantum computing research. On these bases, various endeavors to establish quantum programming languages have surfaced over the past 20 years. These include imperative languages such as Qiskit [Qiskit contributors (2023)] and Silq [Bichsel et al. (2020)], as well as functional languages such as Quipper [Green et al. (2013)] and Q# [Svore et al. (2018)]. On one hand, the design of quantum programming languages is strongly oriented towards implementing quantum algorithms. On the other hand, the definition of functional paradigmatic languages or functional calculi serves as a valuable tool for delving into theoretical aspects of quantum computing, particularly exploring the foundational basis of quantum computation [Zorzi (2016)]. Given the nature of this work, the focus will be on quantum languages designed with this latter aspect in mind.

QPL, a quantum language within the functional programming paradigm, marks a significant milestone in this context [Selinger (2004)]. This is a first-order functional language featuring a static type system based on the idea of classical control and quantum data.

Most of the current research on algorithms and programming languages assumes that addressing the challenge of noise during program execution will be resolved either by the hardware or through the implementation of fault-tolerant protocols designed independently of any specific application [Chong et al. (2017)]. As previously stated, this assumption is not realistic in the NISQ era. Nonetheless, there have been efforts to address the challenge of approximate program equivalence in the quantum setting. [Hung et al. (2019)] and [Tao et al. (2021)] reason about the issue of noise in a quantum while-language by developing a deductive system to determine how similar a quantum program is from its idealised, noise-free version. An alternative approach was explored in [Dahlqvist and Neves (2022)], using linear

λ-calculus as basis – *i.e* programs are written as linear λ-terms – which has deep connections to both logic and category theory [Girard et al. (1995), Benton (1994)]. Some positive results were achieved in this setting, but much remains to be done.

1.2 Goals

The notion of approximate equivalence for quantum programming explored in [Dahlqvist and Neves (2022)] does not take important operations into account. Specifically, the corresponding mathematical model does not include measurements, classical control flow, or discard operations. Also, the corresponding typing system is often times too strict and cannot properly handle multiple uses of the same resource, such as sampling exactly n-times from a distribution. The overarching goal of this M.Sc. project is to tackle the aforementioned limitations. A successful completion of this goal will provide a fully-fledged quantum programming language on which to study metric program equivalence in various scenarios. This includes not only quantum algorithmics – where, for example, the number of iterations in Grover's algorithm involves approximations – but also quantum information theory, where, for instance, quantum teleportation and the problem of the discrimination of quantum states have important roles [Nielsen and Chuang (2010)].

Chapter 2

State of the Art

2.1 Linear Lambda Calculus

The Lambda-Calculus, developed by Church and Curry in the 1930s, serves as a formal language capturing the key attribute of higher-order functional languages, treating functions as first-class citizens, allowing them to be passed as arguments [Barendregt et al. (1984)]. Beyond its foundational aspects, this calculus incorporates extensions for modeling side effects, including probabilistic or non-deterministic behaviors and shared memory. Centered around the expression of higher-order functions, where functions can serve as inputs or outputs, it emerges as a potent computational tool. Higher-order functions form a pivotal abstraction in practical programming languages such as LISP, Scheme, ML, and Haskell.
In quantum information theory, the role of higher-order functions encompasses two fundamental aspects. The first involves the concept of entangled functions and how well-known quantum phenomena find natural descriptions through such functions. The second concerns the interplay between classical objects and quantum objects in a higher-order context. Quantum computation conventionally handles classical and quantum data, while the higher-order context introduces a third data type: functions. These functions fall into two categories - those "quantum-like" (entangled, single-use) and those "classical-like" (duplicable, reusable). Remarkably, this classification transcends input/output types, highlighting the coexistence of quantum-like functions operating on classical data and classical-like functions operating on quantum data. [Selinger et al. (2009)].

2.1.1 Syntax

The grammar and term formation rules of the linear lambda calculus, discussed in [Dahlqvist and Neves (2022)], are presented in this subsection.

The definition of the grammar for linear lambda calculus is as follows, where G represents a set of ground types.

$$\mathbb{A} ::= X \in G \mid \mathbb{I} \mid \mathbb{A} \otimes \mathbb{A} \mid \mathbb{A} \oplus \mathbb{A} \mid \mathbb{A} \multimap \mathbb{A} \tag{2.1}$$

Regarding the term formation rules, Σ corresponds to a class of sorted operation symbols $f : \mathbb{A}_1, \ldots, \mathbb{A}_n \to \mathbb{A}$, where $n \geq 1$. Typing contexts are represented as lists $x_1 : \mathbb{A}_1, \ldots, x_n : \mathbb{A}_n$ of typed variables, with each variable x_i (where $1 \leq i \leq n$) occurring at most once in $x_1, \ldots, x_n$. The typing contexts are denoted by greek letters Γ, Δ, and E. The concept of shuffling is employed to construct a linear typing system that ensures the admissibility of the exchange rule and enables unambiguous reference to judgment's denotations $[\![\Gamma \triangleright v : \mathbb{A}]\!]$. Shuffling is defined as a permutation of typed variables in a sequence of contexts, $\Gamma_1, \ldots, \Gamma_n$, preserving the relative order of variables within each Γ_i. For instance, if $\Gamma_1 = x : \mathbb{A}, y : \mathbb{B}$ and $\Gamma_2 = z : \mathbb{C}$, then $z : \mathbb{C}, x : \mathbb{A}, y : \mathbb{B}$ is a valid shuffle of Γ_1, Γ_2. On the other hand, $y : \mathbb{B}, x : \mathbb{A}, z : \mathbb{C}$ is not a shuffle because it alters the occurrence order of x and y in Γ_1. The set of shuffles in $\Gamma_1, \ldots, \Gamma_n$ is denoted as $\mathsf{Sf}(\Gamma_1, \ldots, \Gamma_n)$. The term formation rules of the linear lambda calculus are shown in Figure 1.

$$\frac{\Gamma_i \triangleright v_i : \mathbb{A}_i \quad f : \mathbb{A}_1, \ldots, \mathbb{A}_n \to \mathbb{A} \in \Sigma \quad E \in \mathsf{Sf}(\Gamma_1; \ldots; \Gamma_n)}{E \triangleright f(v_1, \ldots, v_n) : \mathbb{A}}(\mathsf{ax}) \qquad \frac{}{x : \mathbb{A} \triangleright x : \mathbb{A}}(\mathsf{hyp})$$

$$\frac{}{- \triangleright * : \mathbb{I}}(\mathbb{I}_i) \qquad \frac{\Gamma \triangleright v : \mathbb{A} \otimes \mathbb{B} \quad \Delta, x : \mathbb{A}, y : \mathbb{B} \triangleright w : \mathbb{C} \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{E \triangleright \mathsf{pm}\ v\ \mathsf{to}\ x \otimes y.w : \mathbb{C}}(\otimes_e)$$

$$\frac{\Gamma \triangleright v : \mathbb{A} \quad \Delta \triangleright w : \mathbb{B} \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{E \triangleright v \otimes w : \mathbb{A} \otimes \mathbb{B}}(\otimes_i) \qquad \frac{\Gamma \triangleright v : \mathbb{I} \quad \Delta \triangleright w : \mathbb{A} \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{E \triangleright v\ \mathsf{to}\ *.w : \mathbb{A}}(\mathbb{I}_e)$$

$$\frac{\Gamma, x : \mathbb{A} \triangleright v : \mathbb{B}}{\Gamma \triangleright \lambda x : \mathbb{A}.v : \mathbb{A} \multimap \mathbb{B}}(\multimap_i) \qquad \frac{\Gamma \triangleright v : \mathbb{A} \multimap \mathbb{B} \quad \Delta \triangleright w : \mathbb{A} \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{E \triangleright vw : \mathbb{B}}(\multimap_e) \qquad \frac{\Gamma \triangleright v : \mathbb{A}}{\Gamma \triangleright \mathsf{dis}(v) : \mathbb{I}}(\mathsf{dis})$$

Figure 1: Term formation rules of affine lambda calculus.

The no-cloning theorem states that it is impossible to duplicate a quantum bit [Wootters and Zurek (1982)]. This principle is upheld by the type system outlined in Figure 1, which does not

allow the repeated use of a variable (seen as a quantum resource). Nevertheless, the linearity constraint is often deemed too restrictive, prompting research into relaxing it in various computational paradigms. In [Dahlqvist and Neves (2023)], the controlled use of a resource multiple times is explored within approximate program equivalence paradigms. Moreover, the grammar introduced allows the specification of how many times a resource can be used—a notion particularly relevant in quantum computation, especially within the NISQ era where resources are scarce.

2.1.2 Metric equational system

Metric equations [Mardare et al. (2016), Mardare et al. (2017)] are a strong candidate for reasoning about approximate program equivalence. These equations take the form of $t =_\epsilon s$, where ϵ is a non-negative rational representing the "maximum distance" between the two terms t and s. The metric equational system for linear lambda calculus is depicted in Figure 2 [Dahlqvist and Neves (2022)].

$$\frac{}{v =_0 v}\ (\text{refl}) \qquad \frac{v =_q w \quad w =_r u}{v =_{q+r} u}\ (\text{trans}) \qquad \frac{v =_q w \quad r \geq q}{v =_r w}\ (\text{weak})$$

$$\frac{\forall r > q.\ v =_r w}{v =_q w}\ (\text{arch}) \qquad \frac{\forall i \leq n.\ v =_{q_i} w}{v =_{\vee q_i} w}\ (\text{join}) \qquad \frac{v =_q w \quad v' =_r w'}{v \otimes v' =_{q+r} w \otimes w'}$$

$$\frac{\forall i \leq n.\ v_i =_{q_i} w_i}{f(v_1, ..., v_n) =_{\Sigma q_i} f(w_1, ..., , w_n)} \qquad \frac{v =_q w \quad v' =_r w'}{v \text{ to } *.v' =_{q+r} w \text{ to } *.w'} \qquad \frac{v =_q w}{\lambda x : \mathbb{A}.\ v =_q \lambda x : \mathbb{A}.\ w}$$

$$\frac{v =_q w \quad v' =_r w'}{\text{pm } v \text{ to } x \otimes y.\ v' =_{q+r} \text{pm } w \text{ to } x \otimes y.\ w'} \qquad \frac{v =_q w \quad v' =_r w'}{vv' =_{q+r} ww'}$$

$$\frac{\Gamma \triangleright v =_q w : \mathbb{A} \quad \Delta \in \text{perm}(\Gamma)}{\Delta \triangleright v =_q w : \mathbb{A}} \qquad \frac{v =_q w \quad v' =_r w'}{v[v'/x] =_{q+r} w[w'/x]}$$

Figure 2: Metric equational system

In the quantum paradigm, a potential notion of approximate equivalence arises from the so-called diamond norm [Watrous (2018)], which induces a metric (roughly, a distance function) on the space of quantum programs (seen semantically as completely positive trace-preserving super-operators). This norm relies on another norm known as the trace norm.

The $\|\|_1$ latter is defined by $\|A\|_1 = \text{Tr}\sqrt{A^\dagger A}$ for matrices $A \in \mathbb{C}^{n\times n}$. The trace distance between two super-operators $E, E' : \mathbb{C}^{n\times n} \to \mathbb{C}^{m\times m}$, denoted as $T(E, E')$, is defined as follows:

$$T(E, E') = \max\{\|(E - E')A\|_1 \mid \|A\|_1 = 1\} \tag{2.2}$$

Unfortunately, this norm is not stable under tensoring [Watrous (2018)], and consequently, the diamond norm, which is based on the trace norm, is used instead. The diamond norm between two super-operators $E, E' : \mathbb{C}^{n\times n} \to \mathbb{C}^{m\times m}$ is defined as:

$$\|E - E'\|_\diamond = T(E \otimes I_n, E' \otimes I_n) \tag{2.3}$$

where I_n is the identity super-operator over the space $\mathbb{C}^{n\times n}$. The notion of a diamond norm is used in [Dahlqvist and Neves (2022)] which introduces a simple metric theory based on the idea of approximating a quantum operation. The authors argue that their deductive system allows to compute an approximate distance between two quantum programs easily as opposed to computing an exact distance "semantically" which tends to involve quite complex operators. Other works in this spirit include [Hung et al. (2019)] and [Tao et al. (2021)]. They reason about the issue of noise in a quantum while-language by developing a deductive system to determine how similar a quantum program is from its idealised, noise-free version. The former introduces the (Q,λ)-diamond norm which analyzes the output error given that the input quantum state satisfies some quantum predicate Q to degree λ. However, it does not specify any practical method for obtaining non-trivial quantum predicates. In fact, the methods used in [Hung et al. (2019)] cannot produce any post conditions other than $(I, 0)$ (*i.e.*, the identity matrix I to degree 0, analogous to a "true" predicate) for large quantum programs. The latter specifically addresses and delves into this aspect.

2.1.3 Interpretation

In order to define the interpretation of judgments $\Gamma \triangleright v : \mathbb{A}$, it is necessary to establish some notation first. Considering $v \in V$, $w \in W$, and $u \in U$ where V, W, U represent vector spaces, $\text{sw}_{V,W} : V \otimes W \to W \otimes V$, denotes the swap operator, defined as $\text{sw}_{V,W} = v \otimes w \mapsto w \otimes v$; $\rho_V : \mathbb{C} \otimes V \to V$ is the left unitor defined as $\rho_V = 1 \otimes v \mapsto v$; $\lambda_V : V \otimes \mathbb{C} \to V$ is the right unitor defined as $\lambda_V = v \otimes 1 \mapsto v$; $\alpha_{V,W,U} : V \otimes (W \otimes U) \to (V \otimes W) \otimes U$ is the left associator, defined as $\alpha_{V,W,U} = v \otimes (w \otimes u) \mapsto (v \otimes w) \otimes u$; and $!_V : V \to \mathbb{C}$ is the trace operation applied to a vector, defined as $!_V = v \to \text{Tr}v$. Moreover, for all operators

$f : V \otimes W \to U$, the operator $\overline{f} : V \to (W \multimap U)$ denotes the corresponding curried version, defined as $\overline{f}(v) = w \mapsto f(v, w)$. The subscripts in these operators will be omitted unless ambiguity arises.

For all ground types $X \in G$ the interpretation of $[\![X]\!]$ is postulated as a vector space V. Types are interpreted inductively using the unit $\mathbb{I}$, the tensor $\otimes$, and the linear map $\multimap$. Given a non-empty context $\Gamma = \Gamma', x : \mathbb{A}$, its interpretation is defined by $[\![\Gamma', x : \mathbb{A}]\!] = [\![\Gamma']\!] \otimes [\![\mathbb{A}]\!]$ if Γ' is non-empty and $[\![\Gamma', x : \mathbb{A}]\!] = [\![\mathbb{A}]\!]$ otherwise. The empty context $-$ is interpreted as $[\![-]\!] = \mathbb{I}$. Given $X_1, ..., X_n \in V$, the n-tensor $(\ldots(X_1 \otimes X_2) \otimes \ldots) \otimes X_n$ is denoted as $X_1 \otimes \ldots \otimes X_n$, and similarly for operators.

"Housekeeping" operators are employed to handle interactions between context interpretation and the vectorial model. Given $\Gamma_1, \ldots, \Gamma_n$, the operator that splits $[\![\Gamma_1, \ldots, \Gamma_n]\!]$ into $[\![\Gamma_1]\!] \otimes \ldots \otimes [\![\Gamma_n]\!]$ is denoted by $\mathsf{sp}_{\Gamma_1;\ldots;\Gamma_n} : [\![\Gamma_1, \ldots, \Gamma_n]\!] \to [\![\Gamma_1]\!] \otimes \ldots \otimes [\![\Gamma_n]\!]$. On the other hand, $\mathsf{jn}_{\Gamma_1;\ldots;\Gamma_n}$ denotes the inverse of $\mathsf{sp}_{\Gamma_1;\ldots;\Gamma_n}$. Next, given $\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta$, the operator permuting x and y is denoted by $\mathsf{exch}_{\Gamma,x:\mathbb{A},y:\mathbb{B},\Delta} : [\![\Gamma, x : \mathbb{A}, y : \mathbb{B}, \Delta]\!] \to [\![\Gamma, y : \mathbb{B}, x : \mathbb{A}, \Delta]\!]$. The shuffling operator $\mathsf{sh}_E : [\![E]\!] \to [\![\Gamma_1, \ldots, \Gamma_n]\!]$ is defined as a suitable composition of exchange operators.

For every operation symbol $f : \mathbb{A}_1, \ldots, \mathbb{A}_n \to \mathbb{A}$ we assume the existence of an operator $[\![f]\!] : [\![\mathbb{A}_1]\!] \otimes \ldots \otimes [\![\mathbb{A}_n]\!] \to [\![\mathbb{A}]\!]$. The interpretation of judgments is defined by induction over derivations according to the rules in Figure 3 [Dahlqvist and Neves (2022)].

$$\frac{[\![\Gamma_i \triangleright v_i : \mathbb{A}_i]\!] = m_i \quad f : \mathbb{A}_1, \ldots, \mathbb{A}_n \in \Sigma \quad E \in \mathsf{Sf}(\Gamma_1; \ldots; \Gamma_n)}{[\![E \triangleright f(v_1, \ldots, v_n) : \mathbb{A}]\!] = [\![f]\!] \cdot (m_1 \otimes \ldots \otimes m_n) \cdot \mathsf{sp}_{\Gamma_1;\ldots;\Gamma_n} \cdot \mathsf{sh}_E} \qquad \frac{}{[\![x : \mathbb{A} \triangleright x : \mathbb{A}]\!] = \mathsf{id}_{[\![\mathbb{A}]\!]}}$$

$$\frac{}{[\![- \triangleright * : \mathbb{I}]\!] = \mathsf{id}_{[\![\mathbb{I}]\!]}} \qquad \frac{[\![\Gamma \triangleright v : \mathbb{A} \otimes \mathbb{B}]\!] = m \quad [\![\Delta, x : \mathbb{A}, y : \mathbb{B} \triangleright w : \mathbb{C}]\!] = n \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{[\![E \triangleright \mathsf{pm}\ v\ \mathsf{to}\ x \otimes y.w : \mathbb{C}]\!] = n \cdot \mathsf{jn}_{\Delta;\mathbb{A};\mathbb{B}} \cdot \alpha \cdot \mathsf{sw} \cdot (m \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E}$$

$$\frac{[\![\Gamma \triangleright v : \mathbb{A}]\!] = m \quad [\![\Delta \triangleright w : \mathbb{B}]\!] = n \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{[\![E \triangleright v \otimes w : \mathbb{A} \otimes \mathbb{B}]\!] = (m \otimes n) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E}$$

$$\frac{[\![\Gamma \triangleright v : \mathbb{I}]\!] = m \quad [\![\Delta \triangleright w : \mathbb{A}]\!] = n \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{[\![E \triangleright v\ \mathsf{to}\ *.w : \mathbb{A}]\!] = n \cdot \lambda \cdot (m \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E} \qquad \frac{[\![\Gamma, x : \mathbb{A} \triangleright v : \mathbb{B}]\!] = m}{[\![\Gamma \triangleright \lambda x : \mathbb{A}.\, v : \mathbb{A} \multimap \mathbb{B}]\!] = \overline{m \cdot \mathsf{jn}_{\Gamma;\mathbb{A}}}}$$

$$\frac{[\![\Gamma \triangleright v : \mathbb{A} \multimap \mathbb{B}]\!] = m \quad [\![\Delta \triangleright w : \mathbb{A}]\!] = n \quad E \in Sf(\Gamma; \Delta)}{[\![E \triangleright vw : \mathbb{A}]\!] = \mathsf{app} \cdot (m \otimes n) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E} \qquad \frac{[\![\Gamma \triangleright v : \mathbb{A}]\!] = f}{[\![\Gamma \triangleright \mathsf{dis}(v) : \mathbb{I}]\!] = !_{[\![\mathbb{A}]\!]} \cdot f}$$

Figure 3: Judgment interpretation

2.2 Quantum Lambda Calculus

In the case of quantum lambda calculus, which combines classical and quantum features, it is natural to consider two distinct basic data types: a type *bit* of classical bits and a type *qbit* of quantum bits. The interpretation of these types is defined as $[\![\mathit{bit}]\!] = \mathbb{C} \oplus \mathbb{C}$ and $[\![\mathit{qbit}]\!] = \mathbb{C}^{2\cdot 2}$. The type $\mathbb{I}$ is interpreted as $[\![\mathbb{I}]\!] = \mathbb{C}$.

The following operations are considered: $\mathit{new}\,0 : \mathbb{I} \multimap \mathit{bit}$, $\mathit{new}\,1 : \mathbb{I} \multimap \mathit{bit}$, $q : \mathit{bit} \multimap \mathit{qbit}$, $\mathit{meas} : \mathit{qbit} \to \mathit{bit}$, and $U : \mathit{qbit}, \ldots, \mathit{qbit} \to \mathit{qbit}^{\otimes n}$. Their correspondent judgment interpretation is shown in Figure 4.

$$[\![\mathit{new}\,0]\!] : \mathbb{C} \multimap [\![\mathit{bit}]\!] \qquad [\![\mathit{new}\,1]\!] : \mathbb{C} \multimap [\![\mathit{bit}]\!] \qquad [\![q]\!] : [\![\mathit{bit}]\!] \multimap [\![\mathit{qbit}]\!]$$

$$1 \mapsto (1,0) \qquad 1 \mapsto (0,1) \qquad (a,b) \mapsto \left(\begin{smallmatrix} a & 0 \\ 0 & b \end{smallmatrix}\right)$$

$$[\![\mathit{meas}]\!] : [\![\mathit{qbit}]\!] \to [\![\mathit{bit}]\!] \qquad [\![U]\!] : [\![\mathit{qbit}]\!]^{\otimes n} \to [\![\mathit{qbit}]\!]^{\otimes n}$$

$$\rho \mapsto (\mathsf{Tr}(M_0 \rho M_0^\dagger), \mathsf{Tr}(M_1 \rho M_1^\dagger)) \qquad \rho \mapsto U \rho U^\dagger$$

Figure 4: Judgment interpretation of the operations in quantum lambda calculus.

Chapter 3

The problem and its challenges

The problem and its challenges.

3.1 Images

Example of inserting an image as displayed text,

— wrapped into the text, bla-bla
— or as a floating body.

Figure 5: Caption

Part II

Core of the Dissertation

Chapter 4

Contribution

Main result(s) and their scientific evidence

4.1 Introduction

4.2 Summary

4.3 Integration of conditionals

The notion of approximate equivalence for quantum programming explored in [Dahlqvist and Neves (2022)] does not encompass classical control flow. As a result, preliminary work based on [Crole (1993); Selinger (2013)] has been undertaken to address the integration of conditionals.

4.3.1 Integration of conditionals

The term formation rules for conditionals are depicted in Figure 6.

$$\frac{\Gamma \triangleright v : \mathbb{A}}{\Gamma \triangleright \mathsf{inl}(v) : \mathbb{A} \oplus \mathbb{B}}\,(\mathsf{inl}) \qquad \frac{\Gamma \triangleright v : \mathbb{B}}{\Gamma \triangleright \mathsf{inr}(v) : \mathbb{A} \oplus \mathbb{B}}\,(\mathsf{inr})$$

$$\frac{\Gamma \triangleright v : \mathbb{A} \oplus \mathbb{B} \quad \Delta, x : \mathbb{A} \triangleright w : \mathbb{C} \quad \Delta, y : \mathbb{B} \triangleright u : \mathbb{C} \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{E \triangleright\ \mathsf{cond}\, v\, \{\mathsf{inl}(x) \Rightarrow w; \mathsf{inr}(y) \Rightarrow u\} : \mathbb{C}}\,(\mathsf{case})$$

Figure 6: Term formation rules for conditionals

Considering $v \in V$, $w \in W$, and $u \in U$ where V, W, U represent vector spaces, $\mathsf{IL}_V : V \to$

$V \oplus W$, denotes the left injection operator, defined as $\mathsf{IL}_V = v \mapsto (v, 0)$; $\mathsf{IR}_V : V \to W \oplus V$, denotes the right injection operator, defined as $\mathsf{IR}_V = v \mapsto (0, v)$; and $\mathsf{dist}_{V,W,U} : V \otimes (W \oplus U) \to (V \otimes W) \oplus (V \otimes U)$, denotes the distributive property of the tensor product over the direct sum, defined as $\mathsf{dist}_{V,W,U} = v \otimes (w, u) \mapsto (v \otimes w, v \otimes u)$. The subscripts in these operators will be omitted unless ambiguity arises. Moreover, the operation either corresponds to:

$$\frac{\begin{array}{c} V \to U \\ W \to U \end{array}}{\begin{array}{l} [T, S] : V \oplus W \to U \\ [T, S] = (v, w) \mapsto T(v) + S(w) \end{array}} \tag{4.1}$$

The interpretation of conditionals is illustrated in Figure 7.

$$\frac{[\![\Gamma \triangleright v : \mathbb{A}]\!] = m}{[\![\Gamma \triangleright \mathsf{inl}(v) : \mathbb{A} \oplus \mathbb{B}]\!] = \mathsf{IL} \cdot m} \qquad \frac{[\![\Gamma \triangleright v : \mathbb{B}]\!] = m}{[\![\Gamma \triangleright \mathsf{inr}(v) : \mathbb{A} \oplus \mathbb{B}]\!] = \mathsf{IR} \cdot m}$$

$$\frac{[\![\Gamma \triangleright v : \mathbb{A} \oplus \mathbb{B}]\!] = b \quad [\![\Delta, x : \mathbb{A} \triangleright w : \mathbb{C}]\!] = p \quad [\![\Delta, x : \mathbb{B} \triangleright w_2 : \mathbb{C}]\!] = q \quad E \in \mathsf{Sf}(\Gamma; \Delta)}{[\![E \triangleright \ \mathsf{cond}\ v\ \{\mathsf{inl}(x) \Rightarrow w;\ \mathsf{inr}(y) \Rightarrow u\} : \mathbb{C}]\!] = \mathsf{either}(p, q) \cdot \mathsf{dist} \cdot \mathsf{sw} \cdot (b \otimes \mathsf{id}) \cdot \mathsf{sp}_{\Gamma;\Delta} \cdot \mathsf{sh}_E} \tag{4.2}$$

Figure 7: Judgment interpretation for conditionals

Proof In order to validate the judgment interpretation for conditionals, it is necessary to demonstrate its correctness.

For the booleans:

$$\begin{array}{c} [\![\Gamma]\!] \xrightarrow{m} [\![\mathbb{A}]\!] \xrightarrow{\mathsf{IL}} [\![\mathbb{A} \oplus \mathbb{B}]\!] \\ [\![\Gamma]\!] \xrightarrow{m} [\![\mathbb{B}]\!] \xrightarrow{\mathsf{IR}} [\![\mathbb{A} \oplus \mathbb{B}]\!] \end{array} \tag{4.3}$$

Now, for the conditional statement:

$$\begin{array}{l} [\![E]\!] \xrightarrow{\mathsf{sh}_E} [\![\Gamma, \Delta]\!] \xrightarrow{\mathsf{sp}_{\Gamma;\Delta}} [\![\Gamma]\!] \otimes [\![\Delta]\!] \xrightarrow{b \otimes \mathsf{id}} ([\![\mathbb{A}]\!] \oplus [\![\mathbb{B}]\!]) \otimes [\![\Delta]\!] \xrightarrow{\mathsf{sw}} [\![\Delta]\!] \otimes ([\![\mathbb{A}]\!] \oplus [\![\mathbb{B}]\!]) \\ \qquad \xrightarrow{\mathsf{dist}} ([\![\Delta]\!] \otimes [\![\mathbb{A}]\!]) \oplus ([\![\Delta]\!] \otimes [\![\mathbb{B}]\!]) \xrightarrow{\mathsf{either}(p,q)} [\![\mathbb{C}]\!] \end{array} \tag{4.4}$$

The quantum lambda calculus with conditionals is illustrated with an example —the quantum teleportation protocol— in **??**.

While the validation of its correctness is ongoing, the metric equations for conditionals are presented in Figure 8. Note that the first two equations are redundant.

$$\frac{v =_q w}{\mathsf{irl}(v) =_q \mathsf{irl}(w)} \qquad \frac{v =_q w}{\mathsf{inr}(v) =_q \mathsf{inr}(w)}$$

$$\frac{v =_q v' \quad w =_r w' \quad u =_s u'}{\mathsf{cond}\ v\ \{\mathsf{irl}(x) \Rightarrow w;\ \mathsf{inr}(y) \Rightarrow u\} =_{\mathsf{max}(q+r;q+s)} \mathsf{cond}\ v'\ \{\mathsf{irl}(x) \Rightarrow w';\ \mathsf{inr}(y) \Rightarrow u'\}}$$

Figure 8: Metric equational system for condicionals

4.3.2 Ilustration: Noisy Quantum Teleportation

To study decoherence in a quantum channel within the presented metric deductive system, one can consider the application of a dephasing channel in the quantum teleportation protocol with a certain probability p. This is exemplified for probabilities $p = 0.5$ and $p = 0.25$. It is worth noting that similar exercises can be done for scenarios such as a malicious attack involving a bit flip during measurement or the presence of a noisy channel.

Chapter 5

Applications

Application of main result (examples and case studies)

5.1 Introduction

5.2 Summary

Chapter 6

Conclusions and future work

Conclusions and future work.

6.1 Conclusions

6.2 Prospect for future work

Chapter 7

Planned Schedule

7.1 Activities

Task	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
Background and **SOA**	•	•	•							
PDR preparation		•	•	•						
Contribution				•	•	•	•	•	•	
Writing up							•	•	•	•

Table 1: Activities Plan

Bibliography

Hendrik P Barendregt et al. *The lambda calculus*, volume 3. North-Holland Amsterdam, 1984.

P Nick Benton. A mixed linear and non-linear logic: Proofs, terms and models. In *International Workshop on Computer Science Logic*, pages 121–135. Springer, 1994.

Benjamin Bichsel, Maximilian Baader, Timon Gehr, and Martin Vechev. Silq: A high-level quantum language with safe uncomputation and intuitive semantics. In *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 286–300, 2020.

Lukas Burgholzer and Robert Wille. Advanced equivalence checking for quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 40(9):1810–1824, 2020.

A Robert Calderbank and Peter W Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098, 1996.

Frederic T Chong, Diana Franklin, and Margaret Martonosi. Programming languages and compiler design for realistic quantum hardware. *Nature*, 549(7671):180–187, 2017.

Roy L Crole. *Categories for types*. Cambridge University Press, 1993.

Fredrik Dahlqvist and Renato Neves. The syntactic side of autonomous categories enriched over generalised metric spaces. *arXiv preprint arXiv:2208.14356*, 2022.

Fredrik Dahlqvist and Renato Neves. A complete v-equational system for graded lambda-calculus. *arXiv preprint arXiv:2304.02082*, 2023.

Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7), 1982.

Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012.

Jean-Yves Girard, Yves Lafont, and Laurent Regnier. *Advances in linear logic*, volume 222. Cambridge University Press, 1995.

Daniel Gottesman. *Stabilizer codes and quantum error correction*. California Institute of Technology, 1997.

Alexander S Green, Peter LeFanu Lumsdaine, Neil J Ross, Peter Selinger, and Benoît Valiron. Quipper: a scalable quantum programming language. In *Proceedings of the 34th ACM SIGPLAN conference on Programming language design and implementation*, pages 333–342, 2013.

Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

Aram W Harrow, Benjamin Recht, and Isaac L Chuang. Efficient discrete approximations of quantum gates. *Journal of Mathematical Physics*, 43(9):4445–4451, 2002.

Shih-Han Hung, Kesha Hietala, Shaopeng Zhu, Mingsheng Ying, Michael Hicks, and Xiaodi Wu. Quantitative robustness analysis of quantum programs. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–29, 2019.

Radu Mardare, Prakash Panangaden, and Gordon Plotkin. Quantitative algebraic reasoning. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 700–709, 2016.

Radu Mardare, Prakash Panangaden, and Gordon Plotkin. On the axiomatizability of quantitative algebras. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

Prakash Murali, Jonathan M Baker, Ali Javadi-Abhari, Frederic T Chong, and Margaret Martonosi. Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. In *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*, pages 1015–1029, 2019.

Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.

Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, 2004.

Peter Selinger. Lecture notes on the lambda calculus, 2013.

Peter Selinger, Benoıt Valiron, et al. Quantum lambda calculus. *Semantic techniques in quantum computation*, pages 135–172, 2009.

Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

Andrew M Steane. Error correcting codes in quantum theory. *Physical Review Letters*, 77(5): 793, 1996.

Krysta Svore, Alan Geller, Matthias Troyer, John Azariah, Christopher Granade, Bettina Heim, Vadym Kliuchnikov, Mariia Mykhailova, Andres Paz, and Martin Roetteler. Q# enabling scalable quantum computing and development with a high-level dsl. In *Proceedings of the real world domain specific languages workshop 2018*, pages 1–10, 2018.

Runzhou Tao, Yunong Shi, Jianan Yao, John Hui, Frederic T Chong, and Ronghui Gu. Gleipnir: toward practical error analysis for quantum programs. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, pages 48–64, 2021.

Joel J Wallman and Joseph Emerson. Noise tailoring for scalable quantum computation via randomized compiling. *Physical Review A*, 94(5):052325, 2016.

John Watrous. *The theory of quantum information*. Cambridge university press, 2018.

Glynn Winskel. *The formal semantics of programming languages: an introduction*. MIT press, 1993.

William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299 (5886):802–803, 1982.

Margherita Zorzi. On quantum lambda calculi: a foundational perspective. *Mathematical Structures in Computer Science*, 26(7):1107–1195, 2016.

Part III
Appendices

Appendix A
Support work

Auxiliary results which are not main-stream.

Appendix B
Details of results

Details of results whose length would compromise readability of main text.

Appendix C
Listings

Should this be the case.

Appendix D
Tooling

(Should this be the case)

Anyone using LaTeX should consider having a look at TUG , the TeX Users Group .

Place here information about funding, FCT project, etc. in which the work is framed. Leave empty otherwise.